



HOKKAIDO UNIVERSITY

Title	Study on Discovery and Exploration Systems Considering User's Intention
Author(s)	翟, 泓杰
Degree Grantor	北海道大学
Degree Name	博士(情報科学)
Dissertation Number	甲第13076号
Issue Date	2018-03-22
DOI	https://doi.org/10.14943/doctoral.k13076
Doc URL	https://hdl.handle.net/2115/70225
Type	doctoral thesis
File Information	Zhai_HongJie.pdf





HOKKAIDO UNIVERSITY

Study on Discovery and Exploration
Systems Considering User's Intention

A DISSERTATION PRESENTED

BY

HONGJIE ZHAI

TO

GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN THE FIELD OF

COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

HOKKAIDO UNIVERSITY

SAPPORO, HOKKAIDO

MARCH 2018

Study on Discovery and Exploration Systems Considering User's Intention

ABSTRACT

In the last two decades, the amount of data accessible via the Internet has increased greatly, particularly in large-scale, open databases such as Wikipedia for encyclopedic knowledge and ImageNet for images. An answer to the question of what objects and their information attract people would depend on the individuals or groups of people. Each object in those datasets has its own aspects and draws people only when some of those aspects match people's interests or search intentions. Because of the diversity of possible aspects, the number of possible groupings of people and objects is generally very large. When groups are large compared with the population size, we can disregard the problem of diversity, as the number of such groups becomes small and standard techniques such as clustering can find them. In contrast, small or medium-sized groups may represent some important individual aspects and are therefore valuable for some users' groups. From this viewpoint, we would like to monitor even small or medium-sized groups of objects to meet the various demands of corresponding users' groups. In this thesis, based on the above considerations, we discuss and solve the following problems:

- (1) Detection of small or medium-sized communities in large graphs (Chapters 1 and 2).
- (2) Associative search considering the diversity of users' intention and directing them to the contents they are potentially interested in (Chapter 3).
- (3) Discovery of feature associations over two distinct datasets under some analogical hints to focus on implicit aspects of interest to users.

In Chapter 1 we introduce the notion of "Enumerating Maximal Clique Sets with Pseudo-Clique Constraints (MCSs)" and develop an efficient algorithm for the purpose. Cliques are well known as the densest subgraphs among graphs. Many fast clique-enumeration algorithms have been designed, particularly for sparse graphs. However, a major problem with cliques is that so many exist and overlap each other. As the degree of overlap increases, they form a pseudo-clique with cliques as its building blocks. We define a pseudo-clique thus obtained as an MCS. Furthermore, some large graphs may contain too many small cliques. We disregard those that are too small and trivial, and introduce a lower-bound constraint on the size of cliques. Our algorithm equipped with this constraint achieves good performance on a graph with over 1, 000, 000 vertices. We also evaluate the quality of our

approach by predicting protein complexes in a Protein–Protein Interaction (PPI) network. The results with this network show that our algorithm achieves higher accuracy than many other community detection methods.

In Chapter 2 “Enumerating Pseudo-Cliques with Density Lower Bound,” we strengthen the pseudo-clique-enumeration method with the j -core constraint to satisfy a density lower bound. MCS in Chapter 1 consists of overlapped cliques and always contains more than one clique as its core. For many real-world applications, those cores may not be cliques, allowing some disconnection in them while maintaining their density. We must consider this kind of pseudo-clique to cope with more realistic situations. Several efficient but heuristic methods for finding pseudo-cliques have been proposed in the literature. The algorithm proposed in this thesis is complete and efficient enough to process large graphs with 1,000,000 vertices. We suppose a class of pseudo-cliques satisfying a density lower bound. During the enumeration process, which generates pseudo-cliques by adding candidate vertices, we show that only candidates appearing in the j -core of the whole candidate set can be considered. In other words, we can safely prune off many meaningless candidates and ineligible pseudo-cliques that cannot be extended to vertex sets satisfying the lower-bound constraint. Moreover, as the operation for identifying j -cores can be carried out in linear time, the additional task of preserving the density lower-bound constraint during the search is not time-consuming.

In Chapter 3 “Associative Search by Shifting Concepts via Bridges,” we consider how we can access relevant and interesting concepts under an initial query by association via bridges. In the previous two chapters, the problem of user’s intention is related to the size and density of subgraphs as communities. In this chapter, we present a new framework for bringing out users’ implicit intention of accessing what they prefer to find. According to the standard study of the associative search for documents, a user can select some words or documents during the search process. This changes the present concepts of words or documents to new ones not realized at the beginning of the search. As the selected concepts retain some relevance to those at the previous stage of the search, the selection process guides us to some shifted concepts that are brother concepts to the previous ones. One possible way to generate such conceptual shifts would be to ask people who know words or documents from various topics. We call such people mediators and let users select them instead of words or documents. Because the selected mediators talk about concepts not covered by the user’s initial query, those concepts can be bridges to other concepts that the user was not explicitly aware of. Technically, we develop our system based on Formal Concept Analysis to process various concepts including query concepts, mediators’ concepts, and bridges. We test the proposed system by some experiments using Twitter data and verify that some interesting association of concepts can be drawn by our system.

In Chapter 4 “Feature Association Discovery by Linear Algebraic Inference,” we describe

a system that tries to discover feature associations between two distinct domains according to a user's hints. Given two domains represented by data matrices with columns as features and rows as objects, a hint is a set of paired features, over two domains, that a user considers analogous or similar. The task is to find other paired features that seem to share the same or similar analogous relationships with those in the hint. As the user's intention or viewpoint under which paired features are put into correspondence is not explicitly shown to our system, the task is a kind of inference. We treat it as a problem of regarding close features in a common space as potential ones to be paired with. We use Nonnegative Matrix Factorization (NMF). Based on nonnegative trifactorization, which is an extension of NMF, we can project the rows into spaces with the same dimension. Furthermore, because the trifactorization decomposes the rows and columns at the same time, we can also embed the column features into vectors with the same dimension. By applying trifactorization to two datasets separately, features as well as objects are embedded into the same-dimensional space, called a common space. To construct the actual common space for our purpose, we introduce several constraints so that the paired features in a given hint become close vectors in the common space. As these constraints can be expressed as regularization terms, we can process the feature association inference as an optimization. We show experimentally that, by the optimization, our extended NMF achieves a good accuracy of inference for news articles with respect to an answer set constructed by Google Translate.

In the last two decades, the amount of data accessible via the Internet has increased greatly, particularly in large-scale, open databases such as Wikipedia for encyclopedic knowledge and ImageNet for images. An answer to the question of what objects and their information attract people would depend on the individuals or groups of people. Each object in those datasets has its own aspects and draws people only when some of those aspects match people's interests or search intentions. Because of the diversity of possible aspects, the number of possible groupings of people and objects is generally very large. When groups are large compared with the population size, we can disregard the problem of diversity, as the number of such groups becomes small and standard techniques such as clustering can find them. In contrast, small or medium-sized groups may represent some important individual aspects and are therefore valuable for some users' groups. From this viewpoint, we would like to monitor even small or medium-sized groups of objects to meet the various demands of corresponding users' groups. In this thesis, based on the above considerations, we discuss and solve the following problems:

- (1) Detection of small or medium-sized communities in large graphs (Chapters 1 and 2).
- (2) Associative search considering the diversity of users' intention and directing them to the contents they are potentially interested in (Chapter 3).
- (3) Discovery of feature associations over two distinct datasets under some analogical hints to focus on implicit aspects of interest to users.

In Chapter 1 we introduce the notion of “Enumerating Maximal Clique Sets with Pseudo-Clique Constraints (MCSs)” and develop an efficient algorithm for the purpose. Cliques are well known as the densest subgraphs among graphs. Many fast clique-enumeration algorithms have been designed, particularly for sparse graphs. However, a major problem with cliques is that so many exist and overlap each other. As the degree of overlap increases, they form a pseudo-clique with cliques as its building blocks. We define a pseudo-clique thus obtained as an MCS. Furthermore, some large graphs may contain too many small cliques. We disregard those that are too small and trivial, and introduce a lower-bound constraint on the size of cliques. Our algorithm equipped with this constraint achieves good performance on a graph with over 1,000,000 vertices. We also evaluate the quality of our approach by predicting protein complexes in a Protein–Protein Interaction (PPI) network. The results with this network show that our algorithm achieves higher accuracy than many other community detection methods.

In Chapter 2 “Enumerating Pseudo-Cliques with Density Lower Bound,” we strengthen the pseudo-clique-enumeration method with the j -core constraint to satisfy a density lower bound. MCS in Chapter 1 consists of overlapped cliques and always contains more than one clique as its core. For many real-world applications, those cores may not be cliques, allowing some disconnection in them while maintaining their density. We must consider this kind of pseudo-clique to cope with more realistic situations. Several efficient but heuristic methods for finding pseudo-cliques have been proposed in the literature. The algorithm proposed in this thesis is complete and efficient enough to process large graphs with 1,000,000 vertices. We suppose a class of pseudo-cliques satisfying a density lower bound. During the enumeration process, which generates pseudo-cliques by adding candidate vertices, we show that only candidates appearing in the j -core of the whole candidate set can be considered. In other words, we can safely prune off many meaningless candidates and ineligible pseudo-cliques that cannot be extended to vertex sets satisfying the lower-bound constraint. Moreover, as the operation for identifying j -cores can be carried out in linear time, the additional task of preserving the density lower-bound constraint during the search is not time-consuming.

In Chapter 3 “Associative Search by Shifting Concepts via Bridges,” we consider how we can access relevant and interesting concepts under an initial query by association via bridges. In the previous two chapters, the problem of user’s intention is related to the size and density of subgraphs as communities. In this chapter, we present a new framework for bringing out users’ implicit intention of accessing what they prefer to find. According to the standard study of the associative search for documents, a user can select some words or documents during the search process. This changes the present concepts of words or documents to new ones not realized at the beginning of the search. As the selected concepts retain some relevance to those at the previous stage of the search, the selection process guides

us to some shifted concepts that are brother concepts to the previous ones. One possible way to generate such conceptual shifts would be to ask people who know words or documents from various topics. We call such people mediators and let users select them instead of words or documents. Because the selected mediators talk about concepts not covered by the user's initial query, those concepts can be bridges to other concepts that the user was not explicitly aware of. Technically, we develop our system based on Formal Concept Analysis to process various concepts including query concepts, mediators' concepts, and bridges. We test the proposed system by some experiments using Twitter data and verify that some interesting association of concepts can be drawn by our system.

In Chapter 4 "Feature Association Discovery by Linear Algebraic Inference," we describe a system that tries to discover feature associations between two distinct domains according to a user's hints. Given two domains represented by data matrices with columns as features and rows as objects, a hint is a set of paired features, over two domains, that a user considers analogous or similar. The task is to find other paired features that seem to share the same or similar analogous relationships with those in the hint. As the user's intention or viewpoint under which paired features are put into correspondence is not explicitly shown to our system, the task is a kind of inference. We treat it as a problem of regarding close features in a common space as potential ones to be paired with. We use Nonnegative Matrix Factorization (NMF). Based on nonnegative trifactorization, which is an extension of NMF, we can project the rows into spaces with the same dimension. Furthermore, because the trifactorization decomposes the rows and columns at the same time, we can also embed the column features into vectors with the same dimension. By applying trifactorization to two datasets separately, features as well as objects are embedded into the same-dimensional space, called a common space. To construct the actual common space for our purpose, we introduce several constraints so that the paired features in a given hint become close vectors in the common space. As these constraints can be expressed as regularization terms, we can process the feature association inference as an optimization. We show experimentally that, by the optimization, our extended NMF achieves a good accuracy of inference for news articles with respect to an answer set constructed by Google Translate.

Contents

0	INTRODUCTION	I
0.1	Background	I
0.2	Contributions	5
0.3	Thesis Organization	13
1	ENUMERATING MAXIMAL CLIQUE SETS WITH PSEUDO-CLIQUE CONSTRAINT	14
1.1	Introduction	14
1.2	Preliminaries	19
1.3	k -Maximal Clique Set	20
1.4	Enumerating Maximal k -Maximal Clique Set	24
1.5	Experimental Results	33
1.6	Conclusions	44
2	ENUMERATING PSEUDO-CLIQUEs WITH DENSITY LOWER BOUND	52
2.1	Introduction	52
2.2	Preliminaries	56
2.3	Maximal Connected k -Plex	58
2.4	j -Cored k -MPC	59
2.5	Search Control Rules, Right and Left Ones	65
2.6	Algorithm for (j, k) -MPCs	67
2.7	Experimental Results	72
2.8	Conclusions	95
3	ASSOCIATIVE SEARCH BY SHIFTING CONCEPTS VIA BRIDGES	96
3.1	Introduction	96
3.2	Preliminaries	101
3.3	Person-Term Relation and Document-Term Relation	102
3.4	Ranking Persons Based on Mediator Level	103
3.5	Method of Associative Search by Mediators	108

3.6	Extracting Target Concepts for Associative Search	115
3.7	Experimental Results	117
3.8	Conclusions	119
4	FEATURE ASSOCIATION DISCOVERY BY LINEAR ALGEBRAIC INFERENCE	121
4.1	Introduction	121
4.2	Preliminaries	126
4.3	Non-negative Matrix Factorization and Tri-factorization	127
4.4	Common Space Embedding using Tri-factorization	131
4.5	Update Rules and Algorithms	136
4.6	Linear Algebraic Inference	139
4.7	Experiment	141
4.8	Conclusions	146
5	CONCLUSIONS AND FUTURE WORK	147
	APPENDIX A PERMISSION TO USE	157
	REFERENCES	167

Listing of figures

1	Image of the whole system	6
1.1	Target Meta-clique	17
1.2	Image of maximal k -MCS	22
1.3	k -Clique graph	26
1.4	Distribution of clique size for com-Amazon	36
1.5	Vertices coverage rate for com-Amazon	37
1.6	Computation times ($bond = 0$) of RANDOM-10000-1	39
1.7	Computation times ($bond = 0$) of RANDOM-10000-2	40
1.8	Computation times ($bond = 0$) of CA-GRQC	41
1.9	Computation times ($bond = 0$) of COM-AMAZON	42
1.10	Number of Solutions ($bond = 0$) of RANDOM-10000-1	43
1.11	Number of Solutions ($bond = 0$) of RANDOM-10000-2	44
1.12	Number of Solutions ($bond = 0$) of CA-GRQC	45
1.13	Number of Solutions ($bond = 0$) of COM-AMAZON	46
1.14	The Change of Computation Times with $k(j = 1, bond = 0)$ of GEOM-0	47
1.15	The Change of Computation Times with $k(j = 1, bond = 0)$ of CA-GRQC	48
1.16	Computation Times on Large Graphs($bond = 0$) of RANDOM-50000-1	48
1.17	Computation Times on Large Graphs($bond = 0$) of RANDOM-50000-2	49
1.18	Computation Times on Large Graphs($bond = 0$) of RANDOM-100000-1	49
1.19	Computation Times on Large Graphs($bond = 0$) of RANDOM-100000-2	50
1.20	Computation Times on Large Graphs($bond = 0$) of YOUTUBE	50
1.21	Comparison of Fi -score on protein-protein interaction network	51
2.1	Example of $core_j(X)$ for $j = 3$ case	60
2.2	Examples showing how j -core is useful.	62
2.3	Enumeration algorithm for (j, k) -MPCs	66
2.4	Search tree	67
2.5	Scales of networks	68

2.6	Degree distributions of BA	68
2.7	Degree distributions of WS	69
2.8	Degree distributions of GrQc	69
2.9	Degree distributions of DAYS	70
2.10	Degree distributions of DBLP	70
2.11	Degree distributions of Google	71
2.12	Computation times of BA	72
2.13	Computation times of WS	73
2.14	Computation times of GrQc	74
2.15	Computation times of DAYS	75
2.16	Computation times of DBLP	76
2.17	Computation times of Google	77
2.18	Density distributions of solution k -Plexes of BA	79
2.19	Density distributions of solution k -Plexes of WS	80
2.20	Density distributions of solution k -Plexes of GrQc	81
2.21	Density distributions of solution k -Plexes of DAYS	82
2.22	Density distributions of solution k -Plexes of DBLP	83
2.23	Density distributions of solution k -Plexes of Google	84
2.24	Numbers of maximal k -Plexes and (j, k) -MPCs	87
2.25	Difference between numbers of maximal k -Plexes and (j, k) -MPCs	88
2.26	Example of maximal j -cored k -Plex	90
3.1	Example of Associative Connection of Concepts	97
3.2	Clustering Results by Traditional k -Means and Extended k -Means	109
3.3	Example of Associative Connection of Concepts for “Smartphone”	119
4.1	Typical case of inferring new association (f, f') in the common space under the constraint for hint features.	124
4.2	The Image of The Whole System	126
4.3	Example of the Selection Matrix	133
4.4	Change of Cost Function	143

I WOULD LIKE TO DEDICATE THIS DISSERTATION TO MY PARENTS. THERE IS NO DOUBT THAT WITHOUT THEIR SUPPORT AND COUNSEL I COULD NOT HAVE FINISHED THIS THESIS.

Acknowledgments

I WOULD LIKE TO acknowledge the people that contributed to the completion of this dissertation. Firstly, I would like to thank my advisor Prof.Makoto Haraguchi. His guidance through these 5 years are greatly appreciated. I remember when I still was a research student, he gave me a chance to participate the researches which were being done by the master's course student. That was not only my first step to learn how to research, but also the origin of the motivation for me to enter the doctoral course. Since that, he provides much freedom to me so that I could work on various themes I interested in without objection. I must apologize for my selfish on choosing the research theme and the inconvenience caused in all these years. In almost each week, I had chances to discuss with him about not only my works but also the themes of other students. He could always provide valuable suggestions on both techniques and applications, even though I talked about the topics I just learned. I enjoy the discussions and learned so much about scientific knowledge from him. I wished I could be able to provide helpful ideas during the discussions. This wish also became the biggest source of motivations for learning and researching. For these I can not imagine finishing this thesis without his help.

Besides my advisor, I would also like to thank the rest of my thesis committee: Prof.Hiroki Arimura and Prof.Masanori Sugimoto, not only for their comments on this thesis, but also for the insightful advice for my whole research.

I would like to give a special thanks to Prof.Etsuji Tomita and Dr.Yoshiaki Okubo. The work of Maximal Clique Sets presented in this thesis is based on the amazing clique enumeration algorithm proposed by Prof.Tomita. In the research of Maximal Clique Sets, especially during the attendance of the conferences in Canada, I have received so much encouragement from Prof.Tomita. Dr.Okubo gave me courage in my everyday life in the laboratory. He is so kind that all the students enjoy the discussion with him. Many works in this thesis were done with his fully supports.

Finally, I give thanks to the Rotary Yoneyama Memorial Foundation (RYMF). The scholarships from RYMF allows me focus on my research and study in the final year of the doctoral course. I offer my appreciation to Mr.Motoaki Sasaki as well as all the people from

the Rotary Club of Sapporo. The attendance of the Rotary Club of Sapporo as a scholarship student was the first time that I learned about the Japanese society. In the third year of the doctoral course, which was the most difficult time, the warm words from Mr.Sasaki helped me move on immensely.

0

Introduction

0.1 BACKGROUND

Recently, with the development of the Internet, the World Wide Web has become a major information source in our daily lives. However, as the contents of the Web (e.g., the number of Web pages) increases, it is not practical for users to search the entire Web to find helpful information. Thus, many techniques and services have been proposed to assist users

to discover and explore the Web. Famous services are search engines, such as Google* and Bing†. Search engines allow users to search all Web pages by providing a query (i.e., keywords) and the search engines usually rank the search results. Pages that are considered likely to be useful will be awarded higher ranks so that users can discover helpful contents efficiently. In this thesis, we say these services help users to *discover* information.

Various recommendation systems are also widely used in online services. For example, Amazon‡ is an online shop website. Unlike physical shops with space limitations, Amazon can provide an almost unlimited number of products. Users cannot check all the products and may not know of the existence of products that might potentially attract them. Recommendation systems, therefore, help users to explore for unknown but interesting products. To assist users to find targets that may interest them, the recommendation systems make use of a user's information (e.g., browsing history, wish list, or previous purchases) to predict the user's interests and provide a list of products that users may wish to buy. Amazon shows recommended products in Web pages while a user is browsing. The user can simply click on the recommended products to explore for more related ones. In this case, the recommendation system assists users to *explore* for products that are new to them.

Note that, in real-world applications, recommendation systems and search engines are usually used together. Even though the users may not have clear images of their targets, search engines can still filter out content that is certainly not related to the user's intention. Even if the users clearly know about the target, they still may have blind spots. Thus, many services combine these two systems. In general, there are several basic points that should be

*www.google.com

†www.bing.com

‡www.amazon.com

considered for these systems:

- (Interest) Discovery results should be consistent with the user's interests or intentions.

Before the discovery process, we usually require users to provide a query giving some information about their interests. In the previous examples of search engines, we can say that the "keywords" represent part of the intention of users. It is a natural requirement that the systems should find results related to the users' interests. Users will be certainly unsatisfied if the discovery results contain too many items unrelated to their intention.

- (Helpfulness) Discovery results should be "helpful."

In this thesis, we say that a result is "helpful" when it provides new information to the user. The information is helpful only if it is closely related to the user's search query. However, if the system is only able to find this type of information, many users will not be satisfied. For example, when a user searches for TV series similar to "Friends," which is a famous comedy and romance series, "Friends Season 2" is a closely related result. Nevertheless, it is highly possible that the user already knows about "Friends Season 2" and will expect some other TV series. In this case, we say the TV series "The Big Bang Theory" is a helpful discovery result because it is also a comedy and romance show but contains more geek jokes.

- (Variety) Discovery results should accord with the user's request.

In many cases, the information provided by the user is not enough to specify the discovery target precisely. In other words, the user's intention is usually not described

completely by the initial query. Therefore, if the system narrows down the discovery results too much, we may miss the chance to discover the targets that the user is really interested in. From this viewpoint, the discovery targets should not only be related to the query but also varied enough to cover the user's possible intention.

Furthermore, discovery results are usually divided into clusters for the ease of user's selection. For example, Amazon divides products into categories so that users can sharpen their search intention by choosing a category. However, if the categories are inappropriate or contain too many products, the user's real interests will be hidden in these clusters. Thus, we prefer to divide the products into small or medium-sized local clusters rather than large ones.

To realize discovery and exploration systems that match these three points, the following problems should be considered.

- (P-View) How to represent user's interests or intention?

Most methods require users to summarize their requirements (i.e., interests or intention) in keywords. These keywords will greatly affect the quality of discovery results. However, it is often difficult for users to describe their requirements precisely by keywords. For example, when a user tries to buy new clothes, it may not be easy to describe the style. Some users may not have enough background knowledge about fashions to think of proper keywords. Thus, the system should allow the user to provide their intention in forms other than keywords.

- (P-Helpful) How to measure the helpfulness of the discovery results?

As the helpfulness depends on the user, there is no common definition of helpfulness. It is necessary to provide a good measure of helpfulness. This measure should be easy to adjust based on the user's requirements. Based on our previous discussion, because the discovery results should be both related to the query and varied, it is also important to keep a balance between helpfulness and variety of the discovery results.

- (P-largescale) How to handle large-scale data?

Recently, the problem of how to handle large-scale data has become much more important for real-world applications. To cope with large-scale data, it is necessary to consider the scalability of suggested algorithms.

0.2 CONTRIBUTIONS

In this thesis, we propose several discovery and exploration algorithms that try to solve the problems described in the previous section. In this section, we discuss the basic idea of how to handle those problems of discovery and exploration systems.

We propose a system with two components: a data compression subsystem and a discovery and exploration subsystem. The input data will first be compressed to medium or small size by the data compression subsystem. Our discovery and exploration systems can then work efficiently on the compressed data. The image of the whole system in our plan is shown in Figure 1. All the necessary techniques of these two subsystems are proposed, implemented, and tested in this thesis; however, an integrated system was not realized and remains as future work.

More precisely, the methods used in the data compression subsystem are based on pseudo-

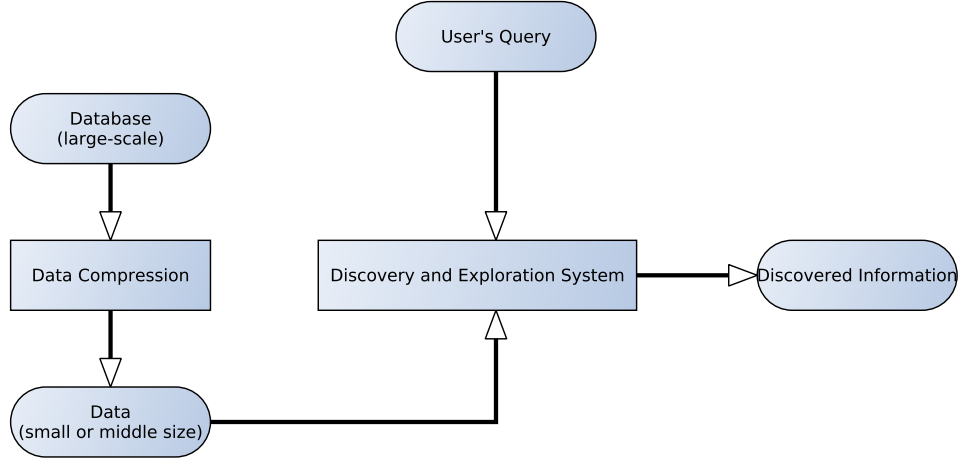


Figure 1: Image of the whole system

clique enumeration. Cliques and pseudo-cliques are popular models in graph theory to represent dense patterns as subgraphs. A clique is a subgraph in which all pairs of vertices are connected [BK73]. From this definition, cliques are the densest subgraphs. We choose pseudo-cliques for data compression because both small and medium-sized dense subgraphs can be represented by pseudo-cliques. As we have discussed previously, a user would be interested in specific small patterns as subgraphs. There are very many small and medium-sized pseudo-cliques. From the viewpoint of computational complexity, the task of enumerating even small pseudo-cliques is much more difficult than the task for large ones because there are so few large ones. In our research, we must consider the small and medium-sized pseudo-cliques to handle the user's various intentions.

Even though many other clustering and compression algorithms, such as k -means clustering [Mac67] and PCA [Jol86], may also find small or medium clusters if good parameters are chosen, most such algorithms suffer from the problem that the detected clusters are not necessarily dense. On the other hand, the pseudo-cliques discussed in this thesis are

required to be dense, and fast enumeration algorithms, even for small and medium-sized pseudo-cliques in large graphs, can be designed. For this reason, we consider that our approach based on pseudo-cliques is better than others.

In this thesis, our research focuses on a pseudo-clique model called k -plex [SF78]. k -Plex has wide applications in many fields, such as the discovery of protein complexes in protein–protein interaction networks [Geo+09; Cao+16], structural change pattern mining [OHT12; OHT13], and social network analysis [BBH11; PYB12; WP07; Du+07]. To apply the k -plex model to large-scale data, we must solve the following problems.

- The k -plex model and many other pseudo-clique models do not work well on large-scale data (the *scale problem*) [For10].
- k -plex may be sparse when its size becomes larger (the *sparsity problem*). Sparse pseudo-cliques are never our target.
- There exist many sparse k -plexes with relatively small sizes compared with k (the *size problem*). For this reason, we consider only nonsparse, i.e., dense, k -plexes that can be bases of data compression.

This thesis proposes two novel methods that cover the above problems. The general idea is to control the density of the k -plexes. The technical details are summarized as follows:

1. Method 1: Maximal Clique Sets (MCS), a pseudo-clique notion constructed from a set of cliques (Chapter 1).

The basic idea of this work is that, instead of finding a set of vertices like the traditional k -plex, we search for a subgraph that is a union of cliques. Because a union

of cliques cannot usually form a k -plex, we focus on the sets of cliques that match the requirement of a k -plex. We also require the cliques in a given set to overlap each other at some level. We call this subclass of k -plexes the Maximal Clique Set (MCS). As the MCS is defined as a family of overlapping cliques, we can avoid the sparsity problem of the k -plex. To improve performance on large-scale data further, we introduce a lower-bound size constraint on the cliques. The lower-bound size allows the MCS to work efficiently on data that contain too many small cliques. By adjusting the lower-bound size, we can also avoid the size problem of the k -plex. Even though overlapping cliques have been discussed previously in [HOo6], our proposal has several advantages, including better performance and preservation of the antimonotonicity property of the k -plex.

A related idea is the Greedy Clique Expansion (GCE) [Lee+10]. GCE is also based on the idea that a community should contain a clique as a dense core. A major difference between GCE and MCS is that GCE only requires the core to be a clique. The other components in the subgraphs detected by GCE may not be cliques. Technically, GCE first selects a clique as a core community and then adds more vertices to this core. During this process, GCE uses a predefined “community fitness” function as the measure to determine whether a vertex can be added or not. This “community fitness” in GCE works like the k -plex requirement in MCS. However, because GCE uses vertices instead of cliques as the components of the communities, the subgraphs detected by GCE may be sparser than those detected by our MCS. To investigate the quality of MCS, we compare our algorithms with GCE on the protein complexes’ prediction task in a Protein–Protein Interaction Network. The experimental results

show that our MCS algorithm is much better.

2. Method 2: (j, k) -maximal plex ((j, k) -MPC), a pseudo-clique notion with a lower-bound density (Chapter 2).

This approach introduces a lower-bound density on maximal k -plexes. This lower-bound density requires all the vertices in a maximal k -plex to have at least j connected edges. Because the naive pseudo-clique approaches, such as [Mok79; SF78] suffer from the sparsity problem, it is a straightforward idea to control the density. A larger lower bound will yield denser patterns. Technically, this constraint can be represented by the j -core [BZ03] model. Based on the j -core constraint, we can cut off undesirable k -plexes in the early stage of the enumeration, where we say that a k -plex is undesirable when it cannot be extended to any dense complex under the density constraint. These undesirable k -plexes are called hopeless k -plexes. We have theoretically proved that we can completely detect all the targets after cutting off those hopeless k -plexes. Because j -core works extremely efficiently on large data, we can ignore many sparse k -plexes during the enumeration process. As a result, our experiments show that this method works well on a dataset with about 900,000 vertices.

Recently, [Con+17] proposed an enumeration algorithm for large maximal k -plexes that also uses the j -core, but in a much more naive way than our (j, k) -MPC. That is, according to the properties of k -plex, all nodes in a k -plex with size m should have degrees at least $m - k$. A k -plex that is larger than m must be in j -cores where $j \geq m - k$. [Con+17] uses this simple fact to filter out those vertices that do not match the requirement of the k -plex. The j -core processing is thus performed once, beforehand, supposing that the targets are only large k -plexes. On the other hand, our algorithm

applies the j -core operation during the whole process of forming dense k -plexes, so it can find dense maximal k -plexes with sizes ranging from small to large.

We have discussed the pseudo-clique methods for representing the user's intention about the size and density requirements. We also consider the following two discovery and exploration problems for considering the user's intention. Before introducing the new problems, we must emphasize that pseudo-cliques can certainly act as data compression algorithms for the following problems. A simple approach is to use the collection of detected pseudo-cliques as input data so that we can compress the data without much loss of possible intention.

The discovery and exploration problems other than pseudo-clique considered in this thesis are summarized as follows.

- Associative search problem (Chapter 3).

In studies of information retrieval, an associative search [Tak+00] is sometimes used to modify a user's queries for documents based on the documents–words relation.

The reason for modifying queries is that the initial queries may not properly express the user's intention. To solve this problem, associative search allows a user to control the direction of the change by selecting words (or documents) according to their intention during the search process. In our research, unlike the traditional associative search, we let the user select mediators, people who write documents over several domains of interest. These mediators can direct the user to targets that the user was not aware of but may find of interest. In comparison, when selecting words or documents in our daily life, we often obtain information from people who talk about

many topics, some of which are also concerned with the present retrieval task. We can formalize this problem [GW12]. The search process can be technically described as follows. First, according to the user’s initial queries, we generate a list of people by building a people–words concept lattice and the user selects several people. Based on the user’s selection, we find a bridge concept that generalizes the user’s intention. Finally, our system generates some concepts that are conditionally similar to the initial one, given the bridge concept. We show that this algorithm can find some interesting results in Twitter data.

In real-world applications, an associative search is usually used as a technique for recommendations. However, in contrast to existing common recommendation techniques, such as collaborative filtering [BHK98] and content-based filtering [IFO15], our approach has the advantage that users can explicitly control the direction of the recommendations by selecting mediators who share the same or similar interests as themselves.

- Feature association discovery problem (Chapter 4).

This problem considers the user’s intention about similar or analogous relationships between features over two datasets. For example, in word–document data, the words can be treated as features and the documents are called objects. Because users have their own understanding of words, different users may assume different similarities between words. To represent the users’ intentions, they provide associations of features over two datasets, where two features are associated when the user regards them as similar or analogous. These associations provided by the user are called “hints.” In the field of analogy [Gre85], association discovery is closely

related to the analogical reasoning problem. We formalized the feature association discovery problem under the framework of NMF [LS99]. We use the nonnegative trifactorization technique [LZY05] to project the features from two datasets to vectors with the same dimensions. Nonnegative trifactorization is designed to cluster the rows and columns of the input data simultaneously by exploiting the duality between rows and columns [LZY05; DP15]. As we consider dyadic data, trifactorization works much better than the traditional NMF in our problem. To construct a common space that reflects the user’s intention, we introduce the constraint that the associated features provided by the user must have similar vector representations in the common space. By embedding all the other features from the two datasets into this common space, we derive new representations of features under the user’s intention. As a result, we can infer the feature associations based on their closeness in the common space. There are also many features that are not related to any of the user’s hints. According to the properties of trifactorization, these unrelated features may be embedded near the features that interest the user. The interference from these unrelated features will have a negative effect on our inference. To deal with the unrelated features in the common space, we introduce Laplacian regularization terms to retain the similarities in the original datasets.

If we consider text datasets from two different languages as the inputs, our feature association discovery algorithm can be used to solve the bilingual word-embedding problem [Zou+13; VM15]. This technique aims at representing words from two languages as vectors in a common space, where words with similar meanings are to be placed closely in that space. By giving pairs of words and their translations as

hints, our algorithm can be used to solve this type of problem. To the best of our knowledge, this work is the first to apply NMF techniques to the bilingual word-embedding problem. We also validate the quality of our algorithm by some experiments with Spanish and Italian news datasets. We use the Google translation service to construct an answer set to evaluate the experimental results.

0.3 THESIS ORGANIZATION

The rest of this thesis is organized as follows: First, we discuss the data compression algorithms. Chapter 1 introduces the technique for enumerating MCSs. In Chapter 2, the details of (j, k) -MPC are discussed. We introduce the methods of the exploration and discovery subsystem in Chapter 3 and Chapter 4. Chapter 3 proposes the associative search methods. The technique introduced in Chapter 4 focuses on finding associations based on the user's intention. Finally, Chapter 5 contains the conclusions and future work for the whole system.

*This work is originally published at 18th International
Conference on Discovery Science (DS 2015) [Zha+15]
and licensed for Thesis/Dissertation use by Springer.
License Number: 4286930255741.*

1

Enumerating Maximal Clique Sets with Pseudo-Clique Constraint

1.1 INTRODUCTION

A social network is often represented by a graph of vertices representing actors in a social network. The edges represent ties between actors [CSW05]. The actors are typically people,

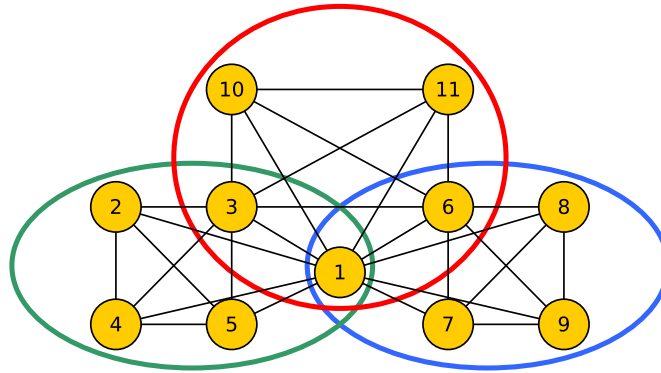
and the ties are their relationships as friendship, the collegiality, the acquaintance, or other types of associations. In recent years, with the increasing availability of network data, the structures of networks have drawn great interest. A key feature of networks is that vertices are often organized into *communities* [GN02]. As a formal model of communities, a clique notion has been used from the beginning of study [BK73]. A clique satisfies three important properties of communities, familiarity, reachability, and robustness. These properties are represented by vertex degree, path length, and connectivity, respectively in graph theoretic terms [MH12].

However, the clique approach has been criticized for its restrictive nature for analysis of real data [SF78]. Most communities are in fact not cliques, allowing exceptional disconnection between actors. This motivates us to study clique relaxation models [PYB12]. Luce [Luc50] has proposed a distance-based model called k -clique and Alba [Alb73] has introduced a diameter-based model called k -club. These models have also been studied along with a variant called k -clan introduced in [Mok79]. In a word, the major parameter k of these models controls admissible distances among vertices. As is well known by the study of Small World Network [WS98], when we allow a longer distance parameter, large dense subgraphs appear which are almost cliques even when the subgraphs w.r.t. the original edge connection are not dense. On the other hand, k -plex model [SF78; WP07] takes the density w.r.t. the original graph into account. A vertex set X is called a k -plex if, for any vertex v in X , the number of vertices not connected to v is at most k including v itself. We also concern in this research the density of vertex sets in the original graph. However, the definition of k -plex is clearly weak in detecting dense vertex sets whose sizes are not small. As we target larger dense vertex sets, more number of missing edges as exceptions must be allowed in the

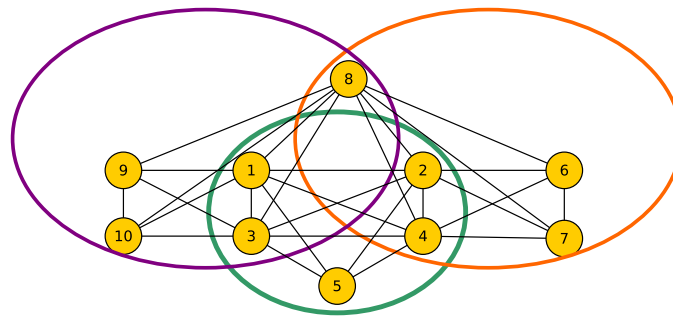
set. This means that we have to set larger k accordingly. Since every vertex set with its size less than $k + 1$ is trivially a k -plex, there exist many trivial k -plexes for such a k . Moreover, k -plexes with their sizes exceeding k are not necessarily densely connected. These simple facts prevent us from designing efficient k -plex enumerators for densely connected vertex sets.

A hint to overcome this difficulty would be found in the study of communities and networks. Various indices for evaluating clusters of vertices have been proposed to show how the vertices in a community cluster together. A clustering coefficient [WS98] is often used as such an index. The more number of small cliques (i.e. triangles) in a cluster, the higher the index value will be. Some of those small cliques are directly connected via overlapped common vertices. In this research, we focus on finding clusters consisting of cliques overlapping each other to keep the density as a whole. We call such a cluster a meta-clique in a graph of cliques in the original graph. As a vertex set, the cluster is simply a set union of component cliques. The cluster thus defined may not be a k -plex when the component cliques less overlap. We require that the clusters must be k -plexes, and call them k -clique sets. We also use a set-theoretic correlation measure, similar to the bond used for transaction databases [Omio3], to guarantee the degree of connection among component cliques. Typical meta-cliques targeted in this research are illustrated in Figure 1.1. Figure 1.1a illustrates the case that all the cliques in the Meta-clique shares one set of vertices, while Figure 1.1b shows the case that the Meta-clique contains more than one cores.

A notion of overlapping cliques has already been studied in [HO06], where its search process is much complicated. The procedure firstly tries to enumerate the intersections called cores, and then try to find out member cliques by adding additional vertices to the cores. Thus, the process invokes additionally clique a enumerator whenever each candidate



(a) single core: $\{1\}$



(b) several cores: $\{1, 3\}$, $\{2, 4\}$ and $\{8\}$

Figure 1.1: Target Meta-clique

core is found. On the other hand, the procedure *MetaClique* developed in this research firstly enumerates possible cliques that can be member cliques and their connections constrained by the bond measure beforehand. Then the remaining task is simply computing cliques in the clique graph under k -plex constraint. This reduces the overall computational cost drastically. Since the clique enumeration is fast enough for sparse graph as has been demonstrated in [ES11], *MetaClique* for finding overlapping cliques is working quite well even for medium k , as we demonstrate by several experiments in Section 1.5.

In our experimentation for several synthetic and real world networks, including large-scale networks with more than 1,000,000 vertices, we compare with an extended maximal k -plex enumerator and verify the effectiveness of our approach. We show that *MetaClique* works efficiently for various types of data. As a remarkable point, the number of solutions by *MetaClique* could be a useful index for characterizing networks because those numbers are not strongly affected by the parameter k . For a protein-protein interaction network, we also evaluate the quality of extracted communities by F_1 -score defined with recall and precision for a given answer set. For each parameter settings, our *MetaClique* shows the highest score among those by existing community detectors for PPI networks. Particularly, we observe that *MetaClique* gives the best score for a parameter setting which it is hard for GCE [Lee+10], known as the most effective system, to detect solutions with high quality. This implies that our *MetaClique* has an ability to detect many meaningful communities missed by GCE.

The remainder of this research is organized as follows. In the next section, we introduce some terminologies used throughout this research. In Section 3, we define our problem of enumerating overlapping cliques as meta-cliques. Our algorithm *MetaClique* for the

problem is presented with a pseudo-code in Section 4. Section 5 presents our experimental results for several networks. In Section 6, we conclude this research with a summary and future work.

1.2 PRELIMINARIES

A *simple* graph is denoted by $G = (V, E)$, where V is a set of vertices and $E \subseteq V \times V$ a set of undirected edges. For a pair of vertices, $v, v' \in V$, v is said to be adjacent to v' if $(v, v') \in E$. For a vertex v , the set of vertices adjacent to v is denoted by $N_G(v)$, that is, $N_G(v) = \{v' | v' \in V \wedge (v, v') \in E\}$, where $|N_G(v)|$ is called the *degree* of v in G and is referred to as $deg_G(v)$. If any pair of vertices $v, v' \in V (v \neq v')$ are adjacent each other, then G is said to be *complete*.

For a graph $G = (V, E)$, a *subgraph* of G induced by $V' \subseteq V$, denoted by $G[V']$, is defined as $G[V'] = (V', E \cap (V' \times V'))$. In the subgraph $G[V']$, the number of edges incident to a vertex v , called degree, is denoted by $deg_G[V'](v)$. A complete subgraph is called a *clique* in G . We often refer to a clique $G[V']$ as simply V' . A clique is said to be *maximal* if it is not a proper subset of any other cliques. For a graph G , the set of maximal cliques is denoted by the set $C(G) = \{c_1, c_2, \dots, c_m\}$. The problem of enumerating all the maximal cliques is called “*Maximal Clique Problem*”. A maximal clique with the largest size is especially called a maximum clique. It should be noted that in general, a maximum clique is not uniquely in G .

As previously discussed, although a clique is a perfect structure for community, it is difficult to find them in real world. Therefore, a notion of k -plex has been introduced as one of the relaxation models [SF78].

Definition 1 (*k*-plex) For a set of vertices $S \subseteq V$, the subgraph $G[S]$ is called a *k*-plex if

$$\deg_{G[S]}(v) \geq |S| - k \text{ for every vertex } v \text{ in } S. \quad \blacksquare$$

A *k*-plex $G[S]$ is also referred to as simply S .

Similar to the case of cliques, a *k*-plex is said to be *maximal* if it is not a proper subgraph of any other *k*-plexes. A *k*-plex is regarded as a pseudo-clique in the sense that it can be obtained by deleting *less than k*-edges from each vertex in a clique. Thus, a clique is a special case of *k*-plex with $k = 1$. We should point out that Definition 1 does not emphasize the requirement of connectivity. The problem of enumerating all *k*-plexes is called “Maximal *k*-plex Problem”. From the definition, it is clear to see that *k*-plex is anti-monotonic, that is, any subset of a *k*-plex is also a *k*-plex.

1.3 *k*-MAXIMAL CLIQUE SET

As has been mentioned, existing algorithms for enumerating maximal *k*-plexes will detect a large amount of sparse patterns and work well with only relatively small *k* values. In this section, we present a new subclass of *k*-plexes to address this problem.

1.3.1 *k*-MAXIMAL CLIQUE SET : *k*-MCS

The new class of pseudo-clique we define is called a *k*-Maximal Clique Set (*k*-MCS). It is formally defined as follows.

Definition 2 (*k*-Maximal Clique Set) Given a graph G , let $Comp$ be a set of maximal cliques in G , where each $c \in Comp$ is considered as a primitive component. Then, a set of components $T \subseteq Comp$ is called a *k*-MCS (Maximal Clique Set) iff $\bigcup_{c_i \in T} c_i$ is a *k*-plex in G . ■

A k -MCS T is defined as a set of component maximal cliques whose union forms a k -plex in G . In what follows, the k -plex defined as the union of components in T is referred to as $Ext(T)$, that is, $Ext(T) = \bigcup_{c_i \in T} c_i$.

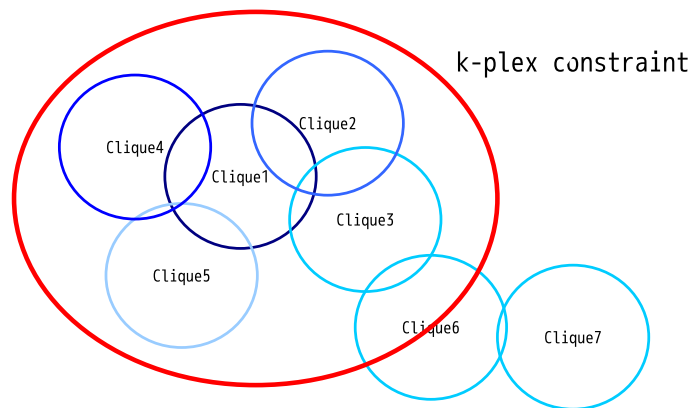
Roughly speaking, by combining maximal cliques, we can expect to efficiently obtain a densely connected community with reasonable cost, because several practical algorithms for enumerating maximal cliques are recently available [TTT06].

From the anti-monotonicity of k -plex, we are particularly concerned with a *maximal* k -MCS, that is, a *maximal set of components* whose union can give a k -plex. It should be noted here that in some cases, we could have different maximal k -MCSs T and T' such that $Ext(T) = Ext(T')$.

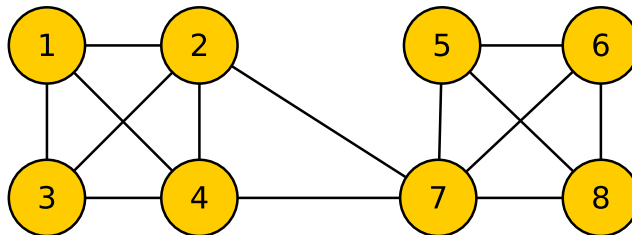
Figure 1.2a shows a simple example of k -MCS. In the figure, c represents the term “clique”. $\{c_1, c_2, c_3, c_4, c_5\}$ is a maximal k -MCS consisting of five components.

We have to point out that in some cases, different k -MCSs may have the same vertex set. For example, in Figure 1.2b, there are three maximal cliques: $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$ and $\{2, 4, 7\}$. Even though k -MCS $\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}$ and $\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{2, 4, 7\}\}$ contain totally the same vertices, since each clique is one primitive component, $\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}\}$ and $\{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{2, 4, 7\}\}$ are two different k -MCSs. It is clear that maximal k -MCS can not be duplicate.

From now on, we use the term “ k -MCS” as the meaning of “maximal k -MCS” in a given graph G unless otherwise stated.



(a) Image of maximal k -MCS



(b) Example of k -MCS that may be duplicate

Figure 1.2: Image of maximal k -MCS

1.3.2 RESTRICTING k -MCS WITH SIZE AND OVERLAPPINGNESS OF COMPONENTS

In Definition 2, each k -MCS is required to form a k -plex. Although the constraint can exclude many useless combinations of components which result in obviously sparse communities, we often have undesirable k -MCSs when we assume a relatively larger k value. For example, we could obtain a k -MCS whose components have little shared vertices. Needless to say, such a community would not be so meaningful because it is not reasonable to combine components with low commonality into one group. Moreover, in many real world networks, there exist a large amount of small maximal cliques consisting of just a few vertices. However, those small cliques (components) exponentially increase the number of useless combinations, making computational efficiency worse. Therefore, we impose the following additional constraints on size of component maximal cliques in order to exclusively find densely connected pseudo-cliques with certain amount of size.

More precisely speaking, given a graph G and a positive integer *minsize*, we consider

$$C_{minsize}(G) = \{c \mid c \text{ is a maximal clique in } G \text{ such that } |c| \geq minsize\}$$

as our set of components and try to find maximal k -MCSs consisting of components in $C_{minsize}(G)$. The constraint on component size can effectively exclude sparse communities mainly consisting of component cliques with small size.

In addition to the constraint, we take overlappingness of components in a maximal k -MCS into account. That is, in order to obtain densely connected communities, we require our maximal k -MCS to consist of components overlapping each other. Formally speaking, we evaluate our degree of overlappingness by *bond measure* [Omi03] which is regarded as

an extension of *Jaccard Coefficient*.

Given a pair of sets A and B , the bond value for A and B , $Bond(A, B)$, is simply defined as

$Bond(A, B) = \frac{|A \cap B|}{|A \cup B|}$. From the definition, it is easy to see that if A and B are highly overlapping, then the bond value $Bond(A, B)$ is close to 1.0.

1.3.3 PROBLEM OF FINDING MAXIMAL k -MCS

From the above discussion, we can now define our problem of enumerating maximal k -MCSs in a given graph.

Definition 3 (*Maximal k -MCS Problem*) *Let G be a graph, $minsize$ a minimum size of components and $minbond$ a threshold for minimum bond value. Then, a Maximal k -MCS Problem is to find every maximal set of components $T \subseteq C_{minsize}(G)$ satisfying the following conditions, where*

$$C_{minsize}(G) = \{c | c \text{ is a maximal clique in } G \text{ such that } |c| \geq minsize\}.$$

- $Ext(T) = \bigcup_{c_i \in T} c_i$ is a k -plex in G .
- For any pair of c_i and c_j in T , $Bond(c_i, c_j) \geq minbond$.

■

In the next section, we present our algorithm for this problem.

1.4 ENUMERATING MAXIMAL k -MAXIMAL CLIQUE SET

In order to enumerate all maximal k -MCSs for given $minsize$ and $minbond$, we have to exhaustively examine possible combinations of components in $C_{minsize}(G)$. For efficient com-

putation, we first introduce a notion of *k-Clique Graph* by which we can exclude quite useless combinations of components in our search process.

1.4.1 *k*-CLIQUE GRAPH

Our maximal *k*-MCS T to be detected must consist of pairwise overlapping components whose union forms a *k*-plex. This implies that any pair of components in T , c_i and c_j , gives a sufficient bond value and the union $c_i \cup c_j$ forms a *k*-plex from the monotonicity of *k*-plex. Let us here consider an auxiliary graph $\tilde{G}$ in which each vertex is a component in $c \in C_{minsize}(G)$ and an edge means the connected vertices (that is, components) give a sufficient bond value and form a *k*-plex. Then, any maximal *k*-MCS can be extracted as a clique in $\tilde{G}$. That is, in order to enumerate our maximal *k*-MCSs, it is sufficient to examine cliques in $\tilde{G}$. Thus, the auxiliary graph $\tilde{G}$ can provide useful information about candidate combinations of components we have to check. Since each vertex in $\tilde{G}$ is a component maximal clique for our problem, we call $\tilde{G}$ a *k-clique graph* which is regarded as an extension of clique graph [Ham68]. It is formally defined as follows.

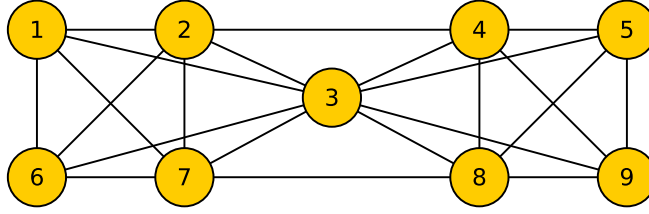
Definition 4 (*k-Clique Graph*) For a given graph G , k , $minsize$ and $minbond$, a graph

$\tilde{G}_{minsize, minbond}^k = (\tilde{V}, \tilde{E})$ is called the *k-clique graph* for the maximal *k*-MCS problem, where

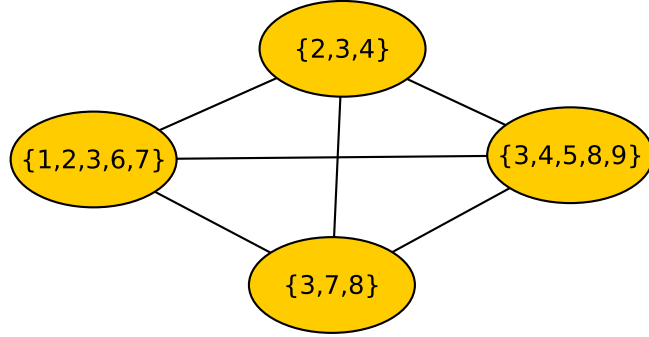
$$\tilde{V} = C_{minsize}(G) = \{c \mid c \text{ is a maximal } k\text{-plex in } G \text{ such that } |c| \geq minsize\}$$

$$\tilde{E} = \{(c_i, c_j) \mid c_i, c_j \in \tilde{V}, Bond(c_i, c_j) \geq minbond \text{ and } (c_i \cup c_j) \text{ is a } k\text{-plex}\}$$

■



(a) Input Graph G



(b) 4-Clique Graph of G

Figure 1.3: k -Clique graph

Figure 1.3 illustrates an example of k -Clique Graph. Figure 1.3a shows the input graph which contains 4 maximal cliques: $\{1, 2, 3, 6, 7\}$, $\{2, 3, 4\}$, $\{3, 7, 8\}$ and $\{3, 4, 5, 8, 9\}$. Figure 1.3b presents the k -clique graph with $k = 4$. From this example, it is easy to know that the k -clique graph can often simplify the input graph.

As is just mentioned above, since a maximal k -MCS can be detected as a clique in $\tilde{G}_{minsize, minbond}^k$, that is, a clique of component cliques, we often call it a *Meta-Clique*.

With the help of the k -clique graph $\tilde{G}_{minsize, minbond}^k$, we can efficiently extract all maximal k -MCSs. We discuss below our algorithm in details.

1.4.2 BASIC SEARCH STRATEGY

Given a graph G , k , $minsize$ and $minbond$, we first identify the set of components $C_{minsize}(G)$ which consists of all maximal cliques in G with size no less than $minsize$. We then construct the k -clique graph, $\tilde{G}_{minsize, minbond}^k$, by examining whether each pair of components in $C_{minsize}(G)$ have a sufficient degree of overlap and forms a k -plex. Finally, we try to enumerate (meta-)cliques in $\tilde{G}_{minsize, minbond}^k$ and check their maximality and k -plexness. Our process of enumerating meta-cliques is based on CLIQUES [TTTo6], an excellent algorithm for listing all maximal cliques. As a basic procedure, in our algorithm, a meta-clique X is tried to expand by adding a component $c \in Cand(X)$ to obtain a larger meta-clique $X \cup \{c\}$, where $Cand(X)$, called the candidate set for X , is the set of components each of which is adjacent to all components of X in $\tilde{G}_{minsize, minbond}^k$. Starting with $X = \emptyset$ and $cand(X) = C_{minsize}(G)$, such an expansion process is iterated in depth-first manner until no meta-clique can be expanded. At each expansion step, the meta-clique X is examined whether the union of all components in X (that is, $Ext(X)$) forms a k -plex or not. If it is true, X is tried to further expand. On the other hand, if it is false, we can safely stop expanding X and then backtrack to the next alternative because any expansion of X can never be our solution from the monotonicity of k -plex.

1.4.3 PRUNING USELESS SEARCH BRANCHES

As has been mentioned, for a pair of component sets X and X' , we often observe $Ext(X) = Ext(X')$. In our search process, for a meta-clique X , if the candidate set $Cand(X)$ includes a component c such that $c \subset Ext(X)$, expanding X with c is quite useless because we must have $Ext(X) = Ext(X \cup \{c\})$. In other words, we do not need to *actually* examine the

search branch corresponding to such an expansion. Just adding c to X is sufficient in our search process.

Based on this idea, at each expansion step for X , we first detect a meta-clique $M \subseteq Cand(X)$ defined as $M = \{c \mid c \in Cand(X) \text{ and } c \subseteq Ext(X)\}$. From the property of k -plex, we know that any subset of a k -plex is still a k -plex. As we require $c \subseteq Ext(X)$, it is clear that $M \subseteq Ext(X)$. Thus, M is a meta-clique. With the meta-clique M , we then try to expand $(X \cup M)$ with a component in $Cand(X \cup M)$. As the result, we can prune $|M|$ search branches from the search node of X without loss of completeness.

1.4.4 ALGORITHMS

Summarizing the above discussion, we present an algorithm for enumerating all maximal k -MCSs based on k -clique graph as well as several pruning strategies. A pseudo-code is presented in Algorithm 1.

Procedure EnumerateClique presented in Algorithm 2 identifies the set of primitive components (maximal cliques in G) with size no less than *minsize*. In Procedure EnumerateClique, we firstly remove all the vertices whose degree are less than *minsize* and then call Procedure SCLIQUES to enumerate all the cliques with size no less than *minsize*. With a set of primitive components, Procedure BuildKCliqueGraph then builds a k -clique graph. Finally, Procedure EnumerateKMCS tries to detect all maximal k -MCSs. All pruning mechanisms presented above are incorporated into the algorithms.

Algorithm 1 Algorithms for finding maximal k -MCSs

```

1: procedure MAIN( $G, k, j, b$ )                                 $\triangleright G = G(V, E)$  is the input graph
2:    $C \leftarrow \text{ENUMERATECLIQUE}(G, j)$                      $\triangleright$  Enumerate maximal cliques larger than  $j$ 
3:    $F \leftarrow \text{BUILDKCLIQUEGRAPH}(C, G, k, b)$ 
4:   return  $\text{ENUMERATEKMCS}(F, G, k)$ 
5: end procedure

```

```

1: procedure BUILDKCLIQUEGRAPH( $C, G, k, b$ )
2:    $V_F = C$ 
3:    $E_F = \emptyset$ 
4:    $N_F = \{n | n = |c|, c \in C\}$ 
5:    $M_F = \emptyset$ 
6:    $F = F(V_F, E_F, N_F, M_F)$                                  $\triangleright$  Initialize  $k$ -clique graph
7:   for each  $c \in C$  do
8:     for each  $c' \in C \setminus \{c\}$  do
9:       if  $\text{bond}(c, c') > b$  and  $c \cup c'$  is  $k$ -plex in  $G$  then
10:         $E_F \leftarrow E_F \cup \{(c, c')\}$                      $\triangleright$  Add edge
11:         $M_F \leftarrow M_F \cup \{|c \cap c'|\}$                  $\triangleright$  Calculate overlap of cliques
12:      end if
13:    end for
14:  end for
15:  return  $F$ 
16: end procedure

```

```

1: procedure ENUMERATEKMCS( $F, G, k$ )
2:    $C \leftarrow V(F)$                                  $\triangleright$  Initial candidate set takes all vertices of  $F$ 
3:    $comp \leftarrow \emptyset$ 
4:    $not \leftarrow \emptyset$ 
5:    $\text{DOENUMERATEKMCS}(comp, C, not)$ 
6: end procedure

```

```

1: procedure DOENUMERATEKMCS(comp, C, not)
2:   if  $C = \emptyset$  and  $not = \emptyset$  then
3:     Print comp ▷ Maximal  $k$ -MCS detected
4:   end if
5:   while  $C \neq \emptyset$  do
6:      $u \leftarrow$  one of the largest cliques in  $C$ 
7:      $C \leftarrow C \setminus \{u\}$ 
8:      $not \leftarrow not \cup \{u\}$ 
9:      $C_{connected} \leftarrow N(u) \cap C$ 
10:     $not_{connected} \leftarrow N(u) \cap not$ 
11:     $C_{kmcs} \leftarrow \emptyset$ 
12:     $not_{kmcs} \leftarrow \emptyset$ 
13:    for each  $d \in C_{connected}$  do
14:      if  $d \cup comp$  is a  $k - plex$  then
15:         $C_{kmcs} \leftarrow C_{kmcs} \cup \{d\}$ 
16:      end if
17:    end for
18:    for each  $d \in not_{connected}$  do
19:      if  $d \cup comp$  is a  $k - plex$  then
20:         $not_{kmcs} \leftarrow not_{kmcs} \cup \{d\}$ 
21:      end if
22:    end for
23:     $D_{bridge} \leftarrow$  the set of all bridges in  $C_{kmcs}$ 
24:    for each  $r$  in ENUMERATECLIQUE( $G[D_{bridge}], 1$ ) do
25:      DOENUMERATEKMCS( $comp \cup \{u\} \cup r$ ,  $C_{kmcs} \cap N(r)$ ,  $not \cup N(r)$ )
26:    end for
27:  end while
28: end procedure

```

Algorithm 2 Enumerate Maximal Cliques with size constraint

```

1: procedure ENUMERATECLIQUE( $G, j$ )                                 $\triangleright j$  is the size constraint
2:    $C \leftarrow \emptyset$ 
3:   for each  $c \in V(G)$  do                                        $\triangleright V(G)$  is all the vertices of  $G$ 
4:     if  $|c| \geq j$  then
5:        $C \leftarrow C + c$ 
6:     end if
7:   end for
8:   return SCLIQUES( $\emptyset, c, \emptyset, G, j$ )
9: end procedure

```

```

1: procedure SCLIQUES( $comp, C, not, G, j$ )
2:   if  $|comp| + |c| < j$  then                                      $\triangleright$  Hopeless to generate cliques larger than  $j$ 
3:     return
4:   end if
5:   if  $C = \emptyset$  and  $not = \emptyset$  then
6:     if  $|comp| > j$  then
7:       Output  $comp$ 
8:     end if
9:     return
10:  else
11:     $u \leftarrow \text{CHOOSEPIVOT}(C, not)$ 
12:    for each  $v \in C - N(u)$  do
13:       $C \leftarrow C - v$ 
14:       $not \leftarrow not + v$ 
15:      if  $|comp| < j$  then
16:         $C_{new} \leftarrow \text{JCORE}(C \cap N(c), j - |comp|)$ 
17:      else
18:         $C_{new} \leftarrow C \cap N(c)$ 
19:      end if
20:      SCLIQUES( $comp \cup \{v\}, C_{new}, not \cap N(u)$ )
21:    end for
22:  end if
23: end procedure

```

```

1: procedure JCORE( $C, j$ )
2:   while Any vertex  $u$  in  $C$  that  $|N(u) \cap C| < j$  do
3:     for each  $c \in C$  do
4:       if  $|N(c) \cap C| < j$  then
5:          $C \leftarrow C - c$        $\triangleright$  Remove vertices who connect less than  $j$  vertices in  $C$ 
6:       end if
7:     end for
8:   end while
9:   return  $C$ 
10: end procedure

```

```

1: procedure CHOOSEPIVOT( $P, X$ )
Require:  $V_{max} \in P \cup X$ 
2:    $n_{max} \leftarrow 0$ 
3:   for each  $u \in P \cup X$  do
4:     if  $n_{max} < |N(u) \cap P|$  then
5:        $n_{max} \leftarrow |N(u) \cap P|$ 
6:        $V_{max} \leftarrow u$ 
7:     end if
8:   end for
9:   return  $V_{max}$ 
10: end procedure

```

1.5 EXPERIMENTAL RESULTS

We present our experimental results in this section. Because our method, referred to as *MetaClique*, points to find a subclass of k -plex, we here compare our method against a maximal k -plex enumerator *MaxKPlex*.

MaxKPlex tries to enumerate maximal k -plexes except obviously undesirable and trivial ones. More concretely speaking, the task of *MaxKPlex* is to extract all *non-pass connected k -plexes each of which contains more than k vertices*.

In order to compare *MetaClique* and *MaxKPlex* under similar conditions, *MaxKPlex* is imposed an additional constraint on size of solution maximal k -plexes to be extracted, because size of k -plexes found by *MetaClique* is at least *minsize*, the minimum size of components. It should be noted here that the additional constraint on size is beneficial for computational efficiency of *MaxKPlex* because its branch-and-bound pruning can work more powerfully.

1.5.1 DATASETS

We observe computational performance of *MetaClique* compared with *MaxKPlex* for various benchmark graphs, including synthetic random graphs and real world networks.

Our random graphs have been generated by the method described in [VL05]. We have selected some degree distributions under power law as well as average degrees to generate different types of graphs. The number of vertices of random graphs varies from 10,000 to 100,000.

CA-GRQC [LK14] is a collaboration network constructed from the e-print arXiv and covers scientific collaborations among authors whose papers submitted to General Relativ-

ity and Quantum Cosmology category.

GEOM-o [BMo6] is an authors' collaboration network produced from Computational Geometry Database *geombib*.

COM-AMAZON [LK14] is based on the feature *customers who bought this item also bought* provided by Amazon website. It is one of the large scale datasets to test the computational performance of our algorithm.

YOUTUBE [LK14; YL15] is another large social network based on the friendships between the users of the video-sharing web site Youtube.

Moreover, in order to verify ability of *MetaClique* to identify meaningful communities, we have made an experiment on a protein-protein interaction network referred to as PPI [Col+07] and compared *MetaClique* with existing community detectors for PPI networks.

Detailed information of those networks are summarized in Table 1.1.

Table 1.1: Information of Graphs

Name	# of Vert.	# of Edges	Density	ACS*	MCS**
RANDOM-10000-1	10,000	40,497	0.00081	2.3	4
RANDOM-10000-2	10,000	95,474	0.00191	3.3	6
RANDOM-50000-1	50,000	949,155	0.00038	2.5	5
RANDOM-50000-2	50,000	1,081,876	0.00043	3.0	6
RANDOM-100000-1	100,000	1,914,690	0.00019	2.3	4
RANDOM-100000-2	100,000	2,159,964	0.00021	2.7	5
GEOM-o	7,343	11898	0.00044	3.1	22
CA-GRQC	5,242	14484	0.00105	3.1	44
COM-AMAZON	334,863	925,872	0.00002	2.9	7
YOUTUBE	1,134,890	2,987,624	0.000002	3.6	17
PPI	1,622	9,070	0.00690	17.9	33

* Average Clique Size

** Maximal Clique Size

1.5.2 COMPUTATIONAL PERFORMANCE

We observe computation times and numbers of solutions detected by *MetaClique* and *MaxKplex*. For each graph, we have executed both systems with *minsize* of the range from 3 to 7 and various *k*-values.

Figure 1.6 to Figure 1.9 show the computation times for each graph. In these figures, missing data points mean that the algorithms have failed to return within 3 hours. Computation times by *MetaClique* include those for enumerating component maximal cliques and constructing *k*-clique graphs. For each graph, however, it takes less than 1% of the total.

For $k = 2, 3$, both *MetaClique* and *MaxKplex* achieve good efficiency. However, when *k* becomes larger than 3, our *MetaClique* greatly outperforms *MaxKplex*. Particularly, for CA-GRQC, *MetaClique* can detect solutions 100,000 times faster than *MaxKplex*.

Figure 1.16 to Figure 1.20 present the computation times for large graphs. Because of the scale of graphs, *MaxKplex* fails to give results in limited time in all the cases. Thus, the computation times of *k*-plex are emitted. When *j* is small, *MetaClique* sometimes fails to finish the enumeration. The reason is that these graphs contain too much small cliques. For example, there exists 492,076 3-vertices maximal cliques in YOUTUBE data. However, when *j* becomes larger, the number of primitive components of meta-cliques reduces greatly. For YOUTUBE data, there are only 862,273 maximal cliques containing more than 5 vertices while the total number of cliques is 3,265,955. As the result, by setting a proper parameter *j*, *MetaClique* can work efficiently for large graphs under large *k*. One may criticize that *MetaClique* may miss too much possible communities under large *j*. However, for real-world graphs, we find that most of the vertices can be covered by only using large cliques. Figure 1.4 is one typical example. Figure 1.4 presents the distribution of clique

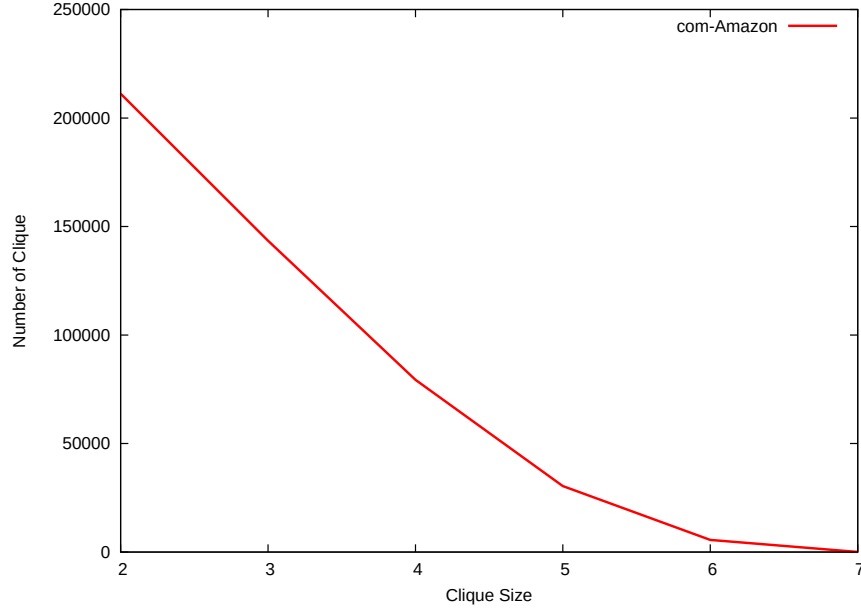


Figure 1.4: Distribution of clique size for com-Amazon

size of COM-AMAZON graph. Figure 1.5 shows that how many vertices can be covered when only using maximal cliques containing more than j vertices as primitive components. From the figure, we can see that only 55% maximal cliques are larger than 4. However, these 55% maximal cliques cover more than 85% vertices of the graph. Thus, under a proper j , *MetaClique* can efficiently detect most significant communities.

In Figure 1.10 to Figure 1.13, we also present numbers of solutions. From the figure, *MetaClique* extracts a much less number of solutions than *MaxKplex* for each graph. It is one of the reasons for high computational performance by *MetaClique*. One might here claim that a k -plex as the union of some maximal cliques is too restrictive and we would be missing many interesting communities. As we will see later, however, *MetaClique* can actually detect many meaningful communities with sufficient quality. This means that from

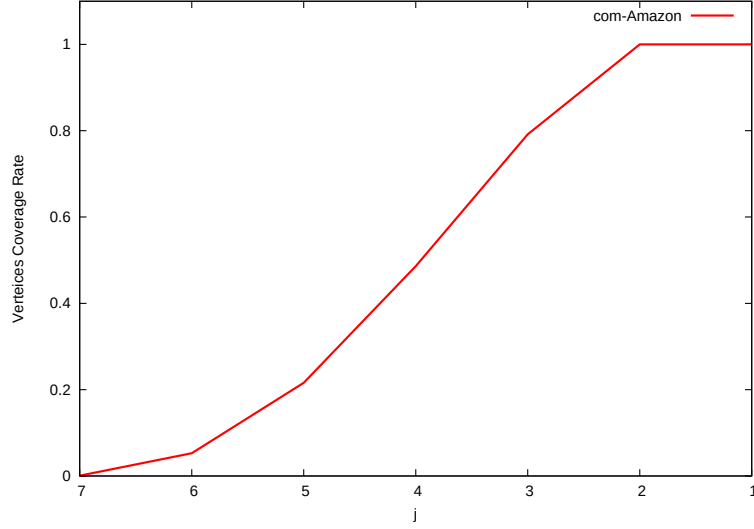


Figure 1.5: Vertices coverage rate for com-Amazon

the standpoint of *MetaClique*, many maximal k -plexes found by *MaxKPlex* are sparse and should not be considered as communities. In addition, since average vertex degree is usually larger than average clique size, when j value is smaller than average vertex degree, *MetaClique* can get more performance improvement than k -plex by increasing j . As a remarkable point the authors would like to emphasize, the figures show that numbers of solutions by *MetaClique* are not strongly affected by the parameter k . For each j as *minsize*, we have almost the same numbers of solutions even for different k values. This means that the number of solutions by *MetaClique* could be a useful index for characterizing networks.

Figure 1.14 and Figure 1.15 show the behavior of *MetaClique* and *MaxKPlex* according to the parameter k . We can see that for small k values, *MaxKPlex* gets better performance than *MetaClique*. As k becomes larger, however, our *MetaClique* outperforms *MaxKPlex*. The reason of this behavior is that when k is smaller, enumeration cost for maximal k -plexes is

expected to be not high. It would often be almost the same as that for enumerating maximal cliques. Thus, *MaxKplex* efficiently works for small k -values. On the other hand, even for small k -values, *MetaClique* must first enumerate maximal cliques to obtain primitive components and then construct a k -clique graph. We have to accept the construction cost as an overhead in our framework. After the construction of k -clique graph, however, *MetaClique* can efficiently extract solutions. Particularly, for a larger value of k , since k -clique graphs usually become sparse, our solutions can be obtained much more efficiently and the overhead for graph construction can be disregarded.

We have found that bond values do not affect both runtime and number of solutions greatly when $bond < 0.75$. When its value becomes larger than 0.75, times for building k -clique graphs decrease greatly. In such a case, since our k -clique graph tends to be sparse, we can obtain solutions very efficiently.

1.5.3 QUALITY

In order to verify quality of communities extracted by *MetaClique*, we compare our system with existing systems for community detection in PPI networks [Lee+10; Gre10].

A PPI network [Col+07] treats proteins as vertices and protein-protein interactions as edges. For a given PPI network, we define pre-discovered multiprotein complexes as the “ground-truth” communities, because proteins in multiprotein complexes are usually linked by protein-protein interactions. These “ground-truth” communities are considered as an answer set to be detected. Then we apply different community detection algorithms on the PPI network and compare their outputs with “ground-truth” communities.

In general, communities detected by those algorithms usually do not perfectly match the

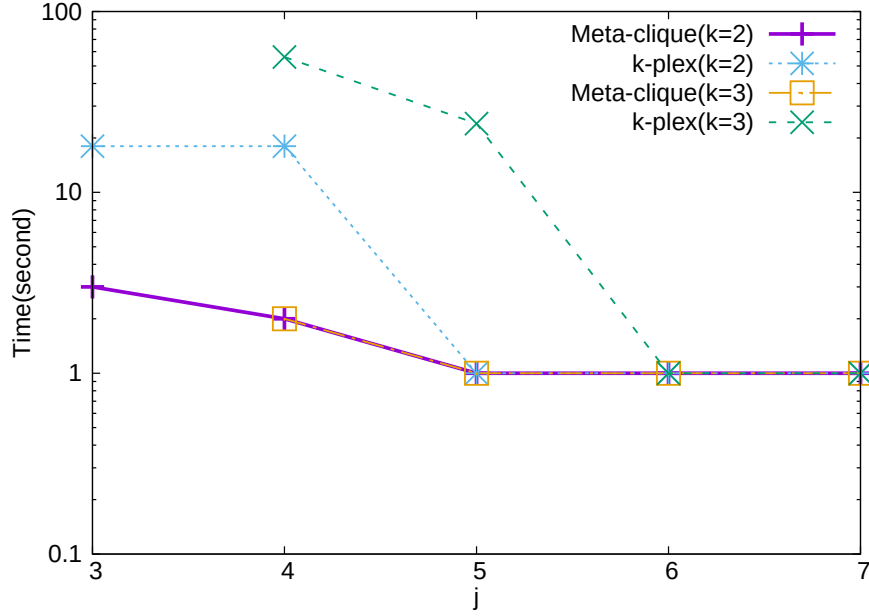


Figure 1.6: Computation times ($bond = o$) of RANDOM-10000-1

“ground-truth” communities. Moreover, we may detected more than one clusters which match one “ground-truth” community. The general definition of precision and recall can not be used directly here. Furthermore, when we consider the size constraint j , any ground-truth community with less than j vertices is out of our targets.

Given a threshold w , for a community c detected by an algorithm, if there exists a “ground-truth” community $\tilde{c}$ such that $\frac{|c \cap \tilde{c}|}{\min(|c|, |\tilde{c}|)} > w$, we consider that c matches $\tilde{c}$. We also say that $\tilde{c}$ is detected by the algorithm and c is a correct community. Based on this judgement, we define precision

$$p = \frac{\# \text{ of correct communities}}{\# \text{ of detected communities}} \quad (1.1)$$

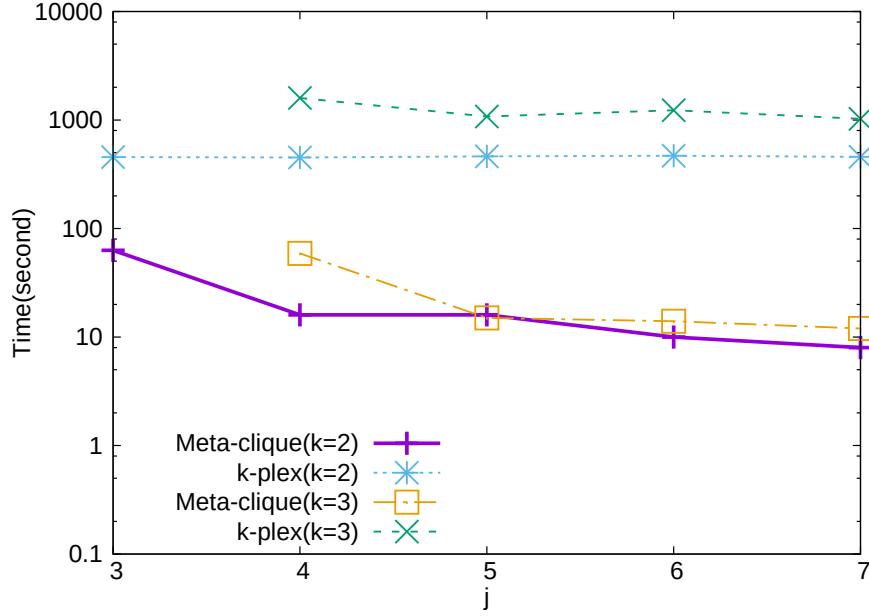


Figure 1.7: Computation times ($bond = 0$) of RANDOM-10000-2

and recall

$$r = \frac{\# \text{ of detected ground-truth communities}}{\# \text{ of ground-truth communities}} \quad (1.2)$$

To measure quality of communities detected by each algorithm, we here use $F1 - score$, defined as follows:

$$F1 - score = 2 \times \frac{p \times r}{p + r} \quad (1.3)$$

The higher the $F1 - score$, the better the quality of communities are. In our preliminary experiment, we have found that $w = 0.8$ is an adequate value to measure the correctness of detected communities allowing some divergence.

In our experimentation, we have used the Combined-AP/MS data [Col+07] from Yeast

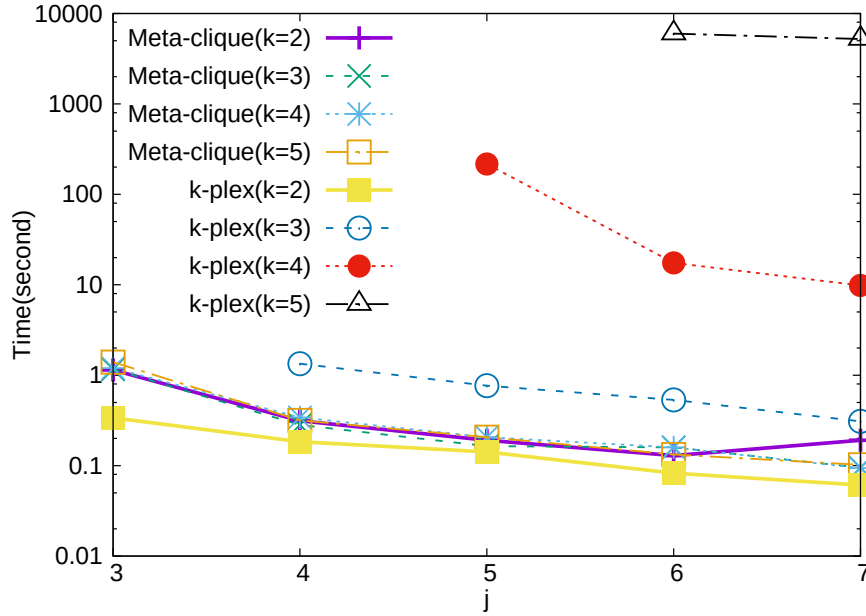


Figure 1.8: Computation times ($bond = o$) of CA-GRQC

Interactome Project as interaction network as a graph dataset. For ground-truth communities, we use CYC2008 [Pu+09] which contains the manually curated 408 heterometric protein complexes in *Saccharomyces cerevisiae*. We have applied different algorithms on the PPI networks then measured quality of detected communities.

Figure 1.21 shows F_1 -scores of *MetaClique*, Greedy Clique Expansion (GCE)* [Lee+10], LINK† [ABL10], Order Statistics Local Optimization Method (OSLOM2)‡ [Lan+11] and Community Overlap PPropagation Algorithm (COPRA)§ [Gre10] on the PPI network. GCE use distinct cliques as seeds and expands these seeds by greedily adding vertices under

*<https://sites.google.com/site/greedycliqueexpansion/>

†<https://github.com/bagrow/linkcomm>

‡<http://www.oslom.org/software.htm>

§<http://www.cs.bris.ac.uk/~steve/networks/software/copra.html>

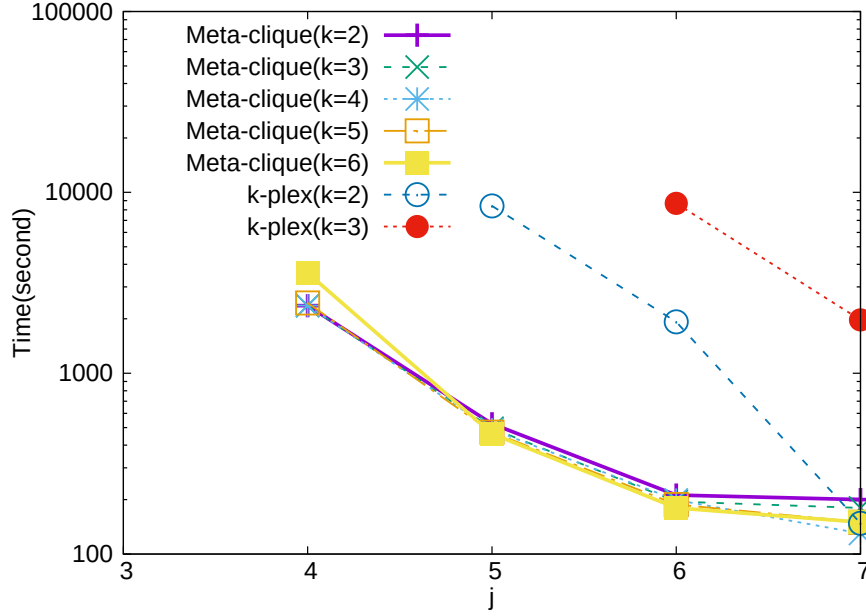


Figure 1.9: Computation times ($bond = o$) of COM-AMAZON

a predefined fitness function. GCE and *MetaClique* share the same idea that communities are formed by the densest cores. LINK is one kind of hierarchical clustering based on the interrelation of edges. OSLOM divide graph into several clusters by optimizing a fitness function expressing the statistical significance of clusters with respect to random fluctuations. COPRA finds communities by propagate vertices and their neighbours. As far as we know, GCE works best for the PPI network.

Each vertical line shows Fi -score and horizontal line $minsize$ -value for our minimum component size. Because COPRA, LINK and OSLOM are not clique-based systems, we can not apply constraints on cliques. Thus, they give horizontal lines through j . Even though Meta-Clique generally detects much more clusters than other algorithms, our algorithm still achieves the highest score in all the listed methods.

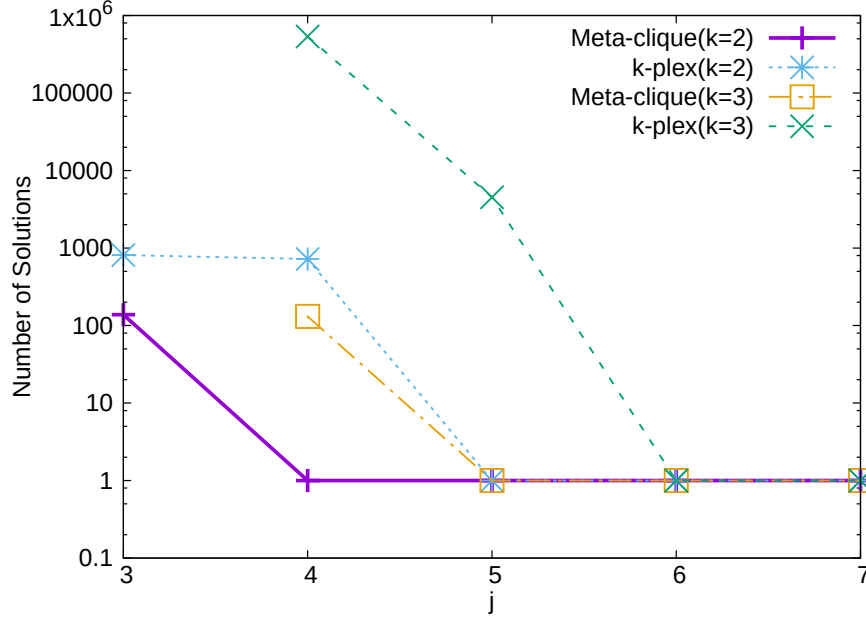


Figure 1.10: Number of Solutions ($bond = o$) of RANDOM-10000-1

When j increases, the quality of solutions by *MetaClique* worsens because only a small number of (larger) components are available. Even in such a case, however, *MetaClique* shows the highest score among all systems. Particularly, although GCE shows the best score at $j = 6$, our *MetaClique* can have a score clearly higher than that. It is emphasized here that *MetaClique* gives the best score at $j = 4$ at which it is difficult for GCE to detect solutions with high quality. This means that our *MetaClique* has an ability to detect many meaningful communities which are unfortunately missed by GCE.

We find that *MetaClique* outputs many highly-overlapped clusters which usually only have one vertex different from each other. In our evaluation, these highly-overlapped clusters are all marked as correct. However, in the task of community detection, highly-overlapped clusters may be unwelcomed. Thus, we need to reduce the overlapness of Meta-

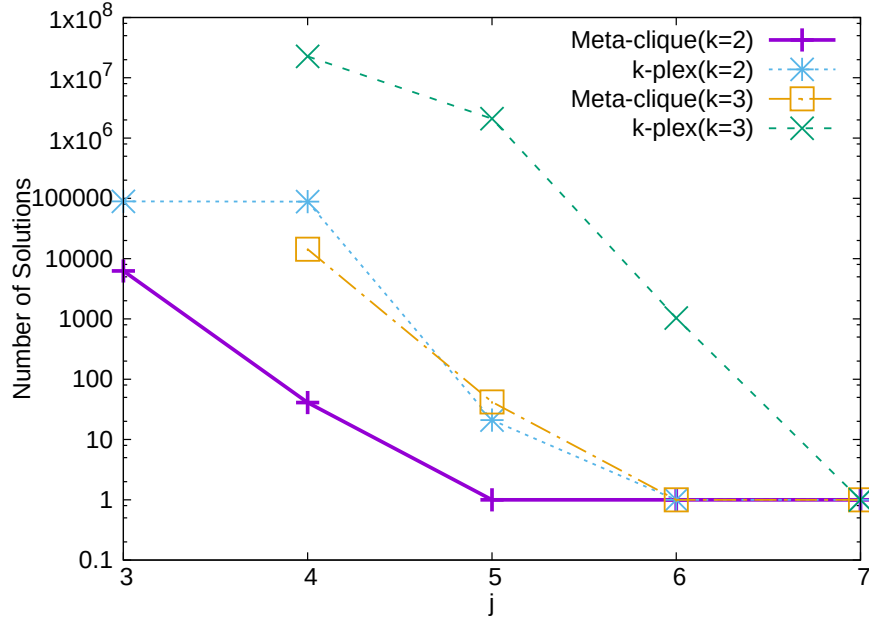


Figure 1.11: Number of Solutions ($bond = o$) of RANDOM-10000-2

Clique's output in future works.

1.6 CONCLUSIONS

In this research, we discussed Maximal k -MCS Problem for finding densely connected communities in a given graph. In order to restrict the solutions to what we want, we introduced constraints on component size and overlappingness.

Our problem of enumerating k -MCSs can be transformed into a problem of finding meta-cliques in a k -clique graph. Therefore, our algorithm has been designed based on CLIQUES [TTTo6], an efficient maximal clique enumerator. We empirically verified effectiveness of our method for several synthetic and real world networks from the view-

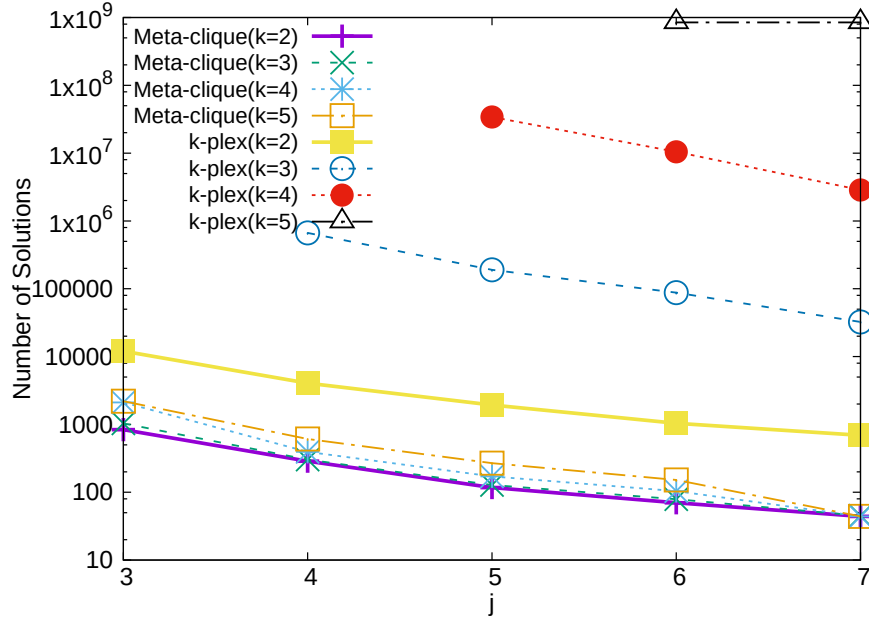


Figure 1.12: Number of Solutions ($bond = o$) of CA-GRQC

points of computational performance and quality of solutions. In order to make our method more appealing, we need to have a theoretical analysis of our algorithm as an important future work. Since the literature [TTTo6] has already investigated the worst-case time complexity for CLIQUES, it would provide many helpful and constructive suggestions in our theoretical analysis.

To get denser communities, we cut off small cliques. However, by removing them, we lose the ability of clustering the vertices contained only by these small cliques. Even though the number of small cliques is huge, most of them are highly overlapped. From this, we can surmise that the cover set of these vertices can be small and able to be considered without slowing down the algorithm. As a solution, a clique coverage heuristic process [Lee+10] can be introduced.

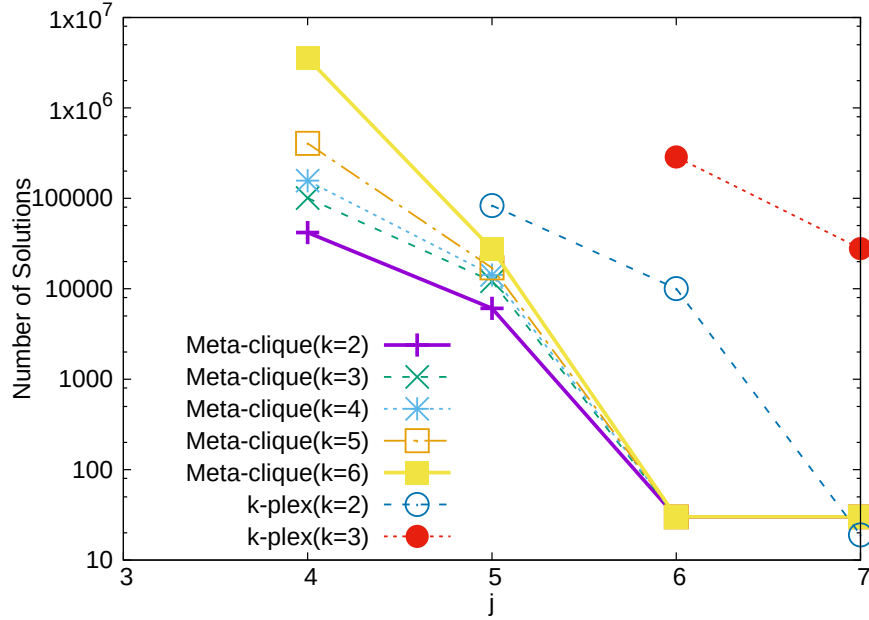


Figure 1.13: Number of Solutions ($bond = o$) of COM-AMAZON

Very recently, it has been argued that in many networks, a lower bound for the typical community size is given as 5 to 7 [SIL14]. Therefore, it is natural to assume a minimum size of components in order to detect meaningful communities. In this sense, our approach in this research would be quite reasonable. However, one might claim that our target k -plex as a combination of component cliques seems too restrictive. In order to make the restriction weaken, we can consider our components to be pseudo-clique so that we can obtain various dense communities more flexibly. Then it would be worth designing an efficient algorithm for such a relaxed maximal k -MCS problem.

As another approach in this direction, it would be promising to impose a constraint on connectivity of vertices in communities. More concretely speaking, for a community, we may require each vertex to be adjacent to a certain number of vertices in the community.

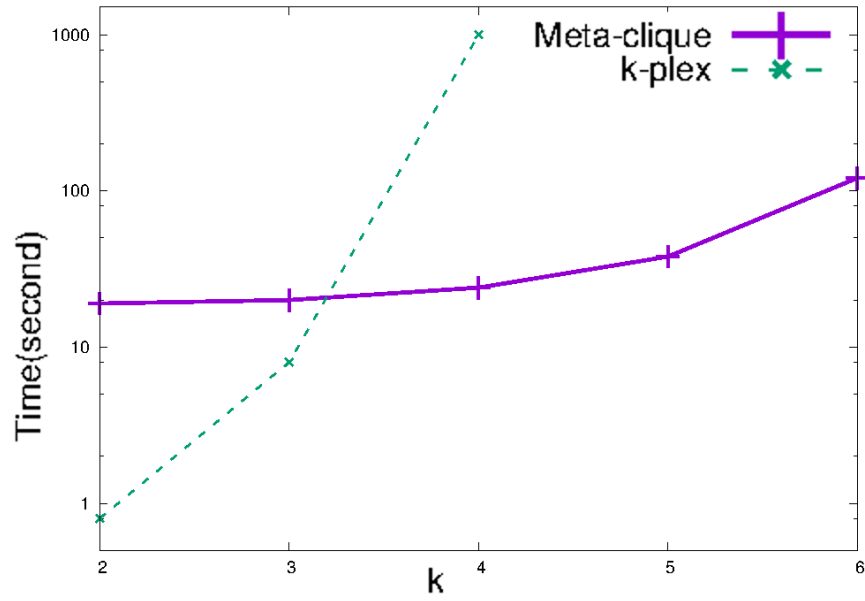


Figure 1.14: The Change of Computation Times with $k(j = 1, bond = 0)$ of GEOM-0

This kind of community can be formalized with the notion of *j-core* [BZO3] and is investigated as *j-cored k-plexes*. We will introduce this *j-cored k-plexes* in Chapter 2.

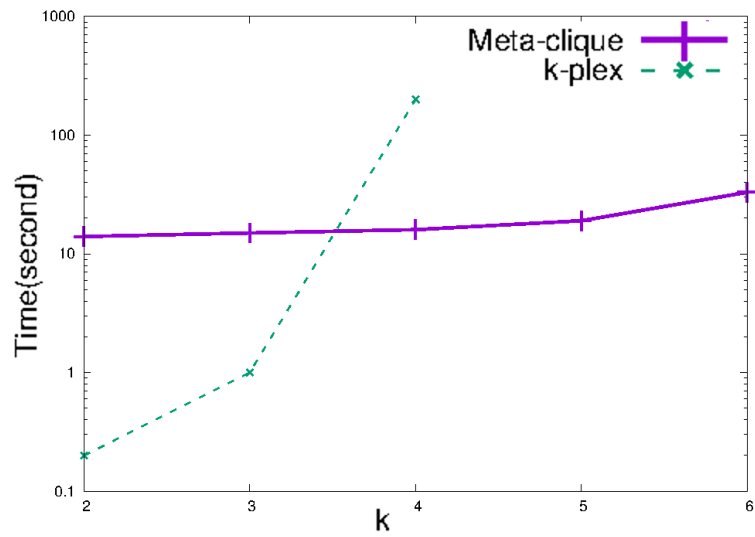


Figure 1.15: The Change of Computation Times with $k|j = 1, bond = 0$) of CA-GRQC

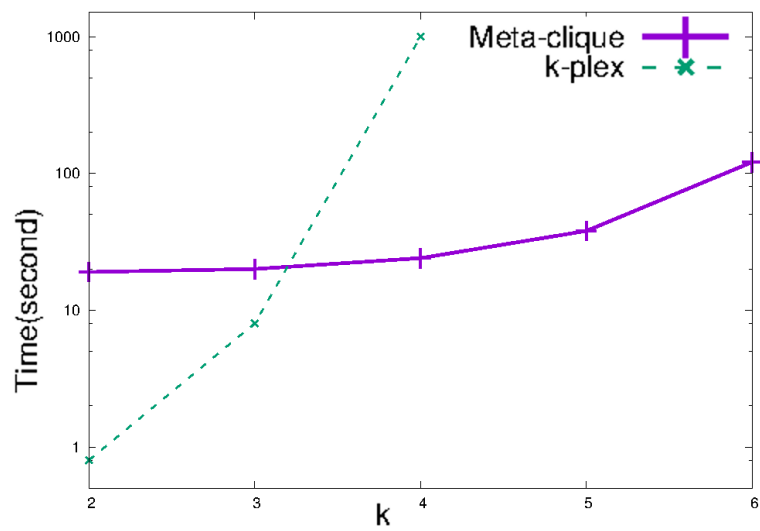


Figure 1.16: Computation Times on Large Graphs($bond = 0$) of RANDOM-50000-1

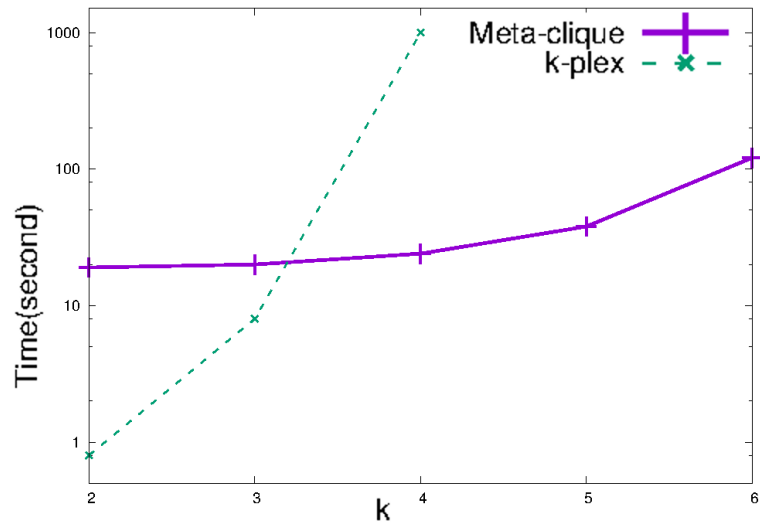


Figure 1.17: Computation Times on Large Graphs($bond = o$) of RANDOM-50000-2

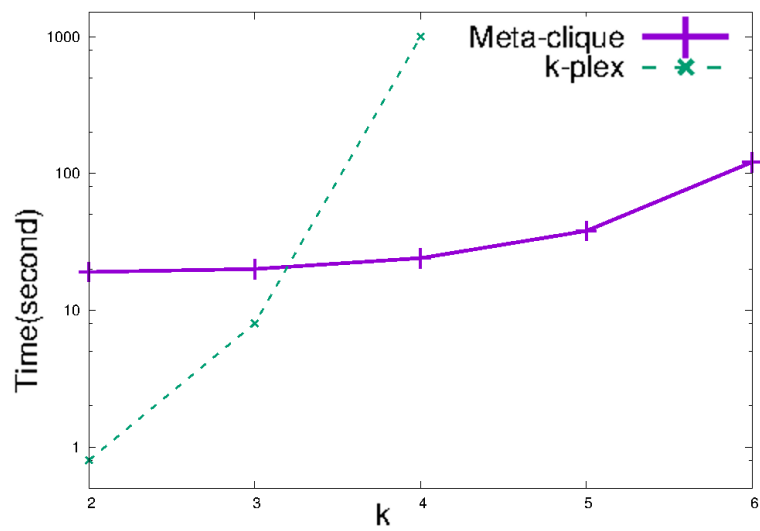


Figure 1.18: Computation Times on Large Graphs($bond = o$) of RANDOM-100000-1

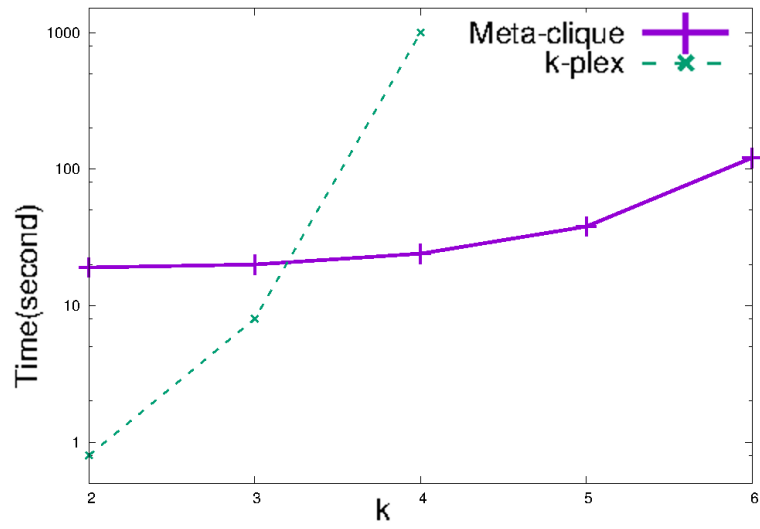


Figure 1.19: Computation Times on Large Graphs($bond = o$) of RANDOM-100000-2

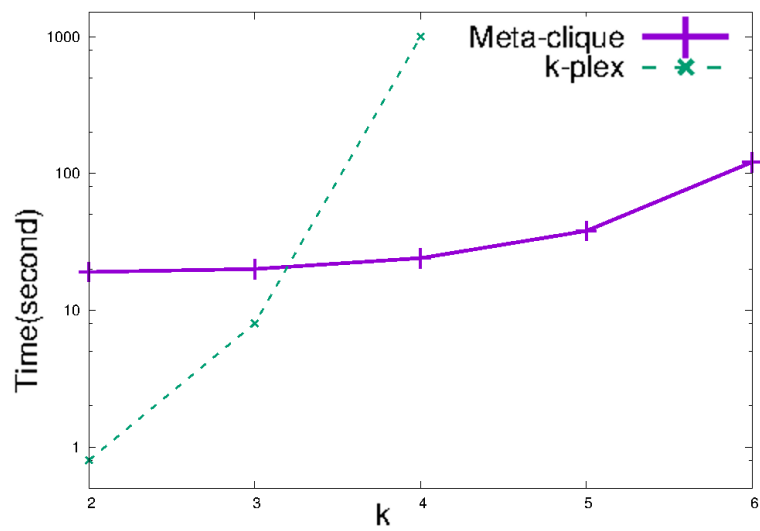


Figure 1.20: Computation Times on Large Graphs($bond = o$) of YOUTUBE

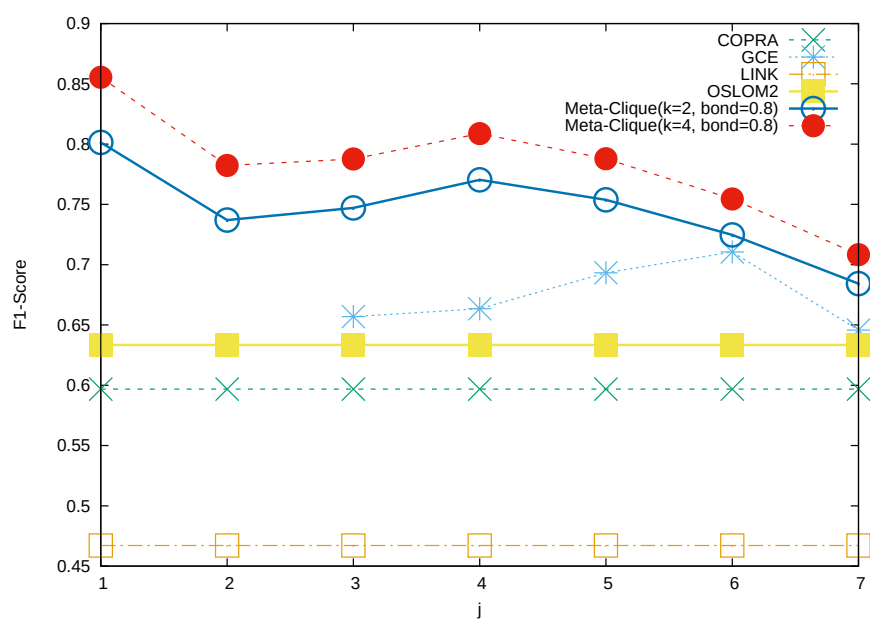


Figure 1.21: Comparison of F_1 -score on protein-protein interaction network

This work is originally published at Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2016) [Zha+16] and licensed for Thesis/Dissertation use by Springer. License Number: 4286930420734.

2

Enumerating Pseudo-Cliques with Density

Lower Bound

2.1 INTRODUCTION

Detecting communities in a network has been an important task in Social Network Analysis [SC11]. Cliques are typical vertex sets understood as potential communities [PYB12].

Moreover, the class of cliques has an anti-monotonicity property helpful in designing an efficient enumerator for them. In a real world network, however, the clique model is too restrictive to capture various communities because it is rare for actual communities to appear as cliques. This would motivate us to study clique relaxation models, and various pseudo-clique models have been proposed [PYB12].

As another standard approach to community detection, graph clustering or partitioning methods are well known to be useful (e.g. [Vono07; Newo06]). However, when we aim at obtaining smaller communities, we shall use (pseudo-) clique detectors instead of those methods because they usually suppose small numbers of clusters whose sizes are consequently non-small and hence our targets are invisibly merged and absorbed into those larger clusters.

In a density-based model of pseudo-cliques (e.g. [ARSo2]), some indices for measuring the density of vertex sets are presented. Unlike the clique model, since this class of pseudo-cliques do not satisfy the anti-monotonicity property, efficient but heuristic detectors have often been proposed for searching them [ARSo2]. We therefore might lose some vertex sets possibly valuable for us. Moreover, even if we have a complete detector, it cannot process large scale networks because of the hugeness of the number of pseudo-cliques. In spite of these difficulties, it is still important to develop an efficient and complete pseudo-clique enumerator that can handle large networks because such an enumerator is useful not only for community discovery but also for analyzing the nature of large networks in terms of statistics about (pseudo-) cliques [SIL14].

As another clique relaxation models, a distance-based model, k -clique [Luc50], and diameter-based models, k -club and k -clan [Alb73; Mok79], have been proposed. The pa-

parameter k controls admissible distances among vertices. As discussed in [WS98], when we allow a longer distance, large dense subgraphs appear which are almost cliques even when its subgraphs w.r.t. the original edge connection are not dense.

On the other hand, a k -Plex model [SF78] discusses the density in terms of original connection by setting an upper bound for the number of missing edges among vertices. The class of k -Plexes has an anti-monotonicity property which helps us to design a simple bottom-up enumerator [WP07; BCK15]. For this reason, we discuss k -Plexes in this chapter, introducing new constraints that can cover its weakness we discuss just below.

A vertex set is called a k -Plex if, for each vertex x in it, the number of vertices not adjacent to x is at most k . *Disconnection Upper Bound (DUB)* is thus specified by the parameter k . Note that a k -Plex could be non-connected. However, since it seems not interesting as a community, we exclude those ones from our consideration.

For a very small k , connected k -Plex is in fact dense if its size is relatively larger than k . On the other hand, as the size of densely connected vertex set increases, the number of disconnected vertices, k per each vertex, shall be non-small depending on density requirement. In other words, we think some constraint taking the sizes of targeted pseudo-cliques into account is important. When we suppose a k -Plex of size n , then each vertex has at least $n - k$ adjacent vertices, where $n - k$ must be a certain number provided we admit the vertex set as a densely connected one. We thus introduce another constraint using a parameter j designating *Connection Lower Bound (CLB)*. Then we try to enumerate all maximal connected k -Plexes meeting the CLB constraint.

A naive strategy, that is, computing maximal k -Plexes and then checking the CLB constraint, does not work well when we consider non-small k . This is because every vertex set

with size no larger than k is trivially a k -Plex, so we have to examine an exponential number of such sets. The key to solve this problem lies in another fact that, if a connected k -Plex X can be extendable to a maximal connected k -Plex under the CLB constraint, X is involved in a “core” of X together with candidates that are potential vertices to be added to X , where the term “core” means the largest subset of vertices with at least j adjacents in the subset [BZ03].

In this chapter, j is fixed beforehand depending on the targeted vertex set size. The monotonicity of core operation is suitable to standard k -Plex enumerator enjoying the anti-monotonicity of k -Plexness. Based on the monotonicities, we can design an efficient complete depth-first algorithm which can exclude numerous hopeless k -Plexes that cannot be extended to maximal ones meeting our requirement. This realizes much improvement of performance of k -Plex enumerator. In our experimentation, we mainly compare our algorithm with a state-of-the-art maximal k -Plex enumerator proposed very recently [BCK15] from the viewpoints of computational performance and quality of solution k -Plexes as pseudo-cliques. For synthetic and real world large networks including a Web graph with over 800,000 vertices, it is verified that the CLB constraint as well as our pruning mechanisms are quite effective. As the results, the proposed algorithm can work very well as a practical tool for detecting dense pseudo-cliques.

As an alternative to our formalization of densely-connected pseudo-cliques as maximal k -Plexes satisfying CLB constraint, one might consider maximal vertex sets which can form k -Plexes satisfying the constraint, where the class of those maximal sets completely subsumes our solutions. It seems more preferable to ours because we would miss possibly many vertex sets densely-connected. Our experimentation, however, shows the number of those sets

we actually miss is quite small. We can, therefore, argue that our model of pseudo-cliques is reasonable and practical.

The notion of densely-connected pseudo-cliques is a fundamental and useful structure not only in Social Network Analysis but also in other application domains, for example, Biological Systems [JS11] and Financial Markets [Nago03]. Our method proposed in this chapter is a general framework because we do not make any assumption specific to some particular domain. Thus we expect our algorithm would be applicable and work well in various application domains.

The remainder of this chapter is organized as follows. In the next section, basic terminologies and notations used throughout this chapter are introduced. Section 3 presents a notion of maximal connected k -Plexes, k -MPCs. Then we define our targets, j -cored maximal connected k -Plexes, (j, k) -MPCs, and show their theoretical properties in Section 4. Effective search control rules for efficient computations of (j, k) -MPCs are discussed in Section 5. Section 6 presents our algorithm for enumerating (j, k) -MPCs with the control rules. Our experimental results for synthetic and real world networks are presented in Section 7. Finally we conclude the paper in Section 8 with a summary and future work.

2.2 PRELIMINARIES

An undirected graph G is denoted by $G = (V, \Gamma)$, where $V = \{v_1, \dots, v_{|V|}\}$ is a set of vertices and $\Gamma(v_n) = \{v_m \in V \mid v_n, v_m \text{ are adjacent}\}$. $\Gamma(v_n)$ is assumed to not include v_n itself, that is, $v_n \notin \Gamma(v_n)$. An ordering $\prec$ over V is defined by $v_i \prec v_j$ and v_i is said younger than v_j iff $i < j$. If the identifiers, i of v_i , need not to be specified, we prefer the notation for vertices as v, x, u and so on. For a vertex set $X \subseteq V$, $G[X]$ is a subgraph of G induced by X .

Notation	Definition	Meaning/Interpretation
G	(V, Γ) , where $\Gamma \subseteq (V \times V) \setminus \{(x, x) x \in V\}$	an undirected graph with a set V of vertices and an adjacency relation Γ on V
$\Gamma(x)$	$\{v \in V (x, v) \in \Gamma\}$	the set of vertices in V adjacent to a vertex x
$G[X]$	$G[X] = (X, \Gamma \cap (X \times X))$	the subgraph of G induced by a set of vertices X
$\Gamma_X(x)$	$\Gamma(x) \cap X$	the set of vertices adjacent to a vertex x in $G[X]$
$\deg_X(x)$	$ \Gamma_X(x) $	the degree of a vertex x in $G[X]$
$Cand(X)$	$\{y \in V Xy \text{ is a } k\text{-Plex}\}$	the set of k -Plex candidates of a vertex set X
$dist(X, y)$	the minimum length of paths in G from X to y	the distance between a vertex set X and a vertex y
$D_n(X)$	$\{y \in V dist(X, y) = n\}$	the set of vertices whose distance from a vertex set X is exactly n
$K_1(X)$	$D_1(X) \cap Cand(X)$	the set of k -Plex candidates adjacent to a vertex set X

Table 2.1: Basic notations

For a vertex $x \in X$, $\Gamma_X(x)$ denotes $\Gamma(x) \cap X$. $|\Gamma_X(x)|$ is often referred to as $\deg_X(x)$.

A vertex set X is called a k -Plex if $|X - \Gamma_X(x)| \leq k$ for any $x \in X$. It is easy to see that for a k -Plex Y , any subset X of Y is also a k -Plex. Thus, the class of k -Plexes has an anti-monotonicity which is often used to design maximal k -Plex enumerators. A vertex $y \notin X$ is called a k -Plex candidate if Xy is still a k -Plex, where Xy is an abbreviation of $X \cup \{y\}$. $Cand(X)$ denotes the set of all k -Plex candidates of X . We especially discuss in this chapter connected k -Plexes (c- k -Plexes, for short).

For a vertex set X and a vertex y , the distance between X and y , $dist(X, y)$, is given by the minimum length of paths in G from X to y , where $dist(X, x) = 0$ whenever $x \in X$.

$dist(X, y) = \infty$ only when y is not reachable from X . $D_n(X)$ is defined as

$\{y \in V \mid dist(X, y) = n\}$. $K_1(X) = D_1(X) \cap Cand(X)$, the set of k -Plex candidates directly connected to X , plays a key role in discussing c - k -Plexes and is called a K_1 -candidate set at X .

The basic notations are listed in Table 2.1.

2.3 MAXIMAL CONNECTED k -PLEX

A maximal c - k -Plex, abbreviated as k -MPC, is a c - k -Plex that is maximal among c - k -Plexes.

It is clear a c - k -Plex X is extendable to its super c - k -Plex iff $K_1(X) \neq \emptyset$. Thus, the extension is made by adding K_1 -candidates at X to X .

A formation X^f of c - k -Plex X w.r.t. an indexing f is a sequence of vertices $v_{f(1)}, (v_{f(1)}, v_{f(2)}), \dots, v_{f(|X|)}$, where $f(i)$ is the identifier of i -th vertex added to form $X = \{v_{f(1)}, \dots, v_{f(|X|)}\}$ as sets, and all the intermediate $X_i^f = \{v_{f(1)}, \dots, v_{f(i)}\}$ must be a c - k -Plex. That is,

$$v_{f(i+1)} \in K_1(X_i^f). \quad (2.1)$$

The sequence $(X_1^f, \dots, X_{|X|}^f)$ is also called a formation of X . For any k -MPC Z , its formation ends at $Z = Z_{|Z|}^f$ with $K_1(Z) = \emptyset$.

We have multiple formations Z^f for a k -MPC Z , depending on the order of vertex addition. In fact, for an arbitrary initial vertex $v_{n_1} \in Z$, repeat the addition of $v_{n_i} \in Z$ such that v_{n_i} is adjacent to some vertex added prior to v_{n_i} . Then any intermediate $X_i(\subseteq Z)$ is a c - k -Plex by the monotonicity of k -Plexes and Z^f with the index f such that $f(i) = n_i$ gives a formation of Z . In Section 2.5, we introduce some control rules to disregard useless for-

mations under the restriction that formations must be those for k -MPCs whose density is guaranteed by another parameter j .

2.4 j -CORED k -MPC

In this section, we define a class of k -MPCs, as our targets, which are dense in the sense that each vertex has at least j adjacents in them. We then discuss how to enumerate those targets completely and efficiently.

Given a graph $G = (V, \Gamma)$, a vertex set X is said to be j -cored if $\deg_X(x) \geq j$ for any $x \in X$. The largest j -cored set, denoted by $\text{core}_j(V)$, is called j -core of G [BZ03]*. For a vertex set X , $\text{core}_j(X)$ is the j -core of the induced subgraph $G[X]$. We can observe a monotonicity of j -core operation.

Fact (Monotonicity of j -cores): For vertex sets X_1, X_2 of V , $\text{core}_j(X_1) \subseteq \text{core}_j(X_2)$ holds whenever $X_1 \subseteq X_2$.

Verification: It is easy to see $\text{core}_j(X_1) \subseteq X_1 \subseteq X_2$. For any $x_1 \in \text{core}_j(X_1)$, since $\deg_{X_1}(x_1) \geq j$, we have $\deg_{X_2}(x_1) \geq \deg_{X_1}(x_1) \geq j$. This means $x_1 \in \text{core}_j(X_2)$ for any $x_1 \in \text{core}_j(X_1)$, thus $\text{core}_j(X_1) \subseteq \text{core}_j(X_2)$. ■

The construction of $\text{core}_j(X)$ for a vertex set X is simple. Roughly speaking, it can be obtained by iteratively removing vertices with degree less than j from X . Since removing some vertices in general decreases degree of other vertices, we can iterate such a removal until no vertex can newly be removed.

*The notion of j -core has originally been defined in [Sei83]. We use in this chapter the definition and construction method for j -core, according to [BZ03].

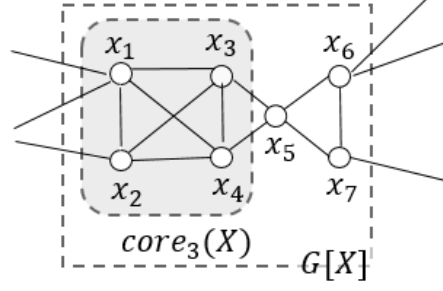


Figure 2.1: Example of $core_j(X)$ for $j = 3$ case

A behavior of j -core operation is illustrated in Figure 2.1. Any connection towards outside of X is ignored as we consider the subgraph $G[X]$. For this X , for $i = 6, 7$, v_i are firstly removed as $deg_X(x_i) = 2 < j = 3$. By this removal, $auxdeg(x_5)$ becomes 2. So x_5 is furthermore removed. Then, no more removal is made by degree lower bound constraint in this case, since all remaining vertices x_1, x_2, x_3 and x_4 still have at least j adjacents even after the removals so far.

Our target is now defined as a k -MPCs which is j -cored, called a j -cored k -MPCs ((j, k) -MPCs, for short). The following discussion is devoted to an efficient algorithm for enumerating every (j, k) -MPCs.

Unlike k -Plexes, the class of j -cored vertex sets does not have anti-monotonicity property. In spite of this fact, we can give a sufficient condition for c - k -Plexes to grow to j -cored ones. Then, we can use the condition conversely to reject hopeless c - k -Plexes appearing in formations.

Fact (Hopeful c - k -Plex): Let X be a c - k -Plex extendable to some (j, k) -MPC by adding ver-

tices to X . Then

$$X \subseteq U(X), \text{ where} \quad (2.2)$$

$$U(X) = \text{core}_j(X \cup \text{Cand}(X)). \quad (2.3)$$

Verification: As $Xz \subseteq Z$ for any $z \in Z - X$ and Z is a k -Plex, Xz is also a k -Plex by the anti-monotonicity of k -Plexes. So, $Z \subseteq X \cup \text{Cand}(X)$. Z is j -cored, so we have $X \subseteq Z \subseteq \text{core}_j(X \cup \text{Cand}(X))$. ■

X satisfying (2.2) is said to be hopeful. Conversely, hopeless ones are those satisfying the negation of (2.2).

$$(\text{Hopelessness}) \quad X - U(X) \neq \emptyset \quad (2.4)$$

The definition (2.3) of $U(X)$ allowing k -Plex candidates $\text{Cand}(X)$ before taking j -core is weak particularly for small X with its size less than k . In that case, a large number of vertices appear as k -Plex candidates independently of the length of paths connecting X and candidates. As X is extended to include vertex y more distant from X , the shortest path connecting X and y in the extension becomes longer. Consequently, such an extended Y will not be k -Plex, because there exist many vertices not adjacent to y on the path. As long as we target connected k -Plexes including X , the distance limit becomes a key to define $U_1(X) = U(X) \cap D_1(X)$ as a set of potential candidates to obtain (j, k) -MPCs. Particularly, we can construct $U(X)$ depending on the sizes of X .

Figure 2.2 shows that j -core under such a distance limit is useful, and also suggests that

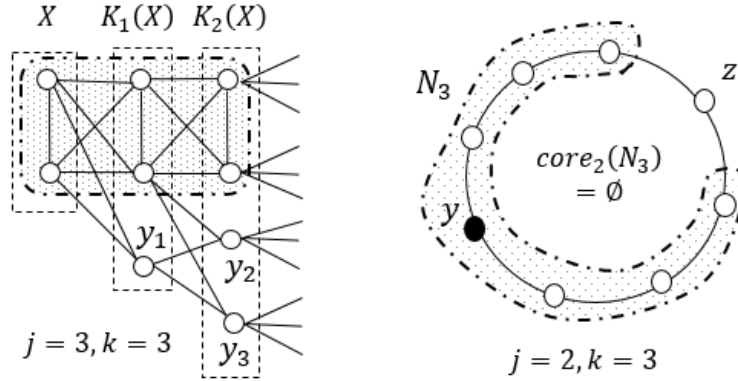


Figure 2.2: Examples showing how j -core is useful.

the construction of $U(X)$ depends on the sizes of X . In the right figure, for a singleton Y of the black circle y , $Y \cup \text{Cand}(Y)$ is the whole vertex set that is j -cored, where $j = 2$. Nothing is therefore removed by taking j -core for $Y \cup \text{Cand}(Y)$. Nevertheless, note that the vertex z cannot be consistent with Y to make a c - k -Plex, as $\text{dist}(Y, z) = 4 > k = 3$. More precisely, on the paths connecting y and z , there are 4 vertices not adjacent to z including z itself. Hence, there is no c -3-Plex involving Yz . For this reason we removed z before we test if Y is extendable to j -cored c - k -Plexes. N_3 is the set of remaining candidates together with Y . Then $\text{core}_j(N_3) = \emptyset$ asserting that there exists no j -cored c - k -Plex containing Y .

On the other hand, the left figure shows a case of X with $|X| = 2$. Similarly, when $k = 3$, vertices with $\text{dist}(X, z) > 3$ cannot be members of c -3-Plex including X . After excluding vertices violating the distance limit, we take 3-core. y_2, y_3 are firstly removed, and then y_1 is. The remaining part, surrounded by dotted line and filled by pattern, is 3-cored and is a maximal 3-Plex in this case.

2.4.1 SMALL C- k -PLEX

A c - k -Plex X is said to be small if $|X| < k$. Note here that any connected X with small size is a c - k -Plex. In this case, therefore, $Cand(X)$ is given as $V - X$ which is generally large. To disregard useless candidates, we make a question whether there exists some c - k -Plex Z such that $Xy \subseteq Z$ for a small c - k -Plex X and $y \notin X$. When $dist(X, y) = 1$, $Z = Xy$ is a trivial positive answer, so we analyze the case of $\ell = dist(X, y) \geq 2$. Consider the shortest path $p = (y_0, y_1, \dots, y_{\ell'} = y)$ from X to y in $G[Z]$, where $y_0 \in X$ and $y_1, \dots, y_{\ell'} \in Z$. Then, $\ell \leq \ell'$, and y is not adjacent to $X, y_1, \dots, y_{\ell'-2}$ and $y_{\ell'} = y$ itself. Z is a k -Plex, so it must be $|X| + (\ell' - 2) + 1 \leq k$. Hence we have $\ell \leq \ell' \leq k - |X| + 1$. In other words, y with $dist(X, y) > k - |X| + 1$ can never be a member of any c - k -Plex including X . So, we have more accurate definition for $U(X)$ for small case as follows.

$$U(X) = core_j(X \cup K(X)),$$

$$K(X) = \bigcup_{i=1}^{k-|X|+1} D_i(X),$$

where $k - |X| + 1$ is the distance limit.

The distance limit decreases as X enlarges, so $K(X)$ is monotonic decreasing.

2.4.2 MEDIUM C- k -PLEX

We say that a c - k -Plex X is medium if $k \leq |X| < j + k$. For a medium c - k -Plex X , any vertex whose distance from X is greater than 1 is never a k -Plex candidate, as it has no connection with at least k vertices in X . Hence, $Cand(X) = K_1(X)$, and $U(X)$ is exactly given as

$$U(X) = core_j(X \cup K_1(X)).$$

2.4.3 LARGE C- k -PLEX

We say that c- k -Plex X is large if $|X| \geq j + k$. Any large X is j -cored, as $j \leq |\Gamma_X(x)|$ is derived from $j + k \leq |X|$ and $|X - \Gamma_X(x)| = |X| - |\Gamma_X(x)| \leq k$. We need not take j -core, that is, $U(X) = X \cap \text{Cand}(X)$. So we have the rule for updating U as follows.

$$U(Xu) = U(X) \cap \text{Cand}(Xu), \text{ where } u \in U_1(X).$$

The rule for large case is the same as one for k -Plexes.

2.4.4 FORMATIONS REVISED FOR (j, k) -MPCs

Using $U(X)$ thus defined depending on $|X|$, a formation $Z^f = (v_{f(i)}, \dots, v_{f(|V|)})$ of (j, k) -MPC Z satisfies

$$U(Z_{i+1}^f = Z_i^f v_{f(i+1)}) \subseteq U(Z_i^f) \text{ and} \tag{2.5}$$

$$v_{f(i+1)} \in U_1(Z_i^f), \text{ where } U_1 \text{ depends on } |Z_i^f|, \tag{2.6}$$

$$U_1(Z = Z_{|Z|}^f) = \varnothing. \tag{2.7}$$

(2.6) is stronger than (2.1) since $U_1(Z_i^f) \subseteq K_1(Z_i^f)$. The condition (2.7) holds because

$U_1(Z) \subseteq K_1(Z)$ and

$K_1(Z) = \varnothing$ by the property of (j, k) -MPC as a k -MPC.

2.5 SEARCH CONTROL RULES, RIGHT AND LEFT ONES

As a c - k -Plex Z is more densely connected, the number of possible formations of Z increases. For our efficient computation, we can enjoy two search control rules, right and left rules, to avoid useless and duplicated formations.

With the right candidate control (abbreviated as RCC), we can exclude many useless formations. A formation $Z^f = (v_{f(1)}, \dots, v_{f(|Z|)})$ can be obtained by extending an intermediate Z_i^f with a vertex in $U_i(Z_i^f)$. In other words, for complete enumeration of formations, we have to examine every vertex in $U_i(Z_i^f)$ to extend Z_i^f . Some of them, however, result in non-maximal k -Plexes. Extending an idea discussed in [TTTo6; ES11], we can identify the candidate set, $R(Z_i^f)$, with the property that adding member vertices in $R(Z_i^f)$ only to the present Z_i^f never achieves a maximal k -Plex. Thus, the set of vertices actually used to extend Z_i^f is given as $NR(Z_i^f) = U_i(Z_i^f) - R(Z_i^f)$, called non-right candidates. Our RCC here can be regarded as an extended version of RCCs in [WP07] in the sense that it is much more applicable for non-small Z_i^f than that in [WP07]. Although we skip the details due to space limitation, interested readers can refer to [OHT12].

In addition to RCC, we can enjoy a left candidate control (abbreviated as LCC) with which just one formation for each (j, k) -MPC can be composed. LCC is in some sense a standard technique for set enumeration [Rym92] and is almost similar to what is stated in [TTTo6]. Roughly speaking, when we extend an intermediate Z_i^f , we do not need to care any vertex y such that $y \prec v_{f(i)}$, called a left candidate, because any formation obtained by extending Z_i^f with such a y can be composed by extending another intermediate formation with some left candidate $\ell(\neq y)$. Thus, the set of vertices we actually use to extend Z_i^f is

given by $NR(Z_i^f) - L(Z_i^f)$, where $L(Z_i^f)$ is the set of left candidates. Although we do not go into the details, effects of the control rules in the form of search tree are illustrated in the next section.

```

procedure MAIN( $G^{input}$ ):
  [Input]  $G^{input} = (V^{input}, \Gamma^{input})$ : an input graph.
  [Output] All maximal  $j$ -cored  $c$ - $k$ -Plexes including  $(j, k)$ -MPCs
  begin
    for each connected component  $C$  of  $core_j(V^{input})$ 
      Let  $G = (C, \Gamma = \Gamma_C^{input})$ ;
       $\tilde{u} = \arg \max_{u \in C} deg(u)$ ;
       $NR = C - \Gamma(\tilde{u})$ ;  $L = \varnothing$ ;
       $X = \varnothing$ ;  $U = NR$ ;
       $Expand(X, NR, L, U)$ ;
    endfor
  end

procedure EXPAND( $X, NR, L, U$ ):
  //  $X$ :  $c$ - $k$ -Plex,  $NR = NR(X)$ ,  $L = L(X)$ ;  $U = U(X)$  (depending on  $|X|$ )
  begin
     $U_1 = U \cap D_1(X)$ ; //for non-small case,  $U_1 = U$ 
    if ( $U_1 = \varnothing$ ) then
      print  $X$ ; return; // maximal  $j$ -cored  $c$ - $k$ -Plex including  $(j, k)$ -MPC
    endif
    for each  $v \in NR - L$  // the order accords to  $\prec$ 
       $X_{new} = Xv$ ;
       $U_{new} = U(X_{new})$ ;
      if  $X_{new} - U_{new} \neq \varnothing$  then continue; //  $X_{new}$  is hopeless
       $NR_{new} = NR(X_{new})$ ;  $L_{new} = (L(X) \cup \{v\}) \cap U_{new}$ ;
       $Expand(X_{new}, NR_{new}, L_{new}, U_{new})$ ;
    endfor
  end

```

Figure 2.3: Enumeration algorithm for (j, k) -MPCs

2.6 ALGORITHM FOR (j, k) -MPCs

We present an algorithm in Figure 2.3 for making formations for (j, k) -MPCs, where, at each intermediate c - k -Plex $X = Z_i^c$, only non-left and non-right candidates x at X are added to extend partial formations, provided X together with x is hopeful. Any hopeless cases are cut off immediately. Although the whole control structure is simple, remind that the construction of $U_1(X)$ and $R(X)$ depends on the size $|X|$.

The algorithm is written using recursive calls of procedure *Expand*. This realizes a depth-first search of (j, k) -MPCs.

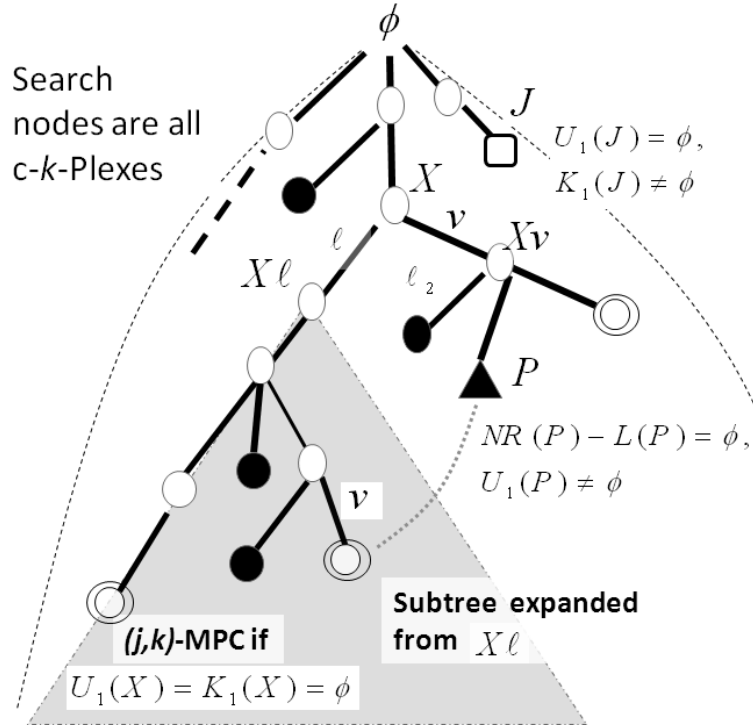


Figure 2.4: Search tree

Name	Number of Nodes	Number of Edges	Density	Cluster Coefficient
BA	10,000	99,900	0.001998	0.010870
WS	50,000	500,000	0.000400	0.519607
GrQc	5,242	14,496	0.001054	0.529636
DAYS	13,332	5,616	0.000063	0.037430
DBLP	317,080	1,049,866	0.000021	0.632431
Google	875,713	4,322,051	0.000011	0.514296

Figure 2.5: Scales of networks

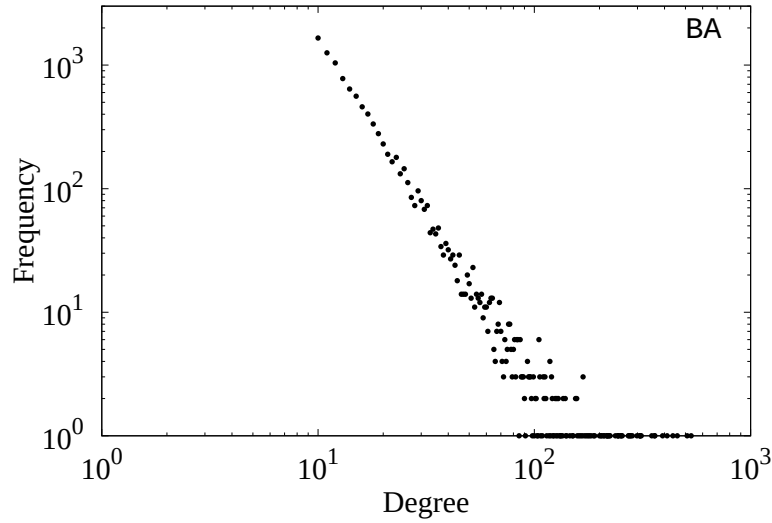


Figure 2.6: Degree distributions of BA

We depict in Figure 2.4 the invocation process of *Expand* in the form of search tree with c - k -Plexes as its nodes. In the search tree, a path from the root, an empty set $\emptyset$, to a leaf just corresponds to a formation of c - k -Plex. Dark circle shows hopeless c - k -Plexes from which no branch is expanded. Double circle is a (j, k) -MPC. Single circle has chance of extension by choosing some of

$NR(X) - L(X) \neq \emptyset$. Dark triangle is a c - k -Plex P such that every non-right candidate is left. This means that (j, k) -MPC Z obtained by extending X is generated by another branch

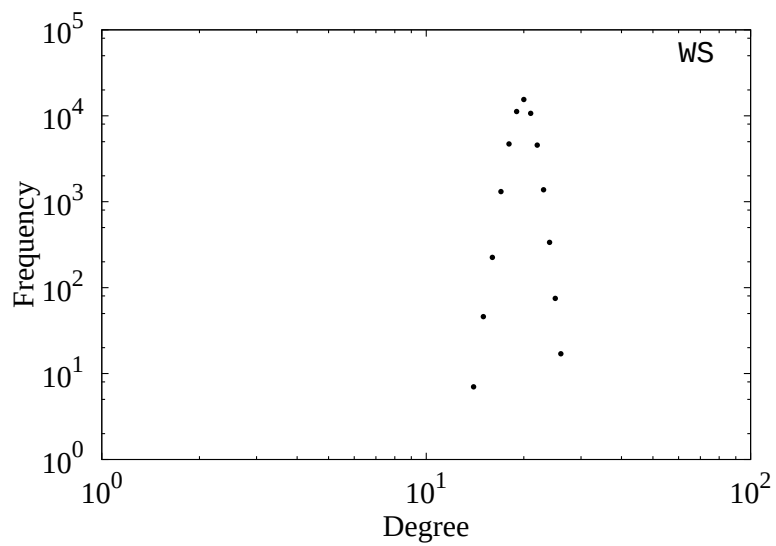


Figure 2.7: Degree distributions of WS

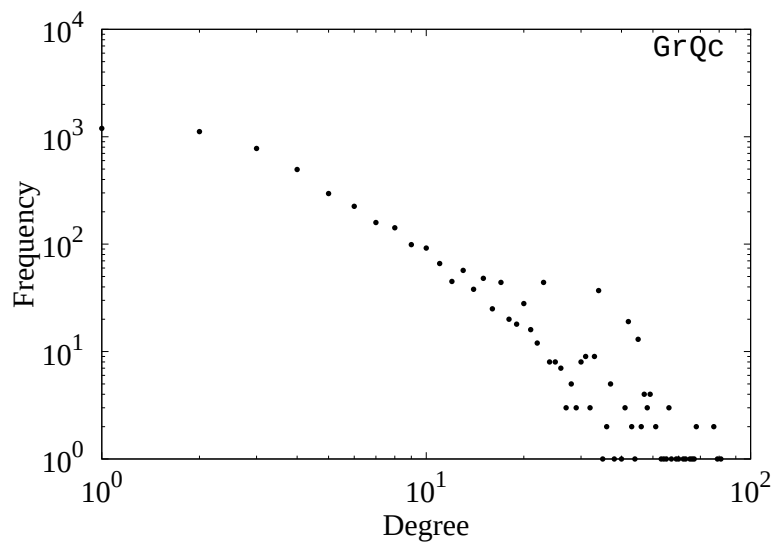


Figure 2.8: Degree distributions of GrQc

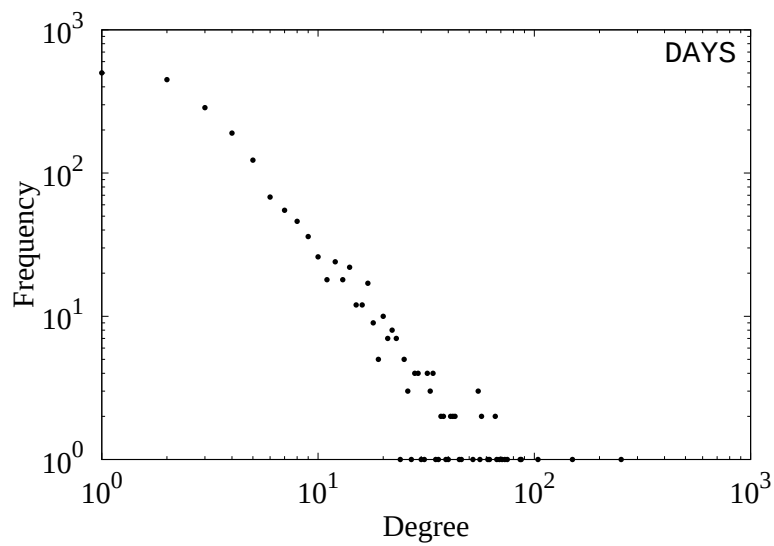


Figure 2.9: Degree distributions of DAYS

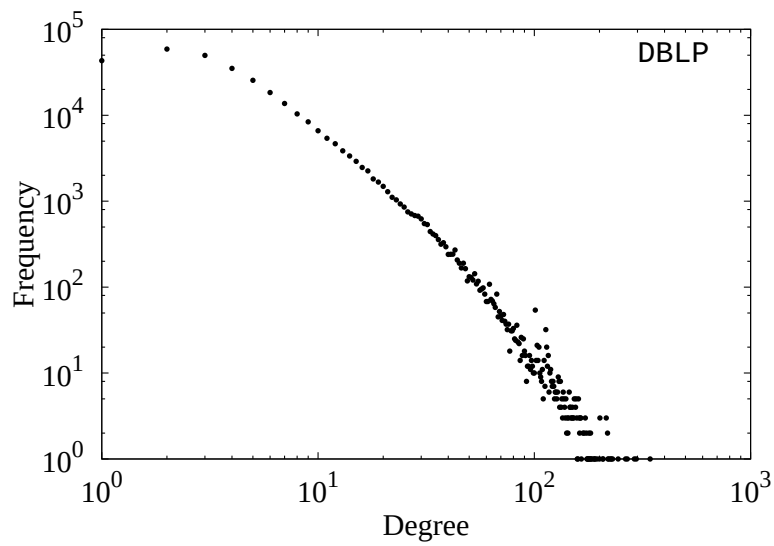


Figure 2.10: Degree distributions of DBLP

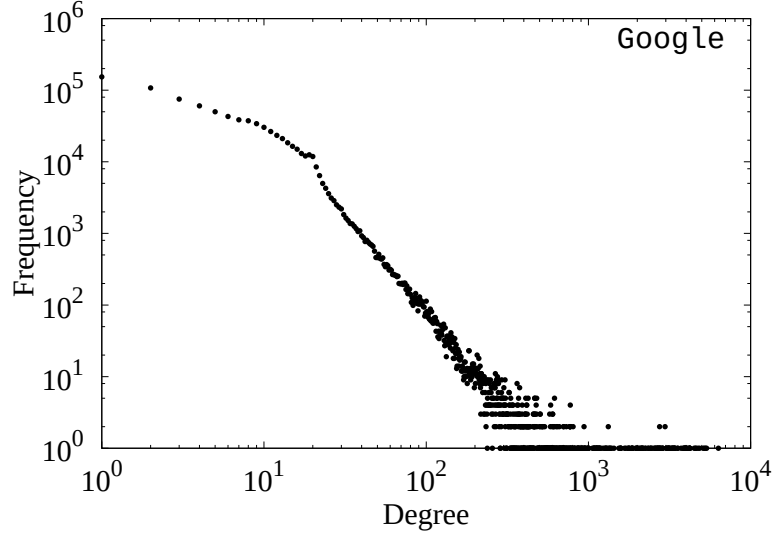


Figure 2.11: Degree distributions of Google

with some left candidates ℓ along the path. In other words, with some initial segment X of Z and its non-left $\ell \in NR(X)$, Z appears in the subtree rooted by $X\ell$. Finally, white square J is a j -cored c - k -Plex which is not k -MPC. It is maximal in the sense that there exists no j -cored c - k -Plexes including J . Although it is straightforward to exclude such a J with the condition $K_1(X) \neq \emptyset$, we allow to output J in addition to all possible (j, k) -MPCs.

Another point we have to note here is that every (j, k) -MPC is a subset of global j -core $core_j(V)$. In addition, since $core_j(V)$ consists of several connected components, and since our targets must be connected, we make the algorithm run for each connected component C of $core_j(V)$.

As the input graph is more sparse and j is larger, the size of global j -core is reduced much more, while the density increases. So, it becomes actually important to set the parameters k and j . The detailed discussion about this issue is given in Section 2.7. For setting parameters k and j , we assume a preferable size range $[n_1, n_2]$ and a density parameter τ . Then the con-

nection lower bound will be $j = n_1\tau$, and the disconnection upper bound $k = n_2 * (1 - \tau)$.

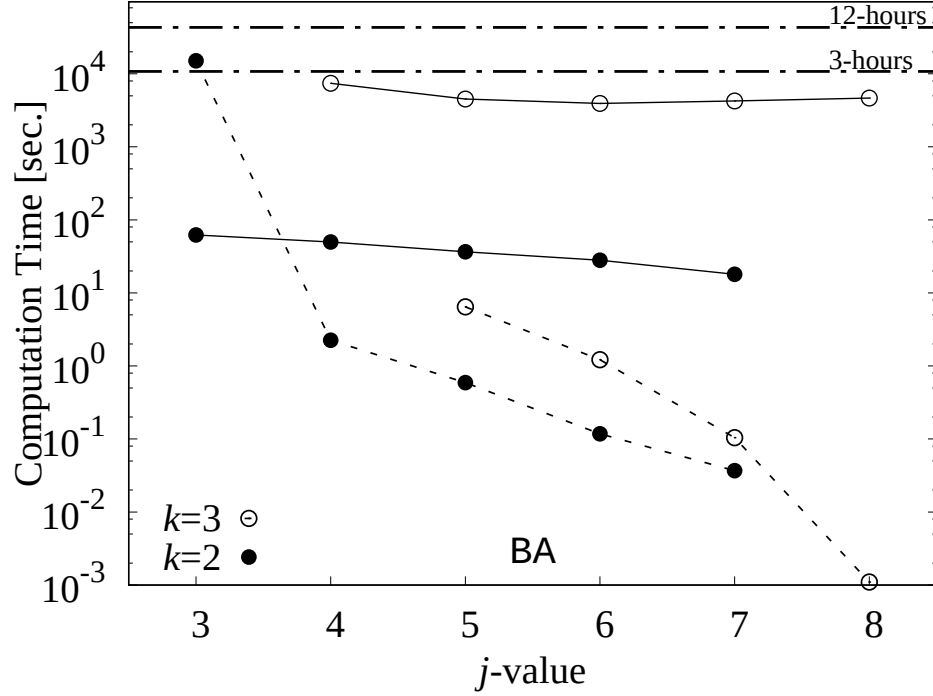


Figure 2.12: Computation times of BA

2.7 EXPERIMENTAL RESULTS

We present in this section our experimental results. The proposed system, referred to as JKMPCC, has been coded in Java and executed on a PC with Intel Core-i7 (1.7GHz) processor and 8GB memory. For several datasets, we observe computation times and quality of solutions as pseudo-cliques.

In order to verify practical efficiency of JKMPCC, we compare it with MaxKplexEnum, a state-of-the-art maximal connected k -Plex enumerator [BCK15][†]. It has been shown in

[†]Its source codes in Python have been kindly provided by the authors of [BCK15].

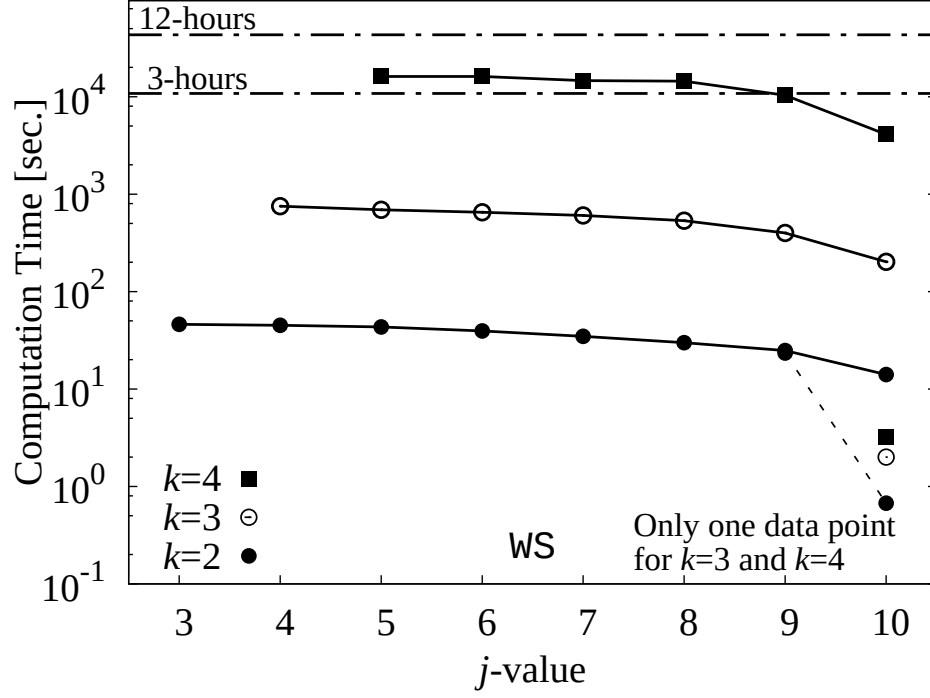


Figure 2.13: Computation times of WS

[BCK15] that MaxKplexEnum can enumerate maximal (connected) k -Plexes much faster than Pemp, a pioneer system for enumerating maximal k -Plexes [WP07][‡].

2.7.1 DATASETS

For our experimentation, we have prepared a collection of synthetic and real networks.

As a synthetic dataset, a scale-free network, referred to as BA, has been created based on *Barabási-Albert Model* [BA99] with the parameters $N = 10,000$ and $m = 10$, where N is the number of nodes and m the number of edges to be attached from each new node to existing nodes. In addition, we have also created a synthetic small-world network, referred

[‡]As has been mentioned in [BCK15; PYB12], Pemp has been known as the only one existing system for the task until recently.

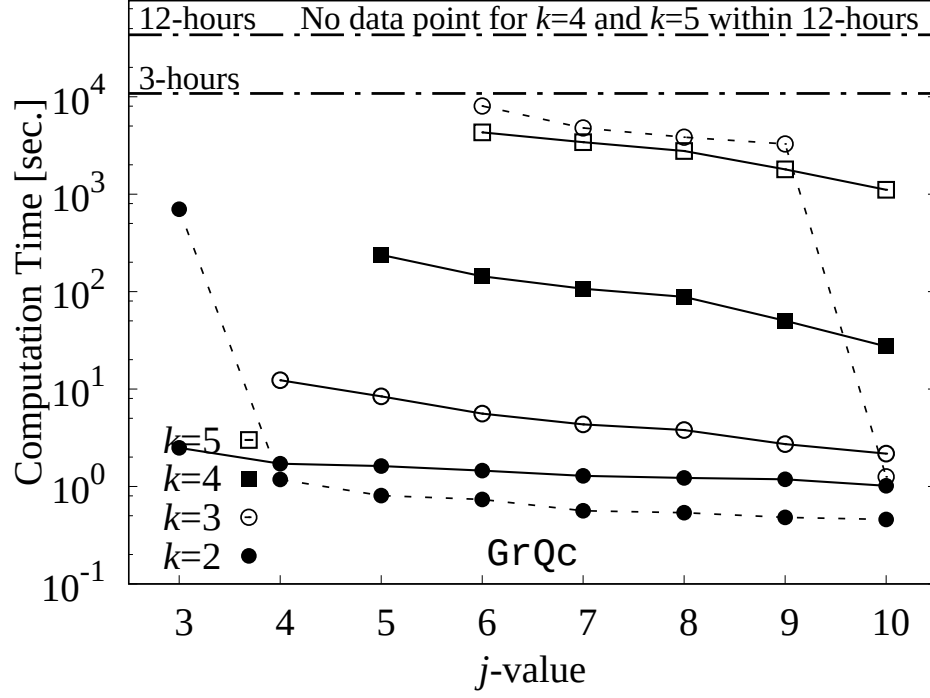


Figure 2.14: Computation times of GrQc

to as WS, based on *Watt-Strogatzs Model* [WS98] with the parameters $N = 50,000$, $K = 20$ and $p = 0.1$, where N is the number of nodes, K the initial degree of each node and p the probability of rewriting edges.

As real datasets, we have prepared benchmark networks, GrQc [LK14], DAYS [BM06], DBLP [LK14] and Google [LK14]. GrQc is a network representing scientific collaborations among authors of arXiv papers submitted to the category of General Relativity and Quantum Cosmology. If a paper has been co-authored by some researchers (as nodes), they are connected by edges each other. DAYS is a word co-occurrence network created from a collection of Reuter News Articles. If a pair of words have been co-occurred in a certain number of articles, they are connected by an edge. DBLP is also a collaboration network constructed from the DBLP Computer Science Bibliography. Authors as nodes are connected if they

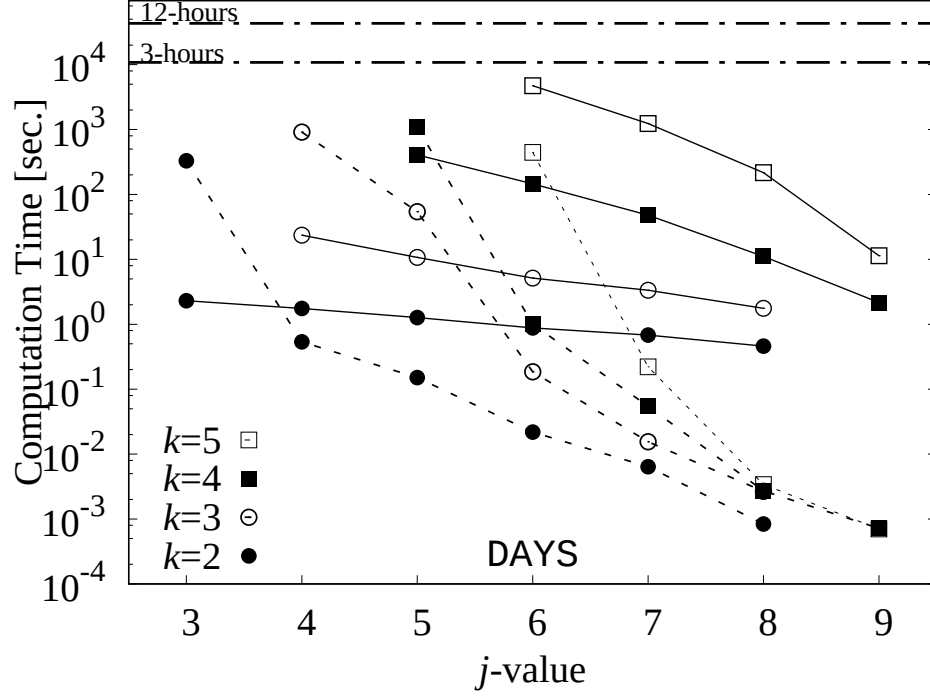


Figure 2.15: Computation times of DAYS

have published a paper together for registered journals or conferences. Google is a Web graph consisting of web pages and their hyperlinks. It has been released in 2002 by Google as a part of Google Programming Contest.

Scales of those networks are presented in Figure 2.5. Degree distributions of each network are illustrated in Figure [2.6, 2.7, 2.8, 2.9, 2.10, 2.11].

2.7.2 COMPUTATIONAL PERFORMANCE

As far as we know, there exists no algorithm devoted to enumerate all (j, k) -MPCs. In this sense, it is difficult to fairly compare JK MPC with others. However, our target k -MPCs can be obtained by any maximal k -Plex enumerator with j -coreness check. Therefore, we here observe computational performance of a state-of-the-art enumerator, `MaxKplexEnum`[BCK15],

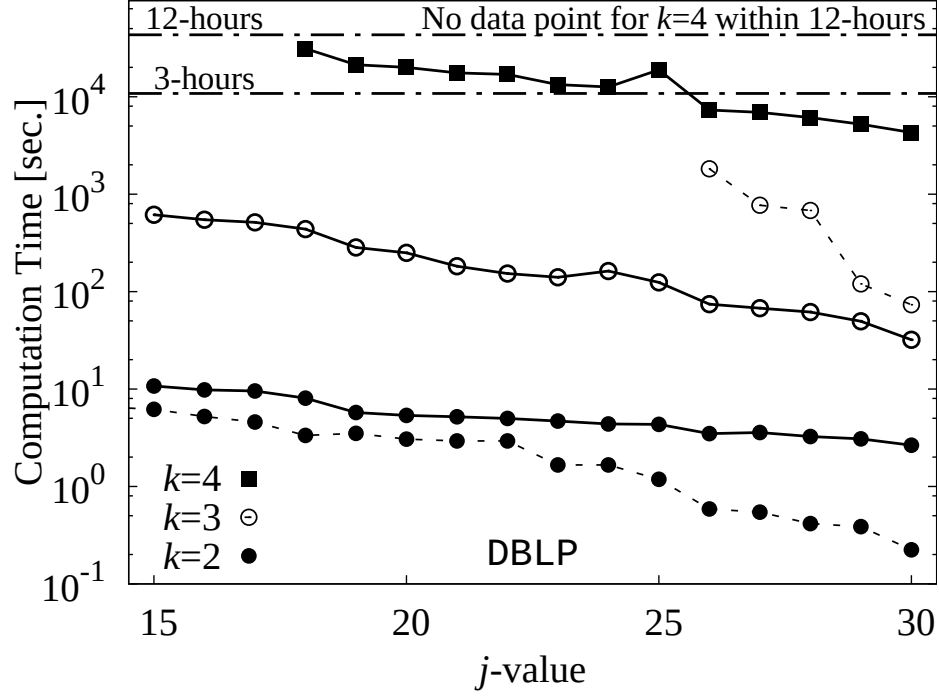


Figure 2.16: Computation times of DBLP

and our JKMPc in order to verify practical efficiency of our system.

Given a network, MaxKplexEnum runs in FPT(Fixed Parameter Tractable) incremental polynomial time with respect to k and can enumerate a designated number N of maximal k -Plexes. That is, in order to completely obtain our target (j, k) -MPCs with MaxKplexEnum , we have to give an adequate value of N which can provide a set of candidates completely including all of our solutions. It is, however, impossible to identify such a suitable N in advance. Therefore, in our comparison here, we first run JKMPc and identify the number of our solutions, say $\tilde{N}$. Then, we try to detect $\tilde{N}$ maximal k -Plexes by MaxKplexEnum . It is noted here that those k -Plexes extracted by MaxKplexEnum do not always contain all of our solution (j, k) -MPCs. That is, the value of $\tilde{N}$ can provide MaxKplexEnum the best-case scenario that k -Plexes found by MaxKplexEnum are fortunately equal to our solutions. At

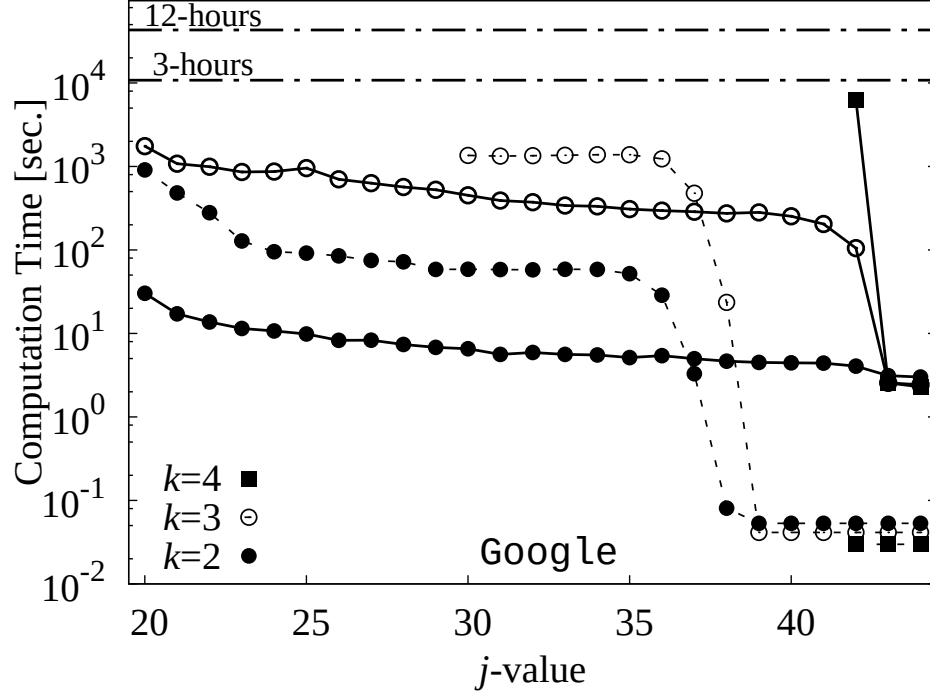


Figure 2.17: Computation times of Google

least, it can never bring any advantage to our JKMPc.

Figure [2.12,2.13,2.14,2.15,2.16,2.17,] shows computation times by both systems, where dotted lines are for MaxKplexEnum and solid lines for JKMPc. Data points for each k -value are distinguished by point types (e.g. $\bullet$, $\square$). For each computation, we have set a time limit of 12-hours. If a computation with some j and k has not been completed within the limit, the corresponding data point is missing in the figures. In our experimentation, we suppose that values of j are larger than k in order to exclude undesirable solutions with low density.

The larger a j -value is, the smaller the number of (j, k) -MPCs becomes. Then, the task of MaxKplexEnum

would be easier at a higher range of j because it is required to enumerate a smaller number of maximal k -Plexes. In fact, MaxKplexEnum can run much faster than ours at higher

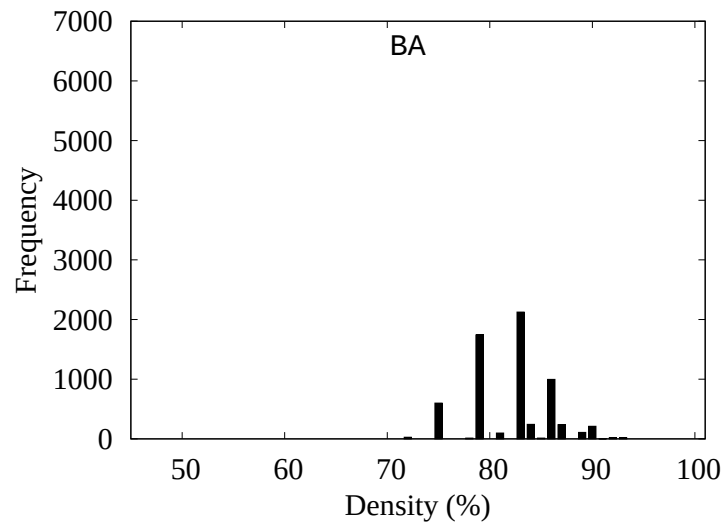
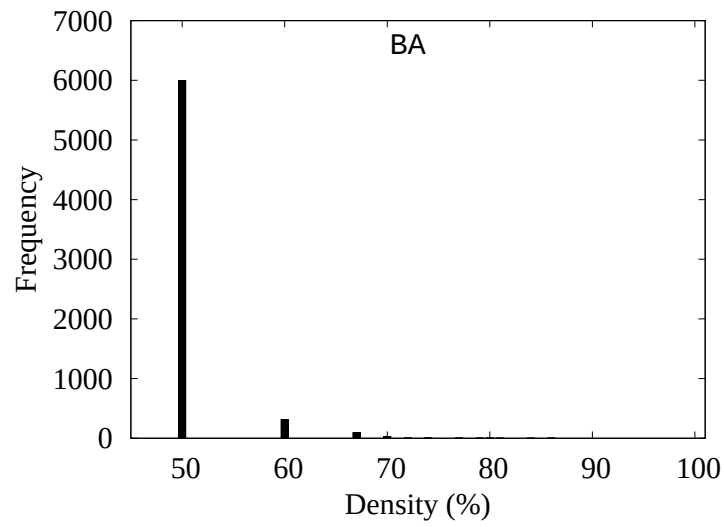
ranges of j . Its performance, however, suddenly gets worse, as j becomes smaller. In many cases, it has failed to complete enumerations within the time limit. For example, in case with $j = 3$ and $k = 2$ for BA where `MaxKplexEnum` has been required to extract 19, 138 maximal k -Plexes, it has not completed the computation within the limit.

The number of solutions we have to enumerate is also affected by a value of k . In most of the cases, it increases as k becomes larger. Actually, `MaxKplexEnum` can work well for a small k , say 2. In case of $k = 3$ or $k = 4$, however, we often observe low performance of `MaxKplexEnum` even for larger j -values. For example, for WS, the computation has been completed within the time limit only for $j = 10$. Moreover, no data point with $k = 4$ and $k = 5$ has been available for GrQC, and $k = 4$ for DBLP.

On the other hand, the performance of JKMPc is almost stable. Even in cases quite hard for `MaxKplexEnum`, our system can enumerate all solutions. In other words, JKMPc has an ability for extracting (j, k) -MPCs even with relatively small size. This is a remarkable advantage of our system because it is quite difficult for standard methods of graph clustering and partitioning to detect such small dense subgraphs. Thus, it is verified that JKMPc is an efficient and practical system for enumerating (j, k) -MPCs.

2.7.3 QUALITY OF SOLUTIONS AS PSEUDO-CLIQUEs

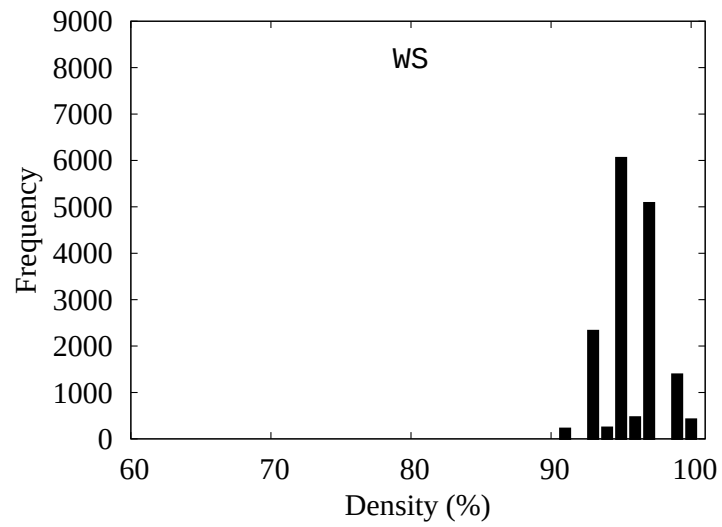
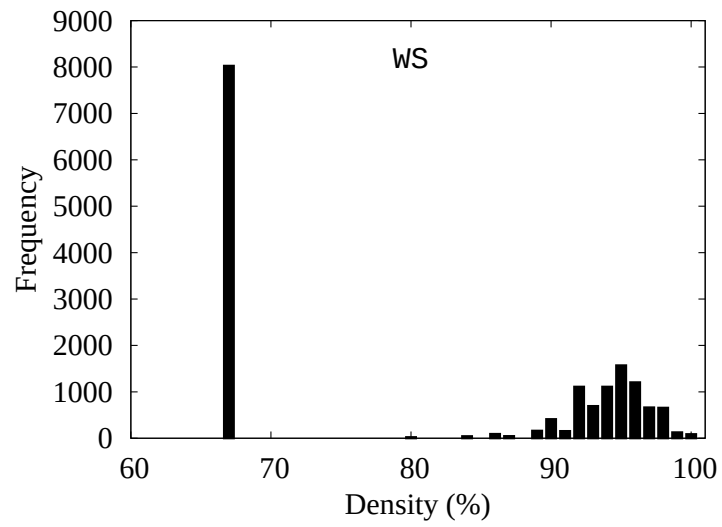
Since the notion of k -Plex has originally been proposed as a relaxation model of clique, any maximal k -Plexes with lower densities would be undesirable. We hopefully desire our solution k -Plexes to have sufficient densities. In order to examine quality of solution k -Plexes from the view point of pseudo-cliques, we observe density distributions of solutions obtained by `MaxKplexEnum` and JKMPc. As is similar to the experiments in Computational



	BA	
	Ave.	S. D.
MaxKplexEnum	0.509	0.035
JKMPC	0.816	0.038

Ave.: Average Density, S. D.: Standard Deviation of Density

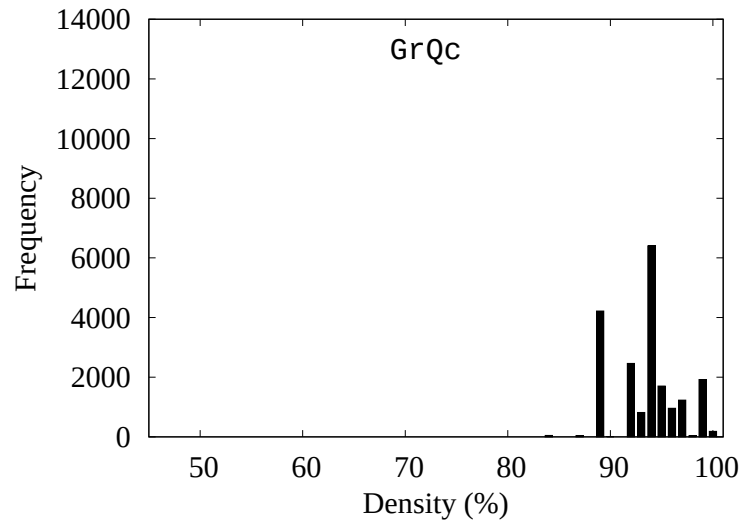
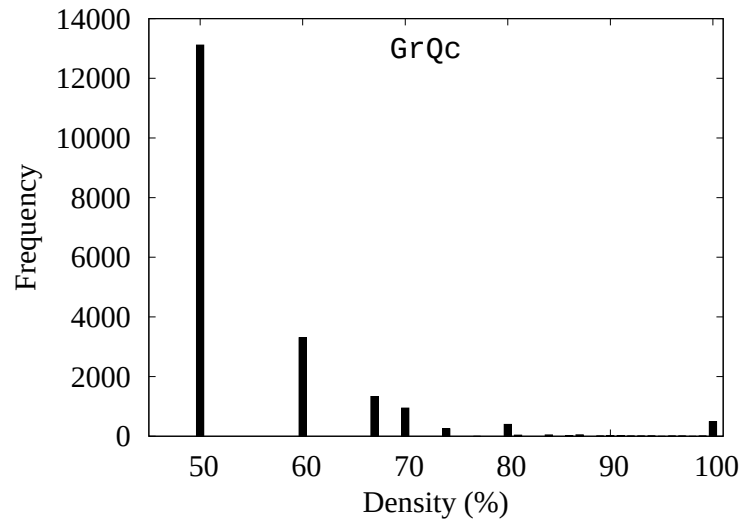
Figure 2.18: Density distributions of solution k -Plexes of BA



	WS	
	Ave.	S. D.
MaxKplexEnum	0.803	0.137
JKMPC	0.953	0.018

Ave.: Average Density, S. D.: Standard Deviation of Density

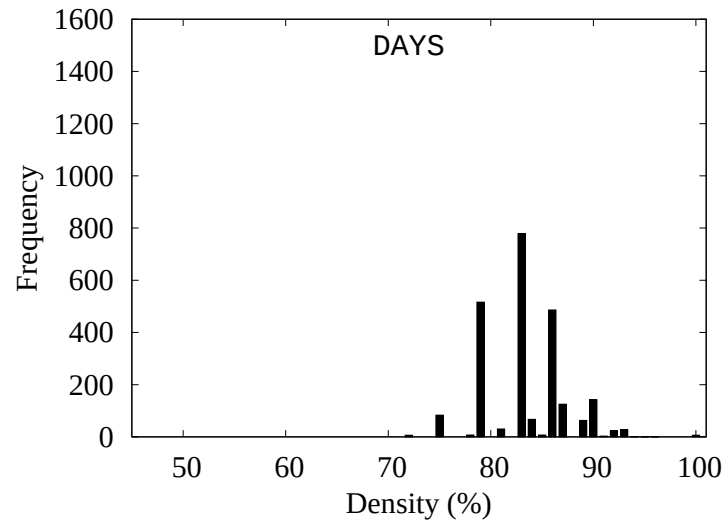
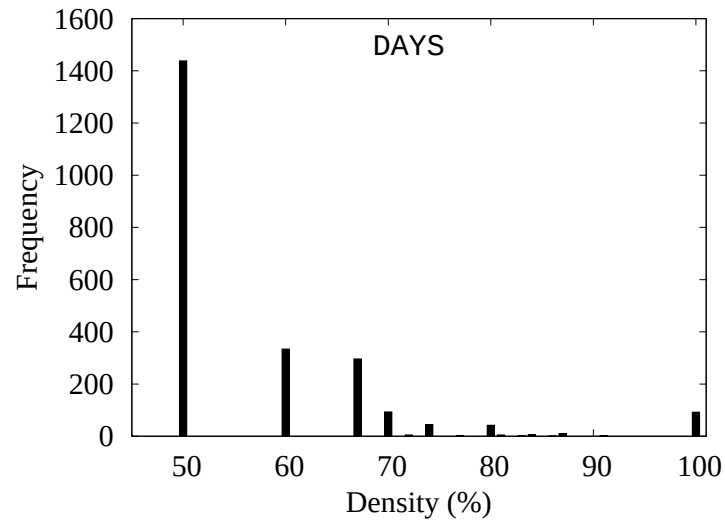
Figure 2.19: Density distributions of solution k -Plexes of WS



	GrQc	
	Ave.	S. D.
MaxKplexEnum	0.562	0.107
JKMPC	0.933	0.030

Ave.: Average Density, S. D.: Standard Deviation of Density

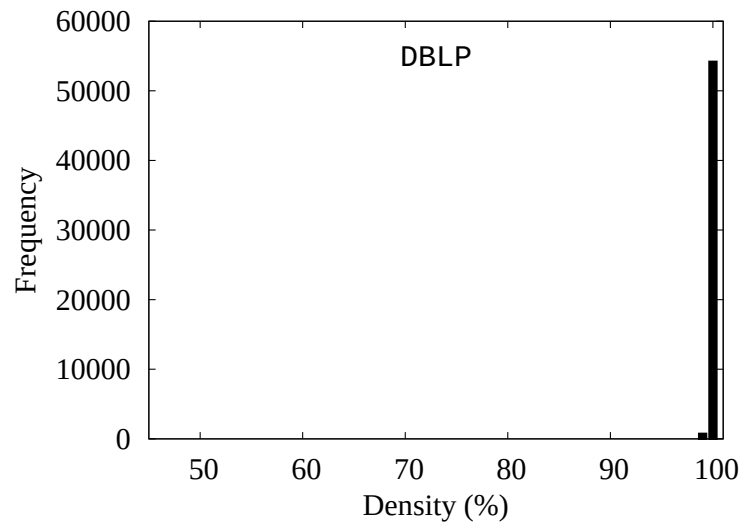
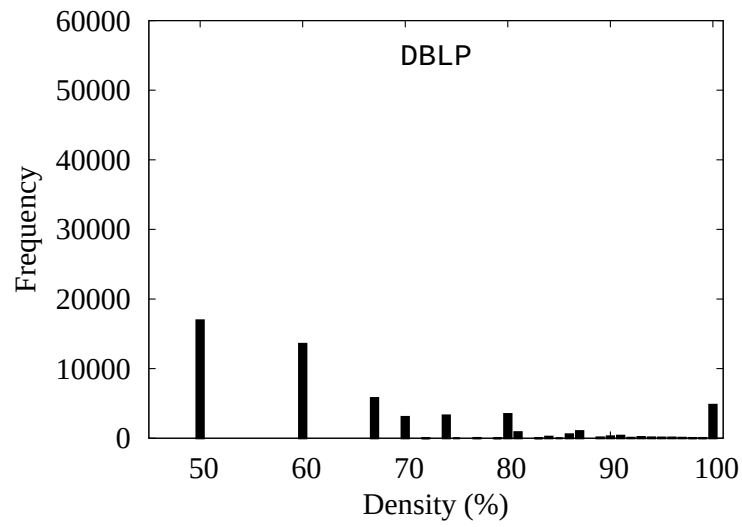
Figure 2.20: Density distributions of solution k -Plexes of GrQc



	DAYS	
	Ave.	S. D.
MaxKplexEnum	0.576	0.120
JKMPC	0.829	0.039

Ave.: Average Density, S. D.: Standard Deviation of Density

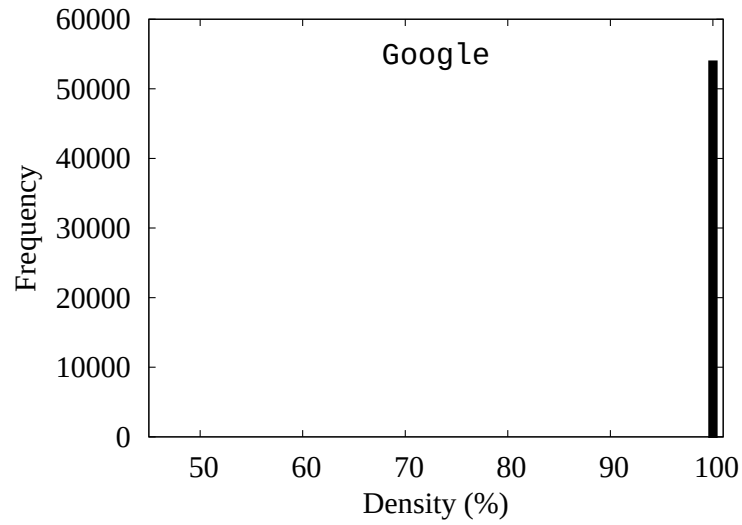
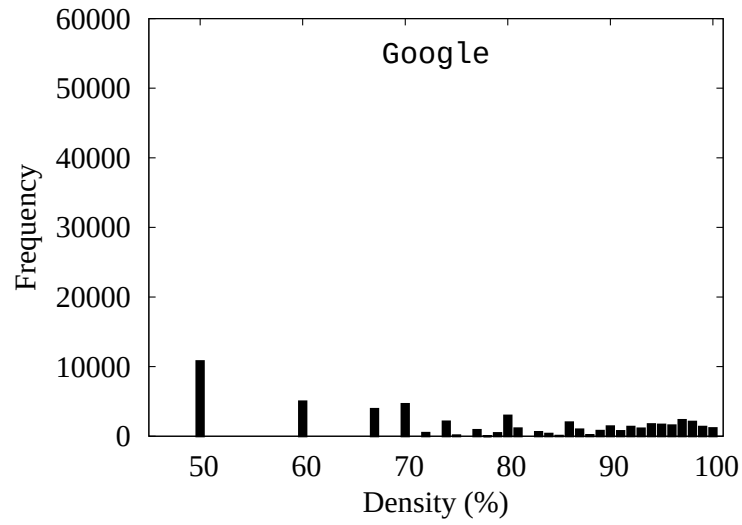
Figure 2.21: Density distributions of solution k -Plexes of DAYS



	DBLP	
	Ave.	S. D.
MaxKplexEnum	0.657	0.154
JKMPc	0.992	0.002

Ave.: Average Density, S. D.: Standard Deviation of Density

Figure 2.22: Density distributions of solution k -Plexes of DBLP



	Google	
	Ave.	S. D.
MaxKplexEnum	0.750	0.170
JKMPc	0.992	0.001

Ave.: Average Density, S. D.: Standard Deviation of Density

Figure 2.23: Density distributions of solution k -Plexes of Google

Performance, for each network, given a pair of j and k , we first enumerate (j, k) -MPCs by JKMPc. Assuming the number of obtained solutions to be $\tilde{N}$, we then try to enumerate $\tilde{N}$ connected maximal k -Plexes by MaxKplexEnum. For the solution k -Plexes enumerated by each system, we observe distribution of their densities.

Density distributions of solutions for each network are presented in Figure [2.18,2.19,2.20,2.21,2.22,2.23]. For each network, the upper figure is for distributions by MaxKplexEnum and the lower by JKMPc. It is obvious that the distributions are quite different for each network. In order to discuss more precisely, we also summarize averages and standard deviations of the densities in the figure.

For BA, although the standard deviations by both systems are low and almost the same, the average, 0.816, by JKMPc is much better than 0.509 by MaxKplexEnum.

For WS, the average of 0.803 by MaxKplexEnum seems fine. The actual densities are, however, a bit apart from the average, showing the standard deviation of 0.137. On the other hand, the densities given by JKMPc are distributed around a high average of 0.953 with a very low standard deviation of 0.018.

For GrQC, the average of 0.562 by MaxKplexEnum is low and the actual densities are in a wide range from 0.5 to 1.0, whereas the densities by JKMPc give a high average and a low deviation. A similar behavior can be observed for DAYS.

For DBLP and Google, the densities by MaxKplexEnum are a bit low, showing 0.657 and 0.750 on average, respectively. Moreover, the standard deviations are also high as 0.154 and 0.170. The densities by JKMPc are distributed quite differently. For both networks, we observe very high average densities above 0.99 and very low standard deviations nearly 0.0.

It is obvious that solutions by JKMPc have sufficient densities. All of them would be regarded as pseudo-cliques without any doubt. Solutions by MaxKplexEnum, however, include many maximal k -Plexes with relatively low densities. In other words, quality of solutions by MaxKplexEnum is unstable. Thus, JKMPc can practically work well as an effective pseudo-clique detector.

2.7.4 EFFECTIVENESS OF j -CORENESS CONSTRAINT

Our (j, k) -MPC is defined as a maximal k -Plex which satisfies the j -coreness constraint. The main purpose of imposing the constraint is to exclude not densely-connected pseudo-cliques. In order to verify effectiveness of the constraint, we here observe the number of maximal k -Plexes and that of (j, k) -MPC. The difference between them gives the number of undesirable maximal k -Plexes which we cannot regard as densely-connected ones.

Since there exist in general a huge number of maximal k -Plexes even in a network with moderate size, enumerating all maximal k -Plexes in a large network like DBLP or Google is quite impractical. We, therefore, present our results for WS and GrQc. More precisely speaking, for the set of (j, k) -MPC, we try to enumerate (count) maximal k -Plexes with size larger than j because the minimum size of (j, k) -MPC is given by $j + 1$ (for any k). For the enumeration, we have implemented a simple extended version of Pemp [WP07] with a branch-and-bound pruning based on the minimum size of the solutions.

Figure 2.25 shows the numbers of maximal k -Plexes and (j, k) -MPCs for each network with various j and k values. In the figures, the former is represented by dotted lines and the latter by solid lines. For each j and k , the difference between the former and the latter gives the number of maximal k -Plexes which cannot satisfy the j -coreness constraint. In order

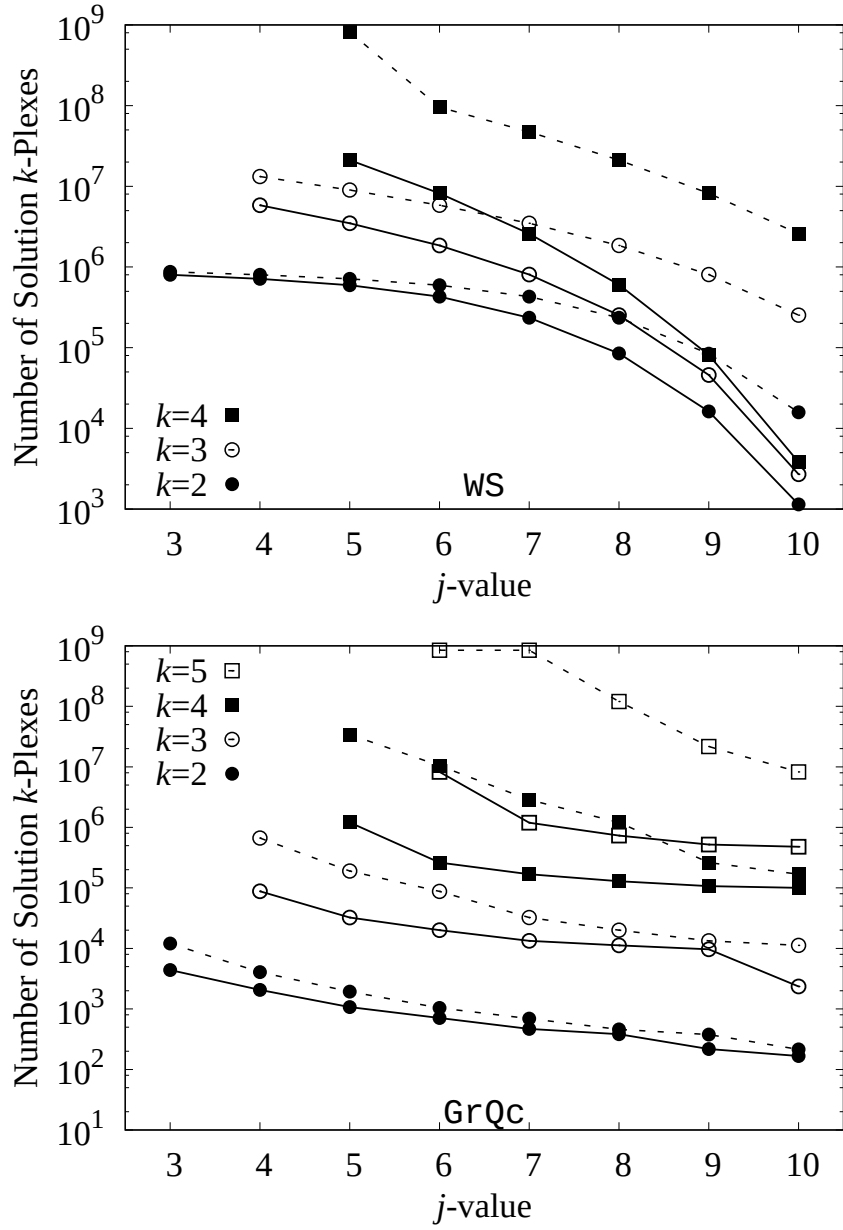


Figure 2.24: Numbers of maximal k -Plexes and (j, k) -MPCs

to clarify the differences, we separately show them in the figure. The figures show that the differences almost coincide the numbers of maximal k -Plexes particularly for larger values

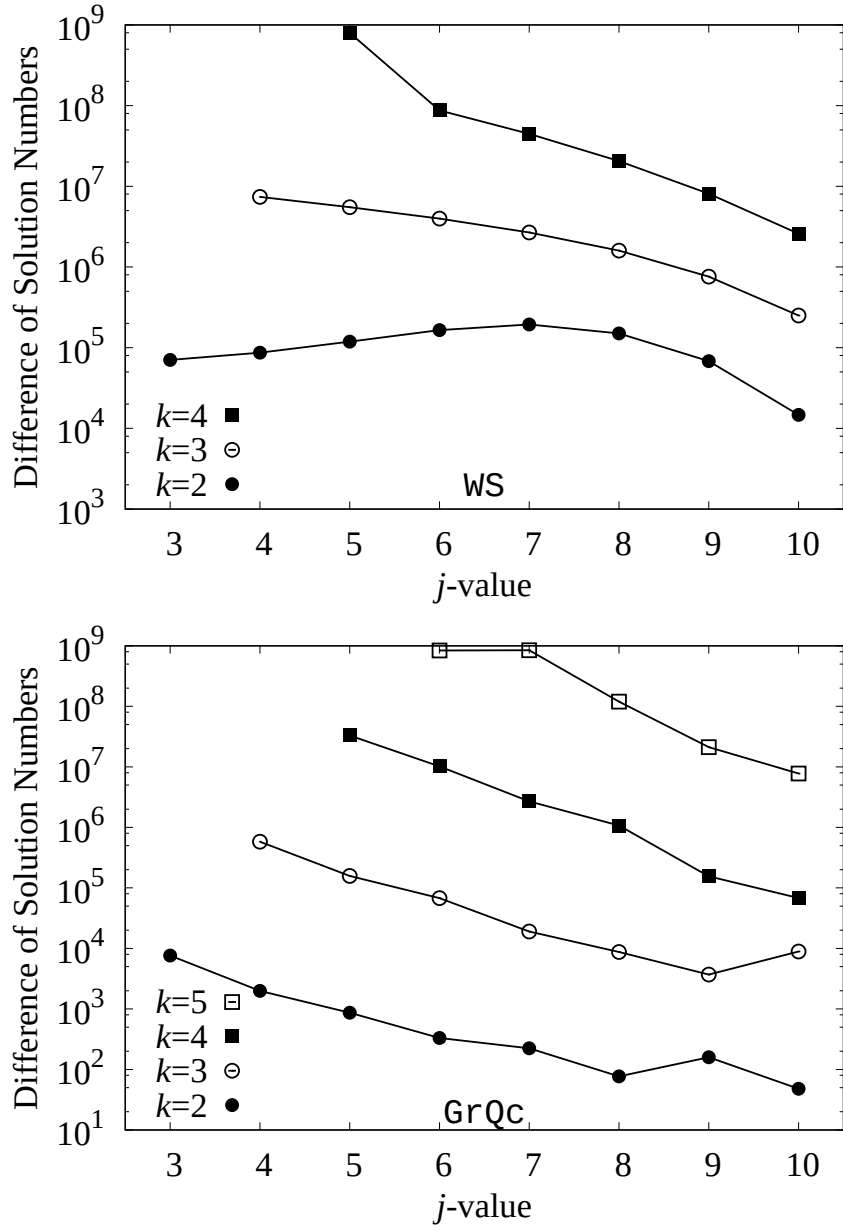


Figure 2.25: Difference between numbers of maximal k -Plexes and (j, k) -MPCs

of k . This reveals a fact that most of the maximal k -Plexes cannot be regarded as densely-connected ones. Our j -coreness constraint can exclude a huge number of those undesirable

maximal k -Plexes. As has already been discussed, since our algorithm enjoys several search control mechanisms based on the constraint, a large number of useless search branches (formations) not arriving at solutions can be pruned in our search process.

It should be noted here that from the definition, every maximal k -Plex with size no less than $j + k$ is always a (j, k) -MPC. For any maximal k -Plex not densely-connected, its size is in the interval of $(j, j + k)$. The above fact, therefore, implies that most of the maximal k -Plexes are considered as relatively small or medium-sized ones. If our task is to detect densely-connected maximal k -Plexes with larger size, standard methods for graph clustering or partitioning would work well. They, however, would not be suitable when we try to obtain such k -Plexes with moderate size because those clusters are usually out of their targets. As has been observed just above, our algorithm can efficiently enumerate all of the (j, k) -MPC including small or medium-sized ones. Non-large clusters could be valuable in the sense that they are often invisible and difficult to find. Without j -coreness constraint, we can never enjoy the ability to detect such potential candidates.

2.7.5 PRACTICAL USEFULNESS OF (j, k) -MPCs

In our formulation, a densely-connected pseudo-clique is defined as a maximal k -Plex satisfying j -coreness constraint. That is, our solution k -Plexes we try to enumerate are supposed to belong to the class of maximal k -Plexes. Theoretically speaking, however, there must exist a k -Plex which is not maximal but can satisfy j -coreness constraint. An example of such a (non-maximal) k -Plex is illustrated in Figure 2.26.

For the graph, we try to enumerate $(4, 3)$ -MPCs, that is, maximal 3-Plexes satisfying 4-coreness constraint. It is easy to find two maximal 3-Plexes, $A = \{a, 1, 2, 3, 4, 5\}$ and $B =$

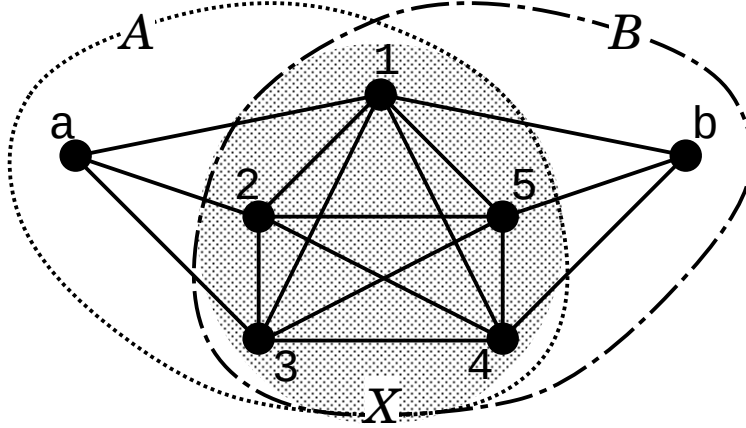


Figure 2.26: Example of maximal j -cored k -Plex

$\{b, 1, 2, 3, 4, 5\}$. However, we see $\deg_A(a) = 3$ and $\deg_B(b) = 3$, that is, none of them satisfy the constraint. Thus, no $(4, 3)$ -MPC can be detected in this example.

On the other hand, let us consider a subset of A (also B), $X = \{1, 2, 3, 4, 5\}$. From the anti-monotonicity of k -Plex, X is a 3-Plex. Moreover, we have $\deg_X(x) = 4$ for any $x \in X$. That is, X is a 3-Plex which satisfies 4-coreness constraint even though it cannot be a solution according to our current formulation.

One might claim that such an X should be detected as a solution because it actually satisfies both constraints, DUB and CLB, we impose. This claim brings us an alternative formulation of densely-connected pseudo-cliques, *maximal j -cored k -Plexes* (max- (j, k) -PC, for short). More precisely, we can define a max- (j, k) -PC as a maximal set of vertices, X , which forms a k -Plex and satisfies the j -coreness constraint. It is noted again that X is not necessarily a maximal k -Plex.

Since a (j, k) -MPC is always a max- (j, k) -PC, the class of (j, k) -MPCs is subsumed by that of max- (j, k) -PCs. As has been mentioned above, in our current framework, we are

WS		$ \mathcal{M}_{miss} $	$ \mathcal{M}_{j,k} $	$\frac{ \mathcal{M}_{miss} }{ \mathcal{M}_{j,k} }$
$k = 3$	$j = 4$	0	5, 821, 233	0.000000
	$j = 5$	0	3, 483, 608	0.000000
	$j = 6$	0	1, 848, 536	0.000000
	$j = 7$	0	806, 784	0.000000
	$j = 8$	16	253, 356	0.000063
	$j = 9$	50	45, 888	0.001090
	$j = 10$	33	2, 706	0.012195
$k = 4$	$j = 5$	0	21, 131, 619	0.000000
	$j = 6$	0	8, 178, 200	0.000000
	$j = 7$	0	2, 591, 030	0.000000
	$j = 8$	7	602, 300	0.000012
	$j = 9$	96	81, 546	0.001177
	$j = 10$	304	3, 815	0.079685

Table 2.2: Ratio of $|\mathcal{M}_{miss}|$ to $|\mathcal{M}_{j,k}|$ of WS

GrQc		$ \mathcal{M}_{miss} $	$ \mathcal{M}_{j,k} $	$\frac{ \mathcal{M}_{miss} }{ \mathcal{M}_{j,k} }$
$k = 3$	$j = 4$	16	87, 786	0.000182
	$j = 5$	6	32, 407	0.000185
	$j = 6$	4	20, 014	0.000200
	$j = 7$	0	13, 364	0.000000
	$j = 8$	0	11, 247	0.000000
	$j = 9$	0	9, 658	0.000000
	$j = 10$	0	2, 355	0.000000
$k = 4$	$j = 5$	18	1, 204, 135	0.000015
	$j = 6$	4	262, 717	0.000015
	$j = 7$	1	168, 025	0.000006
	$j = 8$	0	128, 942	0.000000
	$j = 9$	0	107, 120	0.000000
	$j = 10$	0	100, 080	0.000000
$k = 5$	$j = 6$	0	8, 235, 436	0.000000
	$j = 7$	2	1, 192, 371	0.000002
	$j = 8$	0	731, 706	0.000000
	$j = 9$	0	521, 364	0.000000
	$j = 10$	0	478, 779	0.000000

Table 2.3: Ratio of $|\mathcal{M}_{miss}|$ to $|\mathcal{M}_{j,k}|$ of GrQc

missing every $\text{max-}(j, k)$ -PC which is not a maximal k -Plex. Therefore, the class of $\text{max-}(j, k)$ -PCs would be preferable to that of (j, k) -MPCs because we have more chances to find pseudo-cliques which might be valuable. From the computational point of view, however, the class of (j, k) -MPCs is more advantageous. It is due to the lack of anti-monotonicity of j -coreness. For a set of vertices satisfying the j -coreness constraint, its subset does not necessarily satisfy the constraint. This simple fact requires us to pay some expensive cost for detecting maximal set of vertices with j -coreness. Thus, (j, k) -MPCs and $\text{max-}(j, k)$ -PCs have their strong and weak points. Nevertheless, we argue that the notion of (j, k) -MPCs is a reasonable formulation of pseudo-cliques from the practical viewpoint. In order to make the argument more convincing, we observe the number of $\text{max-}(j, k)$ -PCs we actually miss in our enumeration process for (j, k) -MPCs.

From the definitions, any maximal k -Plex with size no less than $j + k$ satisfies the j -coreness constraint, that is, it can always be regarded as a $\text{max-}(j, k)$ -PC. Moreover, any set of vertices with size no larger than j can never satisfy the constraint. Therefore, for every $\text{max-}(j, k)$ -PC we cannot detect, its size is in the interval of $(j, j + k)$. Particularly, such a $\text{max-}(j, k)$ -PC must appear as a proper subset of a maximal k -Plex not satisfying the j -coreness constraint. Based on these theoretical properties, we can count the number of missing $\text{max-}(j, k)$ -PCs for given j and k . It should be noted here that we never miss any $\text{max-}(j, k)$ -PC in case of $k = 2$. This is because any $\text{max-}(j, k)$ -PC not satisfying the constraint must consist of at most $j + 1 (= j + k - 1)$ vertices and hence the size of its proper subset is out of the interval $(j, j + 2)$ (that is, $[j + 1, j + 1]$).

Let $\mathcal{M}_{j,k}$ be the set of (j, k) -MPCs and $\mathcal{M}_{\text{miss}}$ the set of missing $\text{max-}(j, k)$ -PCs. From the subsumption relation, the set of all $\text{max-}(j, k)$ -PCs, denoted by $\mathcal{M}$, can be given as

$\mathcal{M} = \mathcal{M}_{j,k} \cup \mathcal{M}_{miss}$. Therefore, if the ratio of $|\mathcal{M}_{miss}|$ to $|\mathcal{M}_{j,k}|$ is sufficiently low (hopefully close to 0.0), we can almost identify the class of (j, k) -MPCs with that of $\text{max-}(j, k)$ -PCs, missing quite few solutions. In such a case, (j, k) -MPCs which we can compute efficiently would be considered as a reasonable and practical formalization of pseudo-cliques.

Table 2.2 and 2.3 lists the ratio of $|\mathcal{M}_{miss}|$ to $|\mathcal{M}_{j,k}|$ for various j and k . Since the size interval of $\text{max-}(j, k)$ -PCs in $\mathcal{M}_{miss}$ is defined as $(j, j + k)$, it seems that we would miss more $\text{max-}(j, k)$ -PCs as k becomes larger. For the synthetic network WS, such a trend can be observed for larger values of j . For smaller j , however, we miss very few $\text{max-}(j, k)$ -PC. On the other hand, for the real network GrQC, numbers of missing $\text{max-}(j, k)$ -PCs are quite low, almost 0, for the whole range of j . These different observations would be caused by the size distributions of larger solutions ((j, k) -MPCs).

More concretely speaking, in WS, solution sizes are distributed over a narrow range from $\mathbf{I1}$ to $\mathbf{I4}$ (for $j = \mathbf{10}$ and $k = 4$). In contrast, in GrQC, we observe a wide range of solution sizes from $\mathbf{I1}$ to $\mathbf{46}$ (for $j = \mathbf{10}$ and $k = 4$). Note here that as a simple fact, a larger k -Plex can subsume more k -Plexes. In case of $j = \mathbf{10}$ and $k = 4$, moreover, the size of $\text{max-}(j, k)$ -PCs we miss must be in the interval of $[\mathbf{I1}, \mathbf{I3}]$. Therefore, larger solutions in GrQC would probably make most of the k -Plexes in the interval non-maximal. As the result, there are quite few $\text{max-}(j, k)$ -PCs we are missing. On the other hand, for WS, since there exist a small number of (j, k) -MPCs and all of them are relatively small, k -Plexes in the interval would have less chance to be subsumed. Hence, we might sometime miss detecting a certain number of $\text{max-}(j, k)$ -PCs.

While those cases like WS are undesirable, it would not frequently to arise similar cases in real world networks. In actual networks, such a narrow range of solution size is rarely

happened. As has been stated previously, furthermore, we emphasize an ability of our algorithm to detect densely-connected pseudo-cliques with relatively small size (for smaller j -values). Therefore, the authors consider our (j, k) -MPCs to be a practically useful formalization of densely-connected pseudo-cliques.

2.8 CONCLUSIONS

In this chapter, we designed an efficient complete algorithm for enumerating (j, k) -MPCs. For efficient computation, we discussed several search mechanisms which can effectively prune many useless search nodes. Our experimental results showed the algorithm can work well as a practical tool for extracting densely-connected pseudo cliques in large networks.

ACKNOWLEDGMENTS

The authors of [BCK15] have kindly provided us their program codes of `MaxKplexEnum`. We would like to sincerely appreciate their kindness.

*This work is originally published at Machine
Learning and Data Mining in Pattern Recognition
(MLDM13) [Zha+13] and licensed for Thesis/Disserta-
tion use by Springer. License Number: 4286930037913.*

3

Associative Search by Shifting Concepts via Bridges

3.1 INTRODUCTION

In the studies of Information Retrieval, a kind of technique, called “Associative Search” [Tak+00], is sometimes used to shift queries during the search processes for documents or

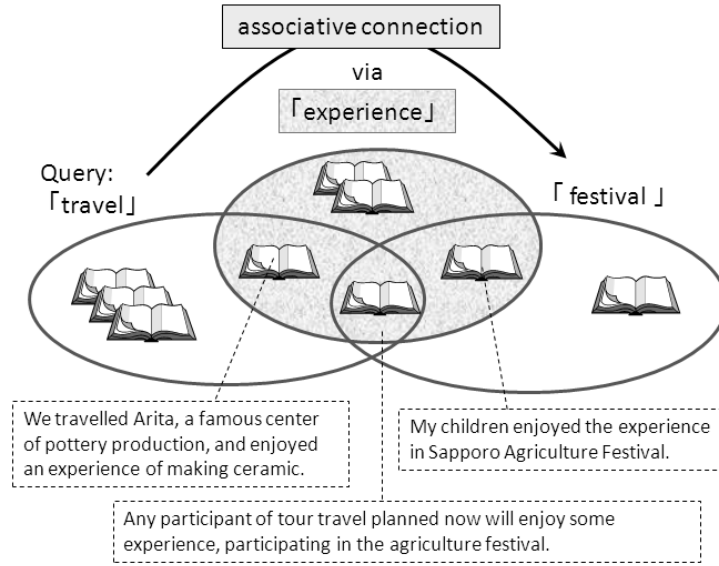


Figure 3.1: Example of Associative Connection of Concepts

messages. The purpose of shifting queries is to suggest feature terms and documents not being properly expressed by the initial query, reminding users of other features which may be related to user's interests or intention. This research aims at formalizing those processes in terms of formal concepts [GW12] and to provide an efficient mining procedure to enumerate target concepts associative with the initial one, enjoying the structures of formal concept lattices.

Fig. 3.1 illustrates an example of associative connection between two concepts, "travel" and "festival", where the former is considered as the initial one, and ovals in the figure denote sets of documents with the feature terms as "travel". At least some documents about "travel" involve another feature term, "experience" for instance, that may play a role of bridge for "travel" and "festival". In other words, the associative connection is supported by such documents about "travel" with the term "experience" that form a sub-concept of

“travel”. A bridge concept “experience” guiding us to “festival” can be constructed by dropping “travel” from the sub-concept of “travel” with “experience”. The dropping operation is performing generalization of concepts. As is well known, the framework of formal concepts is suited to perform such a construction based on Galois connection between feature terms as attributes and documents as objects. Particularly, specialization (generalization) to obtain sub-concepts (super-concepts) is naturally represented in concept lattices.

In the problem of Web navigation [DDEo8], a similar idea for navigating concept search in a concept lattice is presented, guiding related concepts along a lattice structure. Namely the basic operations are those for generalization, specialization and for taking sibling concepts in the lattice. On the other hand in this research, under a bridge concept, which is also constructed by specialization and generalization but is much more constrained as we see later, target concepts reachable from the initial one are never our solution as long as they are not related to some bridge concept. Thus we make stronger structural constraint for the targets.

Major issues we have to solve in making bridge concepts and associative connections based on them are listed as follows.

Selection of Candidate Bridges (SCB) :

There exist many sub-concepts of initial one. Even if we choose some of them, we have many ways to generalize those sub-concepts to get more general super-concepts as bridges.

Selection of Target Concepts (STC) :

Suppose some bridge concept is successfully constructed from the initial one. Even under such an assumption, several possible concepts associated with the initial one

via the bridge are there. Some criterion or condition checking which target concepts are potentially associative under the presence of bridge will be needed.

The STC problem can be solved more directly than SCB by enumerating candidate concepts meeting a similarity constraint that they must be similar to the initial concept w.r.t. a given bridge. Thus we regard STC as a mining task instead of selection task. In the case of Fig 3.1, although “travel” and “festival” may not be similar so much, their extents (the document sets) would become closer when we restrict documents to those of “experience”. Thus we evaluate how object sets are overlapping each other within the extent of bridge. Our similarity is conditioned by bridge concepts. Several similarity measures for formal concepts are proposed in the literatures. For instance, in [AB11], similarity between concepts is defined by considering the overlappingness of their feature sets and object sets at the same time. However, we need only similarity showing how the object sets are overlapping. For this reason, we use bond [Omio3], an extended Jaccard coefficient, to calculate the degree of overlappingness for object sets. In Section 3.6, we present a procedure to solve STC based on the bond measure conditioned by bridges.

On the other hand, SCB is more critical, since STC search completely depends on solutions of SCB. As is already discussed, bridges are constructed from document objects of some sub-concept of initial one. Some clustering methods, [GMW07; Car+09] for instances, may be applicable to have a family of object sets. From each object set, we would form an intent of bridge by examining and by selecting some of its characteristic feature terms. However, almost clustering methods disregard clusters not meeting their own clustering criterion. Consequently, possible candidate bridges are restricted.

To the contrary, in this research, we regard the variation of possible clusters of document

objects. As we have more candidate object sets from which bridges are constructed, users have more chances to become aware of them and to change their initial concepts based on them. It is however computationally hard to keep all of them because of a huge number of object subsets or sub-concepts of initial one.

In order to relax this kind of dilemma, we here propose to use other information about documents or messages, the information of writers or senders. With each object (document or message), we suppose just one person is associated. Then we can build a person-feature relation from the object-feature relation by merging objects each person has. The variation of features possessed by objects is aggregated to those persons have. As the number of persons is much smaller than the number of document objects, we examine person sets, instead of document sets, from which some bridge concepts are drawn more efficiently.

The extraction of bridges from person sets is performed in the corresponding concept lattice for the person-feature incident relation. In order to make the process more efficient, we introduce the following heuristic:

Mediator Heuristics:

When we wonder which direction of associative connection is better to be developed, we normally ask some persons who talk about various areas of documents and feature terms. We call here such persons mediators. As more number of features or topics they are talking about, we regard them as better mediators.

Using a membership function of fuzzy k -means clustering [Bez13], we define a probability distribution on features showing how a person is related to the features, and then calculate its entropy as the degree of mediator. Then, every person is ranked according to the entropy, and the relevance to the initial concept is tested by user. Some persons may

be inadequate from the user’s viewpoint, while another may be good conversely. We assign negative or positive signs to the former and the latter types of mediators, respectively. Then the target bridge concepts are required to cover positive mediators and not to cover negative ones. Consequently, the possible bridges are to be placed in a sub-lattice with the initial concept as its top and the least common generalization of positive mediators as its bottom. Using the search strategy presented in [LHO08], we show in Section 3.6 a very efficient search procedure to find candidate bridge concepts in the sub-lattice.

The reminder of this research is organized as follows. In the next section, we introduce some terminologies. Section 3 discusses a pair of incident relations as datasets we assume. Our user ranking method is presented in Section 4. In Section 5 and 6, we formalize our associative search and present its computation procedure, respectively. In our experimentation in Section 7, we apply our method to data of 500,000 tweets, where features are compressed as 155 clusters of 912 original feature terms (nouns) to avoid the problem of very sparseness of the data. Such compressed features are also used to build message-feature incident relation. As an instance, we show an associative connections between “smartphone” and “digital book” via “mobile” actually extracted. The paper is concluded in Section 8 with a summary and some future directions.

3.2 PRELIMINARIES

In this research, we are concerned with a notion of *concepts* in *Formal Concept Analysis* (FCA) [GW12]. Let $\mathcal{O}$ be a set of *objects* (or individuals) and $\mathcal{A}$ a set of *attributes* (or features). For a binary relation $R \subseteq \mathcal{O} \times \mathcal{A}$, A triple $(\mathcal{O}, \mathcal{A}, R)$ is called a *formal context*. If $(x, y) \in R$, we say that the object x has the attribute y . Then, for a set of objects $X \subseteq \mathcal{O}$, the

set of attributes associated with X is denoted by X' , that is, $X' = \{y \in \mathcal{A} \mid \forall x \in X, (x, y) \in R\}$, where “ $\prime$ ” is called a *derivation operator*. Similarly, for a set of attributes $Y \subseteq \mathcal{A}$, the set of objects sharing Y is denoted by Y' , that is, $Y' = \{x \in \mathcal{O} \mid \forall y \in Y, (x, y) \in R\}$.

For a pair of $X \subseteq \mathcal{O}$ and $Y \subseteq \mathcal{A}$, (X, Y) is called a *formal concept (FC)* under the formal context iff $X' = Y$ and $Y' = X$, where X and Y are called the *extent* and the *intent* of the concept, respectively. For concepts (X_i, Y_i) and (X_j, Y_j) , if $X_i \subseteq X_j$ (or $Y_i \supseteq Y_j$), then we say (X_i, Y_i) is a *sub-concept* of (X_j, Y_j) .

Since we are concerned with concepts in different formal contexts, the derivation operator in a context $\mathcal{C}$ is often explicitly denoted by $\prime^{(\mathcal{C})}$ or $\prime\prime^{(\mathcal{C})}$.

3.3 PERSON-TERM RELATION AND DOCUMENT-TERM RELATION

In order to formalize our associative search, we assume a *person-term* relation and a *document-term* relation as formal contexts.

Let P be a set of *persons* and V a set of *terms* or *words* as a vocabulary. It is assumed that a *document* is written at the vocabulary V by a person in P . For a document d , the set of terms appeared in d is referred to as $terms(d)$ and the person by whom d is written as $person(d)$.

Given a set of documents D , we can define a document-term relation $R_D \subseteq D \times V$, where $(d, t) \in R_D$ if and only if $t \in terms(d)$. It is clearly represented as a formal context $\mathcal{D} = (D, V, R_D)$.

In addition, a person-term relation $R_P \subseteq P \times V$ can also be defined from D . For the document set D , the set of documents written by a person $p \in P$ is denoted by D_p , that is, $D_p = \{d \in D \mid person(d) = p\}$. Then, for a person p , the set of terms used by p in some document is given by $T_p = \cup_{d \in D_p} terms(d)$. Based on T_p , we can define $R_P \subseteq P \times V$

as $(p, t) \in R_p$ if and only if $t \in T_p$. Thus we have a corresponding formal context $\mathcal{P} = (P, V, R_p)$.

3.4 RANKING PERSONS BASED ON MEDIATOR LEVEL

Let P be a set of persons. We assume that each person in P is assigned a *mediator level*. Intuitively speaking, if a person is related to various kinds of topics, it seems possible for us to associatively access to several topics via the person. In this sense, it would be reasonable to consider that such a person is a good mediator for associative search and given a higher mediator level. On the other hand, if a person is concerned with some particular topic, he/she is assigned a lower mediator level.

3.4.1 MEDIATOR LEVEL

We consider a person to be a good mediator, if he/she is related to a more number of topics than the others. In order to find such a person, assuming a cluster of terms (words) to be a *topic*, we evaluate a user's mediator level based on how closely related to topics the user is.

For a person-term context $\mathcal{P} = (P, V, R_p)$ and a document-term context $\mathcal{D} = (D, V, R_D)$, the mediator level of a person in P is computed as follows:

1. The person context $\mathcal{P}$ is projected by preserving all terms with *TF-IDF* values greater than $(\max + \min) * \alpha$, where *alpha* is a control parameter, and $\max$ and $\min$ are the maximum and minimum values of *TF-IDF*. The projected context is denoted by $\tilde{\mathcal{P}} = (P, \tilde{V}, \tilde{R}_p)$.
2. Based on $\tilde{V} \subseteq V$, the document context is also projected. The projected context is

denoted by $\tilde{D}$ and defined as $\tilde{D} = (D, \tilde{V}, \tilde{R}_D = (R_D \cap (D \times \tilde{V})))$. Then we consider its corresponding matrix $M_{\tilde{D}} = (\vec{w}_1, \dots, \vec{w}_{|\tilde{V}|})$, where $\vec{w}_i^T = (d_{i1}, \dots, d_{i|\mathcal{D}|})$ and for each j , if $(j, i) \in \tilde{R}_D$, then $d_{ij} = 1$ and otherwise $d_{ij} = 0$.

3. The vectors $\vec{w}_i$ ($1 \leq i \leq |\tilde{V}|$) are clustered into several groups of terms. Each cluster is regarded as a topic and is represented by its central vector.
4. For the (projected) person context $\tilde{P}$, we consider its corresponding matrix $M_{\tilde{P}} = (\vec{p}_1^T, \dots, \vec{p}_{|P|}^T)^T$, where $\vec{p}_i^T = (d_{i1}, \dots, d_{i|\mathcal{D}|})$ and for each j , if $(i, j) \in \tilde{R}_P$, then $d_{ij} = 1$ and otherwise $d_{ij} = 0$.
5. In order to transform each person vector $\vec{p}_i^T$ into those in the document space, we apply *Singular Value Decomposition* (SVD) to $M_{\tilde{P}}$.
6. Degree of relatedness of a person p to a topic t is given by the distance measure used in fuzzy K-means in [Bez13],

$$R(p, t) = \frac{1}{\sum_{t_i \in \mathcal{T}} \left(\frac{\text{dist}(p, t)}{\text{dist}(p, t_i)} \right)^{\frac{2}{m-1}}}$$

where $\mathcal{T}$ is the set of topics, m is a control parameter for fuzzy level and $\text{dist}(p_i, t_j)$ is Euclidean Distance in the document space between a person p_i and a topic t_j .

7. Regarding the vector of $R(p, t_i)$ -value for each topic t_i as a probability distribution, the mediator level of the person p is given as the entropy of the distribution.

For each cluster i and j , we believe that $\langle c_i, c_j \rangle \rightarrow 0$ by using the center of similar words. In other words, each topic has no relation with others. If one user has a more gen-

eral interest, he/she will get involved into a more number of separated topics. Entropy will help us to judge the variance of a user's interest.

By sorting the persons in P in descending order of their mediator levels, we can define a ranked list of persons in P . For each person $p \in P$, the mediator rank and the mediator level of p are referred to as $rank(p)$ and $level(p)$, respectively.

3.4.2 CLUSTERING TERMS

In our clustering step, we use *Laplacian Eigenmaps* described in [NJW02] to get closely related terms into one cluster. Moreover, for each topic, the document distribution is assumed to be Gaussian and the Gaussian kernel is used to make this points linear. Our clustering algorithm is performed as follows:

1. Calculate the matrix A as

$$A_{ij} = \begin{cases} \exp(-dis_{i,j}/\sigma^2), & \text{if } r \neq j \\ 0, & \text{otherwise} \end{cases}$$

2. Calculate the matrix D as $D_{ij} = \sum_j(A_{ij})$, where D is a diagonal matrix.
3. L is defined as $L = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$.
4. Find $x_1, x_2, \dots, x_k$, the k largest eigenvectors of L (chosen to be orthogonal to each other in the case of repeated eigenvalue), and form the matrix $X = [x_1 x_2 \dots x_k] \in \mathbb{R}^{n \times k}$ by stacking the eigenvectors in columns.

5. Form the matrix Y from X by renormalizing each of X 's rows to have unit length (i.e.

$$Y_{ij} = \frac{x_{ij}}{(\sum_j x_{ij}^2)^{1/2}}.$$

6. Treating each row of Y as a point in $\mathbb{R}^K$, cluster them into k clusters via an extended k -means.
7. Finally, assign the original point s_j to cluster j if and only if row i of the matrix Y was assigned to cluster j .

In this procedure, σ^2 is a parameter for how rapidly the affinity $\mathcal{A}_{ij}$ falls off with the distance between s_i and s_j . In fact, however, a fixed parameter may lead to problems when topics' documents number and variance varies, because if we have topics $\mathcal{A}$ and $\mathcal{B}$, where topic $\mathcal{A}$ has a larger variance and more documents than $\mathcal{B}$, then we get a document d which talks about $\mathcal{A}$ and $\mathcal{B}$ at the same time and assume that topics $\mathcal{A}$ and $\mathcal{B}$ in d have the same importance. If creating a new cluster is not allowed, we may prefer clustering d into $\mathcal{A}$ because it has a larger variance and it seems containing more property. So we introduce a auto-adaptive parameter selection.

To make the parameter σ adaptive, we use the following method to select a σ parameter for point i and j , which is similar to the work in [ZPO5].

The affinity between points s_i and s_j is defined as

$$\mathcal{A}_{ij} = \exp\left(-\frac{|s_i - s_j|^2}{\sigma_i \sigma_j}\right),$$

where the σ_i and σ_j are in a local scale $\sigma_i = \sqrt{\frac{1}{n} \sum_n |s_i - s_j|^2}$. This method allows σ to be automatically adjusted according to local variance, that is, no parameter setting is re-

quired. Our preliminary experimental results show effectiveness of this parameter selection method.

3.4.3 EXTENDED k -MEANS ALGORITHM

For twitter data, it is usually hard to set a reasonable cluster number, and at most time, adjusting this parameter is a hard work for us. In actual experiment, we usually tend to select a lesser cluster number to avoid the classes which only contain one item. So we would like to adapt a simple clustering method which is able to adjust the number of clusters automatically.

As mentioned above, we use an extended k -means which allows to make new clusters in our clustering process by adjusting inner variances of clusters under a threshold so that we can obtain a good result with smaller distortion. The algorithm is summarized as follows:

1. Given k as an initial cluster number, k centers are randomly selected.
2. We calculate the squared errors with

$$J = \sum_{C_k \in \mathcal{CLS}} \sum_{\forall x_i \in C_k} (|c_k - x_i|^2),$$

where $\mathcal{CLS}$ is the family of clusters, c_k is the center of cluster C_k .

3. For each point, a cluster with the smallest distance to the point is identified.
4. The variance of the cluster with this point added is calculated. If the variance is larger than a threshold, a (singleton) cluster with the point is newly created. If otherwise, the point is merged into to the cluster.

5. When all of the points are processed, all empty clusters are deleted.
6. For each cluster C_i , its center vector (point) c_i is re-calculated by

$$c_i = \frac{1}{n} \sum_{x \in C_i} x$$
, where n is the number of points in C_i .
7. If the change of squared error is smaller than a given threshold, the clusters are output and the process is terminated. If otherwise, return to 2.

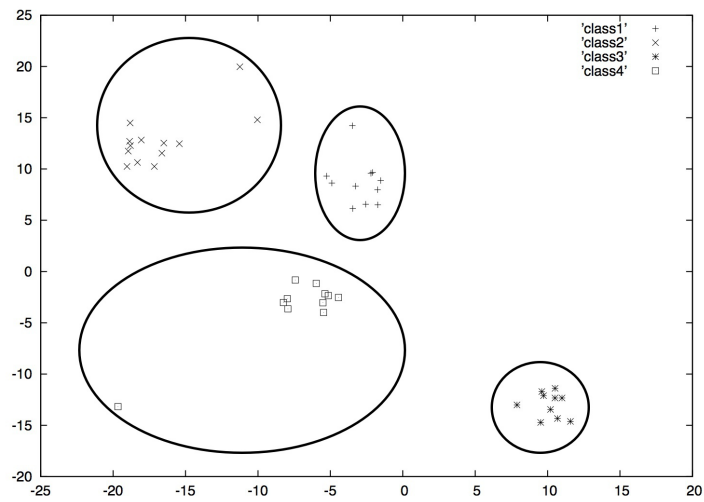
In order to verify the ability of extended k -means, we have tested it on a dataset with several centers and noise. Figure 3.2 shows our preliminary experimental results. It is clear that a traditional k -means cannot handle this situation as is shown in Fig. 3.2(a). On the other hand, our algorithm results in a good separation (Fig. 3.2(b)). In the method of k -means++ [AV07], although a set of good initial k seeds can be identified to get a better performance, we cannot change the initial k value which is inadequately given. However, in our method, it is no matter what the preset k is. Especially, if we provide a k smaller than the actual number of clusters, we can expect a relatively good result, which can allow us to select the cluster number more easily.

3.5 METHOD OF ASSOCIATIVE SEARCH BY MEDIATORS

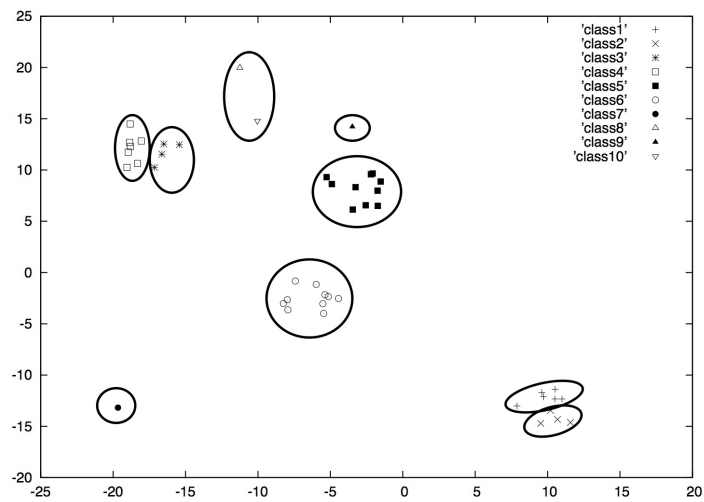
In this section, we formalize our method of associative search.

We are given a pair of relations, a person-term relation and a document-term relation, each of which is represented as a formal context, $\mathcal{P} = (P, V, R_P)$ and $\mathcal{D} = (D, V, R_D)$, respectively. As has been discussed in the previous section, it is assumed that the set of persons P is sorted (ranked) in descending order of their mediator levels.

Our associative search is performed by the following five steps.



(a) Traditional k -Means



(b) Extended k -Means

Figure 3.2: Clustering Results by Traditional k -Means and Extended k -Means

1) Identifying Query Concept:

Given a set of keywords $K \subseteq V$ as a query, a query concept $Q_{\mathcal{P}} = (P_Q = K^{(\mathcal{P})}, T_Q = K'^{(\mathcal{P})})$ in $\mathcal{P}$ is computed.

2) Getting User Interest:

For each person $p \in P_Q$ with Top- N higher mediator level, the user is asked to assign a mark “+”, “-” or “*” according to his/her interest.

3) Extracting Maximal Concepts Consistent with User Interest:

We try to extract maximal sub-concepts of $Q_{\mathcal{P}}$, $C_B = (P_B, T_B)$, which is consistent with the user interest.

4) Identifying User Aspects as Concepts:

Based on C_B in $\mathcal{P}$, a user aspect is identified as a concept $C_A = (A^{(\mathcal{D})}, A)$ in $\mathcal{D}$, where $A = T_B \setminus T_Q$.

5) Extracting δ -Similar Concepts w.r.t. User Aspect:

Regarding C_A as a bridge concept, we try to extract concepts in $\mathcal{D}$, C_{target} , each of which is δ -similar to the query concept $Q_{\mathcal{D}} = (K^{(\mathcal{D})}, K'^{(\mathcal{D})})$ w.r.t. A .

3.5.1 IDENTIFYING QUERY CONCEPTS

Let us assume that a query Q a user interested in is given as a set of keywords (terms) $K \subseteq V$. We can consider a *query concept* in each context. They are defined as $C_Q^{\mathcal{P}} = (K^{(\mathcal{P})}, K'^{(\mathcal{P})})$ in $\mathcal{P}$ and $C_Q^{\mathcal{D}} = (K^{(\mathcal{D})}, K'^{(\mathcal{D})})$ in $\mathcal{D}$, respectively. $K^{(\mathcal{P})}$ is a set of persons related to the query and $K'^{(\mathcal{D})}$ a set of documents matching the query.

3.5.2 GETTING USER INTEREST

We try to associatively search concepts of documents which are similar to $C_Q^{\mathcal{D}}$ with respect to some user's interest in persons. In order to get such a user's interest, we interactively ask the user to express his/her preference on the persons related to the query, that is, the persons in $K^{(\mathcal{P})}$. Since the number of persons in $K^{(\mathcal{P})}$ is large, the burden of such a user interaction would be a bit heavy for the user. Therefore, the user is asked his/her preference only on the Top- N ranked persons with higher mediator levels.

More precisely speaking, let $\mathcal{M} \subseteq P$ be the list of Top- N ranked mediators in P , that is, $\mathcal{M} = \{p \in P \mid \text{rank}(p) \text{ is in the top-}N\}$. For the query concept in $\mathcal{P}$, $Q_{\mathcal{P}} = (P_Q, T_Q)$, according to the user interest, for each person $p \in (P_Q \cap \mathcal{M})$, the user assigns a sign $+$, $-$ or $*$ to p for “favorite”, “dislike”, and “don't care”, respectively, where the sign given to p is referred to as $\text{sign}(p)$.

Assuming $*$ as default sign to each $p \in P_Q \setminus \mathcal{M}$, we can divide P_Q into three groups, POS , NEG and DC defined as $POS = \{p \in P_Q \mid \text{sign}(p) = +\}$, $NEG = \{p \in P_Q \mid \text{sign}(p) = -\}$ and $DC = \{p \in P_Q \mid \text{sign}(p) = *\}$.

3.5.3 FINDING CONCEPTS CONSISTENT WITH USER INTEREST

Based on the user interest, we try to find a concept in $\mathcal{P}$ which is *consistent* with the user interest. The consistency of concept is formally defined as follows.

Definition 5 (Consistent Concept)

Let $Q_{\mathcal{P}} = (P_Q, T_Q)$ be a query concept in $\mathcal{P}$, where P_Q is divided into POS , NEG and DC based on the user interest. Then, a concept $B = (P_B, T_B)$ in $\mathcal{P}$ is said to be *consistent*

with the user interest if and only if $P_B \subseteq P_Q$, $P_B \supseteq POS$ and $P_B \cap NEG = \emptyset$. ■

A consistent concept B is a sub-concept of the query concept whose extent must subsume POS and whose intent must completely exclude NEG . Since there are in general several concepts satisfying the constraints, we try to extract maximally general ones among them.

3.5.4 IDENTIFYING USER ASPECT AS CONCEPT

Roughly speaking, an associative search of concepts can be realized by finding a concept C which is similar to a query concept Q with respect to some aspect user interested in. If we can observe a concept which corresponds to the user aspect and bridges Q and C in some sense, it would be reasonable to accept similarity between Q and C under the aspect. In order to realize this kind of associative search, we here formalize a user aspect as a concept reflecting the user interest.

Let $B = (P_B, T_B)$ be a (maximal) concept in $\mathcal{P}$ which is consistent with the user interest. From the definition, since B is a sub-concept of $Q_{\mathcal{P}} = (P_Q, T_Q)$, $T_Q \subseteq T_B$ holds. Therefore, $\mathcal{A} = T_B \setminus T_Q$ can be viewed as the set of attributes (terms) which can implicitly characterize the user interest. In other words, $\mathcal{A}$ can be considered to represent a preferable aspect of the user.

In order to find similarity of concepts in $\mathcal{D}$ under the aspect reflecting the user interest in persons, we consider a concept in $\mathcal{D}$ which is defined based on $\mathcal{A}$. Formally speaking, if $\mathcal{A}$ is a closure in $\mathcal{D}$, that is, $(\mathcal{A}'^{(\mathcal{D})}, \mathcal{A}''^{(\mathcal{D})} = \mathcal{A})$ is a concept in $\mathcal{D}$, then we regard the concept $C_{\mathcal{A}} = (\mathcal{A}'^{(\mathcal{D})}, \mathcal{A})$ as an aspect reflecting the user interest in persons.

3.5.5 EXTRACTING CONDITIONALLY SIMILARITY CONCEPTS W.R.T. USER ASPECT

Intuitively speaking, our associative search is realized by finding a concept C which is similar to a query concept Q with respect to some user aspect. Before presenting such a conditional similarity between concepts, we define a similarity between concepts without any conditioning. Our similarity can be defined based on the notion of *bond* [Omio3] which is regarded as an extension of *Jaccard Coefficient* and can easily be calculated in our concept search.

Definition 6 (Concept Similarity Based on Bond Measure)

Let $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be a pair of concepts. Then a similarity between C_1 and C_2 , denoted by $\text{sim}(C_1, C_2)$, is defined as

$$\text{sim}(C_1, C_2) = \text{bond}(Y_1 \cup Y_2) = |X_1 \cap X_2| / |X_1 \cup X_2|. \quad \blacksquare$$

The measure is extended for *conditional similarity* between concepts.

Definition 7 (Concept Similarity Based on Conditional Bond)

Let $C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$ be a pair of concepts. For a set of attributes R as some aspect, a similarity between C_1 and C_2 with respect to the aspect R , denoted by $\text{sim}(C_1, C_2|R)$, is defined as

$$\text{sim}(C_1, C_2|R) = \text{bond}(Y_1 \cup Y_2|R) = \left| \bigcap_{y \in Y_1 \cup Y_2} (R' \cap y') \right| / \left| \bigcup_{y \in Y_1 \cup Y_2} (R' \cap y') \right|. \quad \blacksquare$$

Let δ be a given threshold for the minimum bond value. For a pair of concepts C_1 and C_2 , if $\text{sim}(C_1, C_2) \geq \delta$ holds, then C_1 and C_2 are said to be δ -similar. Moreover, $\text{sim}(C_1, C_2|R) \geq$

δ for an aspect R , C_1 and C_2 are said to be *conditionally δ -similar* with respect to R .

For a query concept Q and the user aspect C_A , if C_A bridges Q and a concept C in some sense, then it would be natural to find similarity between Q and C under the aspect. Here, C_A is an essential concept which corresponds to the user aspect and can work as the basis of the similarity between Q and C . We call such a C_A a *bridge concept* for Q and C .

Definition 8 (δ -Bridge Concept)

Let $C_R = (X_R, Y_R)$ and $C_L = (X_L, Y_L)$ be a pair of concepts. For a given similarity threshold δ , if a concept $C_W = (X_W, Y_W)$ satisfies the following conditions, then C_W is called a *δ -bridge concept* between C_R and C_L w.r.t. the *aspect* Y_W .

Structural Constraint: $X_R \cap X_W \neq \emptyset$ and $X_L \cap X_W \neq \emptyset$.

Conditional Similarity Constraint: C_R and C_L are conditionally δ -similar with respect to Y_W , that is, $\text{sim}(C_R, C_L | Y_W) \geq \delta$. ■

The structural constraint requires that the concept C_W properly bridges (overlaps) with both C_R and C_L at their extents. Moreover, by the conditional similarity constraint, C_R and C_L must be δ -similar under the conditioning with the intent of C_W , Y_W . It should be emphasized here that C_R and C_L are not always δ -similar in the original context (without conditioning).

Regarding the user aspect $C_A = (\mathcal{A}^{(\mathcal{D})}, \mathcal{A})$ as a bridge concept, for the query concept $Q_{\mathcal{D}} = (K^{(\mathcal{D})}, K''^{(\mathcal{D})})$ in $\mathcal{D}$, we try to find a target concept C_{target} which is conditionally δ -similar to $Q_{\mathcal{D}}$ with respect to $\mathcal{A}$.

3.6 EXTRACTING TARGET CONCEPTS FOR ASSOCIATIVE SEARCH

As has been discussed in the previous section, our computation procedure of associative search consists of five steps. In the procedure, the primary tasks are “finding maximal consistent concepts” and “finding conditionally δ -similar concepts”. For each of the tasks, we can design a simple depth-first algorithm.

3.6.1 EXTRACTING CONCEPTS CONSISTENT WITH USER INTEREST

Our task here is to find every maximal concept C_B in $\mathcal{P}$ which is a sub-concept of the query concept $Q_{\mathcal{P}} = (P_Q, T_Q)$ and whose extent must subsume POS and exclude NEG .

Assume P_Q can be divided into POS , NEG and DC . In order to find C_B satisfying the constraints, we can basically expand POS by adding a person in DC step by step in a depth-first manner. More precisely, for the closure of person set X_i such that $(POS \cup DC) \supseteq X_i \supseteq POS$, we expand X_i by adding a person $x \in (DC \setminus X_i)$ and compute the closure $X_{i+1} = (X \cup \{x\})''^{(P)}$. Then we check whether $X_{i+1} \subseteq (POS \cup DC)$ or not. If yes, X_{i+1} is tried to further expand by adding a person in $DC \setminus X_{i+1}$. If otherwise, X_{i+1} is discarded and X_i is expanded with another person by backtrack because X_{i+1} includes some person in NEG or $(X_{i+1}^{(P)}, X_{i+1})$ is not a sub-concept of $Q_{\mathcal{P}}$. If a closure X_i cannot be expanded, then X_i is stored into a list of maximal concepts as a tentative candidate. Such an expansion process recursively iterated until no closure remains to be expanded. Details of the algorithm can be found in [LHOo8].

3.6.2 EXTRACTING CONDITIONALLY SIMILAR CONCEPTS W.R.T. USER ASPECT

The task here is to find concepts each of which is conditionally δ -similar to the query concept and can be connected with the query concept by a concept corresponding to the user aspect.

Let $Q_D = (K^{(\mathcal{D})}, K'^{(\mathcal{D})})$ be a query concept and $C_A = (A^{(\mathcal{D})}, A)$ a user aspect (as a concept). A target we try to find is a concept $C_{Target} = (T'^{(\mathcal{D})}, T)$ such that (1) $T \cap A = \emptyset$, (2) $K'^{(\mathcal{D})} \cap T = \emptyset$, (3) $K^{(\mathcal{D})} \cap T'^{(\mathcal{D})} \neq \emptyset$ and (4) $bond(K'^{(\mathcal{D})} \cup T | A) \geq \delta$. Particularly, we try to extract maximal ones satisfying the constraints. It should be noted here that the constraints (1) and (2) are not included in our original definition of δ -bridge concepts. They are assumed to obtain more interesting associations of terms. Moreover, from a computational view point, they can restrict our search space for efficient computation.

In order to find such a concept, we try to recursively expand a closure of terms in depth-first manner. Let $X_i \subseteq V$ be the closure of a set of terms such that $X_i \cap (K'^{(\mathcal{D})} \cup A) = \emptyset$. For a term $x \in V \setminus (K'^{(\mathcal{D})} \cup A \cup X_i)$, we check whether $X_{i+1} = (X_i \cup \{x\})'^{(\mathcal{D})}$ as T satisfies all of the four constraints. If X_{i+1} does not satisfy (1) or (2), X_{i+1} can be discarded because any expansion of X_{i+1} can never satisfy the constraints. If X_{i+1} does not satisfy (3), that is, $K^{(\mathcal{D})} \cap X_{i+1}^{(\mathcal{D})} = \emptyset$, then any expansion of X_{i+1} also violates the constraint. Therefore we can stop expanding X_{i+1} . If the constraint (4) cannot be satisfied for X_{i+1} , any expansion of X_{i+1} can be pruned. This is because the bond measure is monotonically decreasing as a set of attributes (terms) becomes larger. Therefore, (4) cannot be satisfied for any expansion of X_{i+1} . If X_{i+1} is discarded, X_i is tried to expand with another term in $V \setminus (K'^{(\mathcal{D})} \cup A \cup X_i)$ by backtrack.

On the other hand, all of the constraints are satisfied for X_{i+1} , then $(X_{i+1}^{(\mathcal{D})}, X_{i+1})$ becomes

a candidate of our target. Then X_{i+1} is further tried to expand with a term in $V \setminus (K'^{(D)} \cup \mathcal{A} \cup X_{i+1})$. Such an expansion process is recursively iterated until no closure remains to be examined.

3.7 EXPERIMENTAL RESULTS

In this section, we present our experimental results. Our system coded in C has been executed on a PC with Intel Core-i3 M380 (2.53GHz) CPU .

3.7.1 PERSON-TOPIC RELATION AND TWEET-TOPIC RELATION

We have created our datasets from a set of tweets gathered with *Twitter* search API *. We have collected Japanese tweets including a keyword “*Soccer*” † in November, 2012. The number of collected tweets is about 600, 000.

As a preprocess, we have first applied *Morphological Analysis*, and then extracted nouns as feature terms. Then, too frequent and too infrequent nouns have been removed. Since each tweet consists of at most 140 characters, a number of tweets have become empty after the preprocess. We have exclude those empty ones and obtained 538, 355 tweets. The number of terms in the remaining tweets is 51, 927 and the number of persons by whom those are tweeted 238.

In our mediator ranking process, the tweets are first concatenated into a single document for each people. Then, all terms with lower *TF-IDF* values have been further removed. With our extended k -means, the remaining 912 terms have been clustered into 155 topics.

*<https://dev.twitter.com>

†In practice, each keyword is in Japanese.

The persons have been assigned mediator levels based on the distances to the topics and ranked according to the levels.

It is noted here that after the ranking, the original tweets have been compressed into ones represented in corresponding topics. This is because each of the original tweets consists of a small number of nouns and it is hence difficult to extract concepts without the compression. For example, if a tweet consists of a set of (original) terms *a*, *b*, *c* belonging to topics *A*, *B* and *C*, respectively, then it is compressed into one with *A*, *B* and *C*. By the compression, our chance to obtain concepts can be enhanced because several distinct terms are identified as the same topic. Thus, sharing 155 topics, we have prepared a person-topic relation with 238 persons and a tweet-topic relation with 538, 355 tweets.

3.7.2 EXAMPLE OF TARGET CONCEPT BY ASSOCIATIVE SEARCH

We show here an example we have actually extracted from the above relations.

Given a keyword “smartphone” as a query, we first convert it into its corresponding topic, *topic-73*. The extent of the query concept in the person-topic relation consists of 200 persons. For a user interest setting the highest ranked person as NEG, we have obtained 21 maximal consistent concepts. Then, those concepts provides 21 candidate aspects. Some of them correspond to concepts in the tweet-topic relation and can work as δ -bridge concepts each of which connects the query concept and a target concept. An example of associative connection of concepts is presented in Figure 3.3.

In the figure, *topic-55* of the bridge concept includes “mobile” and *topic-62* of the target concept “digital book”. This means that we can associatively access to “digital book” from “smartphone” via “mobile”. More concretely speaking, the following tweet which is

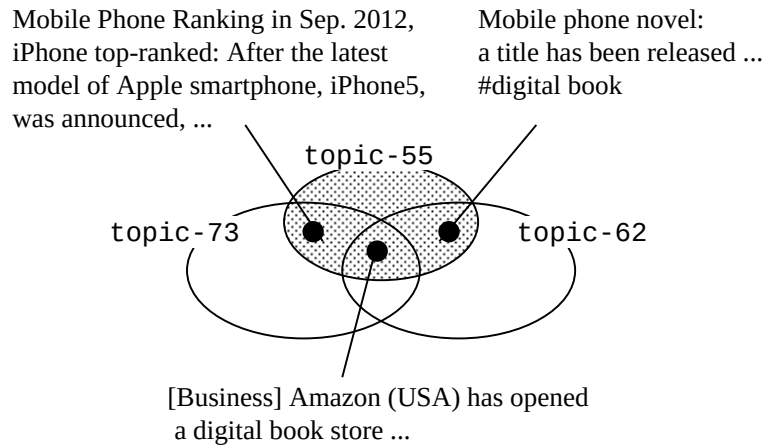


Figure 3.3: Example of Associative Connection of Concepts for “Smartphone”

an instance of the bridge concept can connect “smartphone” and “digital book”:

Amazon (USA) has opened a digital book store “Kindle Store” in Japan. Although Kindle Paperwhite will be shipped in November, users can access to the store with their smartphones (multi-function mobile phones).

Needless to say, such a flexible access cannot be obtained by standard search engines. Thus, our associative search provides us a chance to get various useful information which might be interesting for us.

The total computation time has been 5.0 seconds, including 0.24 sec. for extracting 21 maximal consistent concepts. Thus, our associative search can be performed efficiently.

3.8 CONCLUSIONS

In this research, we have discussed a framework of associative search. A remarkable point of the framework is that our search is dependent on user interests in good mediators. We have

presented a computational procedure for our associative search and designed depth-first algorithms for extracting our target concepts.

© 2017 IEEE. Reprinted, with permission, from Hongjie
Zhai, Makoto Haraguchi, *A Linear Algebraic Inference
for Feature Association*, 12th International Conference
on Knowledge, Information and Creativity Support
System, Nov 9-11, 2017.

4

Feature Association Discovery by Linear Algebraic Inference

4.1 INTRODUCTION

In this chapter, we proposed a discovery system which focus on finding feature associations based on user's interest. This system is based on a *linear algebraic inference* method. It is

designed for finding association among features over two distinct domains represented by data matrices with different dimensions. As the standard definition of similarities cannot be directly applied to our problem situation, we introduce in this chapter a new technique for building “common space”, biased by a hint for association, in which similarity/closeness in that space can be used to infer paired features to be associated. In case that the domains are described by terms, the hint could be “A is like X” or “A is associated with X”. Then, by analogy, it would be reasonable to infer the association of another “B” and “Y”, provided similar relationships between “A” and “B” and one for “X” and “Y” hold simultaneously. In a pioneering study about analogy [Gre85], the associated pair of “A” and “X” is called an analogical hint for triggering off reasoning by analogy. In this chapter, the domains for such inference and the terms for describing them are supposed to be data matrices and their features (column vectors), respectively. This is simply because the latter data representation is often used in many applications. In fact, we show in Section 4.7 some experimental results for two bag-of-words data sets built from documents written in different two languages.

From the above assumption about data representation, the hint treated in this chapter must be given as a small set of paired features which users agree with associating them. One of important points remarked here is that the number of paired features given in the hint is small, compared with the number of features. Although it would be possible to formulate our problem as “association learning” with large hint as a training data, it is often the case that we can infer unknown association even from a relatively smaller number of known associated pairs.

Given two data matrices and a hint of known or approved association, an inference schema is as follows, borrowing a general idea from the studies of analogy [Gen83; Hes66]:

(Hint) First we have an assumption that a pair of feature $f_{h_j}^f$ in one domain and $(f_{h_j}^g)$ in another shows an approved association. Such paired features form a hint $H = \{(f_{h_j}^f, f_{h_j}^g)\}_{1 \leq j \leq n_H}$.

(Inference) Infer another pair (f^f, f^g) , provided the relationships between $f_{h_j}^f$ and f^f are similar for $i = 1, 2$

The problems we have to solve in order to perform the schema are:

(DimP: Dimension Problem) The dimensions of features as column vectors are not the same in general.

(RelC: Relationship Criterion) How do we define the relationship between $f_{h_j}^f$ and f^f ? Under what criterion do we decide that two relationships are similar?

To solve (DimP) and to give a plausible answer to (RelC), we use Non-Negative Tri-Factorization for data matrix [DP15]. First of all, (DimP) can be solved by applying the standard Non-Negative Matrix Factorization (NMF) [PT94; LSo1]. Suppose, for $i = 1, 2$, we have a data matrix $X^i = (f_1^i \cdots f_{N_i}^i)$ of r^i -dimensional features (column vectors) f_j^i . To equalize r^1 and r^2 , we apply NMF to have the decomposition $X^i \approx W^i L^i$, where L^i is a $k_1 \times N_i$ matrix with the same reduced dimension k_1 for row. The decomposition is based on some correlation among row vectors. By Tri-factorization, correlations among column vectors after NMF for rows are furthermore considered to have a $k_1 \times k_2$ C^i with $X^i \approx U^i C^i R^i$, where C^i consists of k_2 column vectors with the dimension k_1 . As a result, $X^i \approx U^i C^i R^i$ with the weight matrices U^i and R^i defines a linear embedding, ϕ^i in Fig. 4.1, which transforms the original features in X^i into the subspace spanned by column vectors of C^i . Such a tri-factorization is simultaneously performed for both of X^1 and X^2 in this

chapter. Our common space is now defined to be spanned by column vectors in C^1 or in C^2 . By the definition, we can use the mapping ϕ^1 and ϕ^2 into this common space for features in X^1 and X^2 , respectively. Thus, using the common space and the embedding ϕ^i , we can compare features over two domains and evaluate how they are close (similar) or far away.

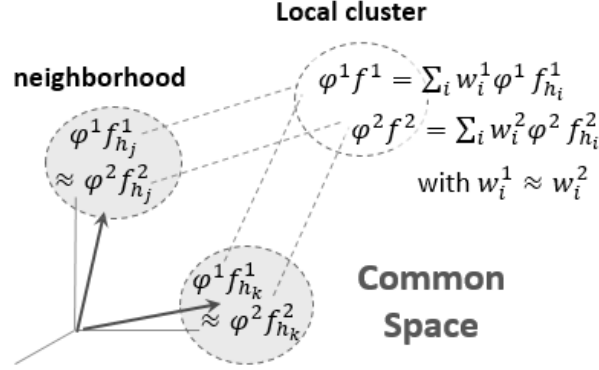


Figure 4.1: Typical case of inferring new association (f, f') in the common space under the constraint for hint features.

Now, let us return to the question (RelC). There exist many possible tri-factorizations and their common spaces. We choose one of them so that we can answer to (RelC) by restricting the relationship among f and hint features $f_1^I, \dots, f_{n_H}^I$ to one expressed by linear combination.

(ARelC: Answer to RelC) Suppose $f = \sum_{j=1}^{n_H} w_j^1 f_j^1$. For weight vector $(w_1^2, \dots, w_{n_H}^2)$ similar to $(w_1^1, \dots, w_{n_H}^1)$, suppose furthermore that we have similar representation $f' = \sum_{j=1}^{n_H} w_j^2 f_j^2$. Then we infer that f and f' can be associated, because they are similarly composed by hint features.

As is depicted in Fig. 4.1, the features f and f' satisfying the condition of (ARelC) also have the same weight vectors for hint features even after embedding. This is simply because

embeddings ϕ^i are linear. Based on this simple fact, we here introduce another important constraint to make a common space biased by hint.

(CHCS: Closeness of hint features in common space): For every pair (f_j^1, f_j^2) in the hint, f_j^1 and f_j^2 must be close after embedding by ϕ^i

It is straightforward to verify that f^1 and f^2 satisfying the condition of (ARelC) become close in the common space satisfying (CHCS). That is, the solution paired features will appear in a local cluster of this constrained common space biased by hint. Besides such a newly inferred pair (f^1, f^2) that is strongly supported by the linear algebraic explanation with respect to hint features, close vectors in the constrained common space have similarity biased by the hint. For this reason, we also regard (g^1, g^2) as potentially associated pair whenever ϕg^1 and ϕg^2 are close in the constrained space.

The image of the whole system is shown in Figure 4.2. Given two databases and user's hint, a common space will be constructed based on the user's hints firstly. In the mean time, databases will also be embedded into this common space. After this, we are able to do the inference in the common space.

The algorithm presented in Section 4.4 is designed to search for one of optimum solutions for the constrained common space biased by hint. It is realized as an extended Tri-Factorization algorithm whose convergence of search is verified. A Laplacian-based constraint is also introduced to have more plausible common space. More precisely, it helps to prevent the optimization process from forming common space in which features far from hint and those near to hint become close. This is a kind of heuristics which is however effective in our experiments.

This chapter is organized as follows: Section 4.2 introduces the basic notation. In sec-

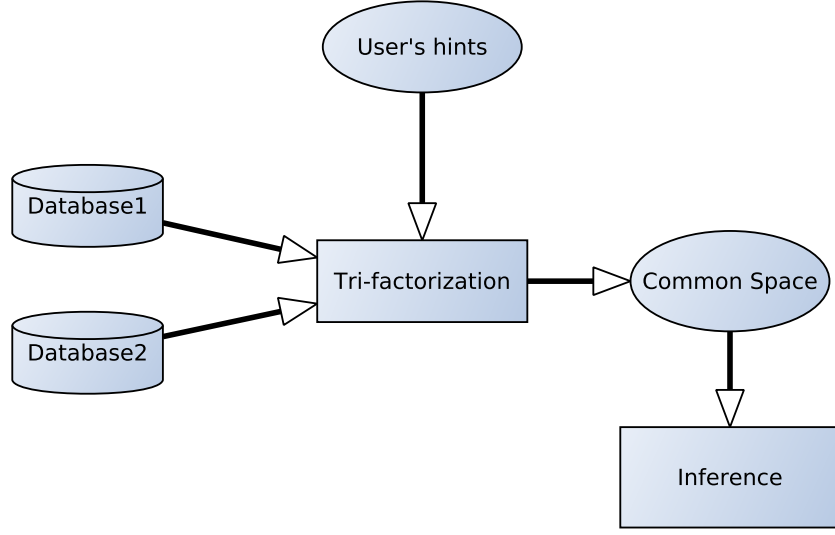


Figure 4.2: The Image of The Whole System

tion 4.3, we briefly review the Non-negative Matrix and Non-negative Tri-Factorization. Section 4.4 describes the details of the formalization of the common space embedding method. Section 4.5 presents the update rule and algorithm for the formalization. In section 4.7, we show the experiment result on the Reuters news articles in Italian and Spanish. By using nouns as features and translations as hints, the proposed algorithm achieves finding many meaningful word pairs. The conclusion and future works was discussed in Section 4.8.

4.2 PRELIMINARIES

Let us consider two datasets X^1 and X^2 . Each dataset contains a set of feature vectors, written as $X^1 = \{f_1^1, f_2^1, \dots, f_u^1\}$ and $X^2 = \{f_1^2, f_2^2, \dots, f_l^2\}$. In real world applications, because X^1 and X^2 are usually come from different datasets, their sizes are usually different, i.e.

$|X^1| \neq |X^2|$. Additionally, the known associations are represents as pairs like (f_i, f_j) . In this chapter, we regard these known associations as “Hints”. The set of hints is written as $H = \{(f_i, f_j) | f_i \text{ and } f_j \text{ are associated}\}$.

4.3 NON-NEGATIVE MATRIX FACTORIZATION AND TRI-FACTORIZATION

Before introducing our algorithm, we would like to firstly review the Non-negative Matrix Factorization and the Non-negative Tri-factorization. Non-negative Matrix Factorization (NMF) [LS99] focuses on analyzing non-negative data by decomposing the data matrix into two non-negative matrices.

Given a data matrix $X = [f_1, \dots, f_N] \in \mathbb{R}^{M \times N}$, where each column of X represents a data vector. NMF tries to find two non-negative matrices $N \in \mathbb{R}^{M \times K}$ and $M \in \mathbb{R}^{K \times N}$ whose product can approximate the data matrix:

$$X \approx NM \tag{4.1}$$

To measure the quality of the approximation, it is necessary to define the cost function. Cost function can be usually constructed by measure the ”distance” between X and NM . In the research of NMF, there are two major definitions of the cost function: Euclidean distance and divergence [LS01].

For Euclidean distance, it is simply defined like:

$$L(N, M) = \sum_{ij} (N_{ij} - M_{ij})^2 \tag{4.2}$$

In fact, this type of distance between matrix is also called Frobenius norm [PT94]. To

make the equations simple, the Frobenius norm is usually write like $\|N - M\|_F^2$. Thus, the Euclidean distance cost function of NMF can be written like:

$$L(N, M) = \|X - NM\|_F^2 \quad (4.3)$$

Another definition called divergence can be written in the following forms [LS01]:

$$D(N||M) = \sum_{ij} (N_{ij} \log \frac{N_{ij}}{M_{ij}} - N_{ij} + M_{ij}) \quad (4.4)$$

Apparently, this definition can be not called a "distance" because usually $D(N||M) \neq D(M||N)$. However, because $D(N||M) \geq 0$ and $D(N||N) = 0$, it is still a good measure for the cost function. Furthermore, if we can guarantee that $\sum_{ij} N_{ij} = \sum_{ij} M_{ij}$, the above definition can be reduce to the following equation:

$$D(N||M) = \sum_{ij} (N_{ij} \log \frac{N_{ij}}{M_{ij}}) \quad (4.5)$$

which is the Kullback–Leibler divergence or Relative entropy.

To get the best approximation, the optimization target of NMF is minimize $L(N, M)$ or $D(N||M)$. We should point out that for some theoretic research, or for some extra constraints, Euclidean distance and divergence may have different property. However, in the real-world application, there is no really differences between this two different cost functions. In this work, we choose the Frobenius norm.

From the viewpoint of clustering, each column of M is a cluster indicator of correspondent data vector, while the rows of N are vectors of cluster centers. In another aspect, we can also say that the columns M is the vector representations in $\mathbb{R}^K$ of the data vectors in

X . NMF has been known to be equal to the spectral clustering [DHS05]. Thus, it is safe to apply NMF for the cases where spectral k -means can be used.

It is easy to see that the cost function Equation 4.3 only convex in N or M . Therefore, we usually expect to find a local minimum of $L(N, M)$. Paper [LS01] presented a fast multiplicative iterative update rules to solve this problem. For Euclidean distance cost function $\|X - NM\|_F^2$, the update rules can be written as:

$$N_{ij} = N_{ij} \frac{(XM^T)_{ij}}{(NMM^T)_{ij}} \quad (4.6)$$

$$M_{ij} = M_{ij} \frac{(N^T X)_{ij}}{(N^T NM)_{ij}} \quad (4.7)$$

Other than the multiplicative update, there are also several gradient optimization methods for NMF [Chu+04; Lino7]. Here, we only introduce one of a simple gradient methods proposed in [Chu+04]:

$$N = \max(0, N - \alpha \nabla_N L(N, M)) \quad (4.8)$$

$$M = \max(0, M - \alpha \nabla_M L(N, M)) \quad (4.9)$$

Here, α is the update rate (i.e. step size) which defined by user. Because all the matrices are non-negative in the NMF, the update rule makes all the negative term to 0 in each round.

Non-negative Tri-factorization [LZY05]* is an extension of the NMF. Instead of decom-

*In original paper, it is also called Non-negative Block Value Decomposition (NBVD).

posing data matrix into two matrices, tri-factorization decomposes the data matrix X into three non-negative matrices $U \in \mathbb{R}^{M \times K_1}$, $C \in \mathbb{R}^{K_1 \times K_2}$ and $R \in \mathbb{R}^{K_2 \times N}$:

$$X \approx UCR \quad (4.10)$$

Similar to NMF, the cost function can also be written in Frobenius norm [LZY05]:

$$L(L, C, R) = \|X - UCR\|_F^2 \quad (4.11)$$

In the tri-factorization, rows of U represent the cluster indicators of the row vectors in X . In the meanwhile, the cluster indicators of column vectors in X are represented by the columns of R . C is the matrix that describe the relation between row clusters and column clusters. Thus, tri-factorization is actually performing the co-clustering on the input data. Similar with NMF, U and R can be treated as the vectors in the space R^{K_1} and R^{K_2} .

When we deriving the update rules, by letting $A = UC$ for R and $B = CR$ for U , we can solve the problem totally the same as the NMF algorithm. Even though the term C is special, it is still possible to derive the update rules by following the method of [LS01]. [LZY05] proposed the update rules as follows:

$$U_{ij} \leftarrow U_{ij} \frac{(XR^T C^T)_{ij}}{(UCRR^T C^T)_{ij}} \quad (4.12)$$

$$C_{ij} \leftarrow C_{ij} \frac{(U^T X R^T)_{ij}}{(U^T U C R R^T)_{ij}} \quad (4.13)$$

$$R_{ij} \leftarrow R_{ij} \frac{(C^T U^T X)_{ij}}{(C^T U^T U C R)_{ij}} \quad (4.14)$$

Many researches [LZY05; Din+06; DP15] showed that the Tri-factorization generally outperforms NMF, especially for the co-clustering tasks [DP15].

Based on the original proposal of NMF and Non-negative Tri-factorization, there also exists many variants. One of the most famous variants may be the graph regularized NMF [Cai+11]. This variant introduces an extra constraint, which tries to keep the topology information in the original data. Technically speaking, graph regularized NMF uses the graph laplacian L as the topology. The data points which are near to each other in the original data will also be embedded as neighborhood in the new space. Other than this, [LBF11] proposed an online version of NMF based on a special divergence. Similar to recent research on Neural Network, there is also deep NMF [LHW15]. Deep NMF tries to decompose factorized matrix more than one times to achieve an effect similar to Deep Learning.

4.4 COMMON SPACE EMBEDDING USING TRI-FACTORIZATION

We have described the basic idea of our method in Section 4.1. This section will focus on introducing the details of our algorithm. As we have discussed in previous sections, to discover new feature associations from hints, we use non-negative tri-factorization to embed all the features $f_i^1 \in X^1, f_j^2 \in X^2$ into a common space, where the correspondence vectors are represented as $f_i^{1'}$ and $f_j^{2'}$. For this common space, we have the following two constraints:

1. (CHCS) If (f_i^1, f_j^2) is a hint, $f_i^{1'} \approx f_j^{2'}$ in the common space.
2. (ARelC) If $|f_k^1 - f_i^1|_F^2 > |f_l^1 - f_i^1|_F^2$ in the original data, $|f_k^{1'} - f_i^{1'}|_F^2 > |f_l^{1'} - f_i^{1'}|_F^2$ in the

common space ($r = 1, 2$).

Firstly, because we are using non-negative tri-factorization to get the embedded vectors, f^r basically comes from the following tri-factorization:

$$\mathcal{O}_1 = \|X^1 - U^1 C^1 R^1\|_F^2 \quad (4.15)$$

where f^r is a row of U^r . The same term can also be written for X^2 :

$$\mathcal{O}_2 = \|X^2 - U^2 C^2 R^2\|_F^2 \quad (4.16)$$

As we would like to build a common space, the dimension for f^1 and f^2 should be the same. In the case of $\mathcal{O}_1$ and $\mathcal{O}_2$, we need $U^1 \in \mathbb{R}^{U \times K_1}$ and $U^2 \in \mathbb{R}^{L \times K_1}$. For $R^1 \in \mathbb{R}^{K_1 \times N}$ and $R^2 \in \mathbb{R}^{K_2 \times M}$, K_2 and K_3 do not need to be equal. However, we still set $K_2 = K_3$ in the following discussion, because column vectors usually share the same dimension with row vectors.

After introducing the basic embedding framework, let us consider the constraint term for the common space. For CHCS, we have the following optimization objective function which is widely used in machine learning techniques:

$$\mathcal{O}_3 = \sum_{(f_i, f_j) \in H} \|f_i' - f_j'\|_F^2 \quad (4.17)$$

In Equation 4.17, we try to minimize the sum of distance between embedded hints. As the result, the all the pairs of features in hint set will have similar vector representation in the common space. However, this optimization happens on the embedded vectors, which

$$P = \begin{bmatrix} \text{I} & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ \text{IO} & \text{II} & \text{I2} \end{bmatrix}$$

(a) Example Matrix

$$S = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

(b) Selection Matrix for selecting 4, 2-th row in order

$$SP = \begin{bmatrix} \text{IO} & \text{II} & \text{I2} \\ 4 & 5 & 6 \end{bmatrix}$$

(c) Matrix Selected by S

Figure 4.3: Example of the Selection Matrix

means the $f_i^{\mathcal{A}}$ and $f_j^{\mathcal{A}}$ will change during the optimization process. Thus, we need a way to select the hint from data matrix in each step for optimization. Here, we use the so-called “Selection Matrix” S , which can be used to select vectors from matrices. It is defined as follows:

To select a_1 -th, a_2 -th, $\dots$, a_u -th row of matrix X in this order, let $\mathcal{A} = [a_1, a_2, \dots, a_i, \dots, a_u]$ and we define the Selection Matrix:

$$S_{ij} = \begin{cases} 1, & \mathcal{A}_i = j \\ 0, & \text{otherwise} \end{cases} \quad (4.18)$$

Then the SX will be the matrix that containing a_1 -th, a_2 -th, ..., a_u -th vectors of X .

For example, to select the 4-th row and 2-nd row of the matrix P shown in Fig. 4.3a, we build the Selection Matrix as Fig. 4.3b. It is easy to validate that SP (Fig. 4.3c) is the matrix that we want.

Back to the case of Equation 4.17, because the hint set does not change, we are able to build selection matrices S^1, S^2 which select the embedded feature vectors from U^1 and U^2 .

Then, we can replace Equation 4.17 by

$$\mathcal{O}_3 = ||S^1 U^1 - S^2 U^2||_F^2 \quad (4.19)$$

This is the final formalization of CHCS. Now let us consider the ARelC. In fact, because the Tri-factorization tries to approximate the original data structure, this constraint has already been achieved at some level by Equation 4.15 and 4.16. However, in our previous researches, we experimently found that Tri-factorization is too weak to keep the distance relations in orginal data. Thus, we need a explicit constraint to preserve the “local structures” of each data.

Recently, it has been shown that the local structures can be effectively modeled by a nearest neighbor graph from the data points [BNo1]. That is: If two features are neighborhood in the original data, they shold also be neighborhood in the embedded space. Furthermore, the nearest neighbor graph has been introduced into NMF and non-negative tri-factorization techniques [Cai+11; PCS15]. These researches showed that the neighbor graph constraint can be actually written as a graph Laplacian. In our case, it is written as

$$\mathcal{O}_4 = Tr(U^1{}^T \mathcal{L}^1 U^1) \quad (4.20)$$

where U^1, U^2 denotes the embedded matrix, $Tr()$ denotes the trace of a matrix and $\mathcal{L}^1 = D^1 - W^1$ is the graph laplacian. W^1 is the Weight Matrix (Affinity Matrix) of the nearest neighbor graph and D^1 is a diagonal matrix where $D^1_{jj} = \sum_l W^1_{jl}$. The W^1 is constructed by

the following rules:

$$w_{ij}^1 = \begin{cases} 0, & |f_i^f - f_j^f|_F^2 > p \\ 1, & p \geq |f_i^f - f_j^f|_F^2 > q \\ 2, & q \geq |f_i^f - f_j^f|_F^2 \end{cases} \quad (4.21)$$

The weights are divided into 3 levels instead of using continuous value such as the euclidean distance. The reason is that to reduce the noises in the common space, we need all the unrelated features be far at the same level. If we use the euclidean distance, because the distance is unbalance, some unrelated fetatures will still be too near and finally become the noises.

Similarly, we have the same term on U^2 :

$$\mathcal{O}_5 = Tr(U^2 T \mathcal{L}^2 U^2) \quad (4.22)$$

Combining all the these objective functions $\mathcal{O}_1, \mathcal{O}_2, \mathcal{O}_3, \mathcal{O}_4, \mathcal{O}_5$, We have the final objective function for the common space embedding:

$$\begin{aligned} L(U^1, C^1, R^1, U^2, C^2, R^2) = & \\ & ||X^1 - U^1 C^1 R^1||_F^2 + ||X^2 - U^2 C^2 R^2||_F^2 + \\ & \lambda ||S^1 U^1 - S^2 U^2||_F^2 + \\ & \alpha Tr(U^1 \mathcal{L}^1 U^1) + \beta Tr(U^2 \mathcal{L}^2 U^2) \end{aligned} \quad (4.23)$$

As a summary, in this problem, $\mathcal{O}_1$ and $\mathcal{O}_2$ provide the basic embedding framework.

$\mathcal{O}_3$ is the constraint for build common space. $\mathcal{O}_4$ and $\mathcal{O}_5$ guarantee that the structures in original data are kept in the common space.

4.5 UPDATE RULES AND ALGORITHMS

In previous sections, we have presented the objective function for the common space emdedding as Equation 4.23. We will discribe the process for deriving the update rules in this section. It is easy to notice that all the constraints are only applied on U^1 and U^2 . Because the update rules come from the partial derivatives, we can simply use Equation 4.13 and Equation 4.14 as update rules for C^1 , C^2 , R^1 and R^2 . Therefore, we only need to consider the case for U^1 and U^2 . Furthermore, because the derivation processes for U^1 and U^2 are almost the same, we only present the case of U^1 in the following discussions.

Let γ_{ik}^1 be the Langrange multiplier for constraint l_{ij}^1 , and $\Gamma^1 = [\gamma_{ik}^1]$ the multiplier matrix. The Lagrange $\mathbb{L}$ is:

$$\begin{aligned} \mathbb{L} = & \\ & ||X^1 - U^1 C^1 R^1||_F^2 + \lambda ||S^1 U^1 - S^2 U^2||_F^2 + \\ & \alpha Tr(U^1 \mathcal{L}^1 U^1) + ||\Gamma^1 \odot U^1|| \end{aligned} \quad (4.24)$$

Here $\odot$ represents the Hadamard product (element-wise multiplication). One should be careful that because we are considering the case for U^1 , all the terms that do not contain U^1 are omitted. The Kuhn-Tucker conditions are:

$$\frac{\partial \mathbb{L}}{\partial U^1} = 0 \quad (4.25)$$

$$\Gamma^{\mathfrak{r}} U^{\mathfrak{r}} = \mathfrak{o} \quad (4.26)$$

The partial derivative of $\mathbb{L}$ with respect to $U^{\mathfrak{r}}$ is:

$$\begin{aligned} \mathfrak{o} = \frac{\partial \mathbb{L}}{\partial U^{\mathfrak{r}}} = & \\ & 2X^{\mathfrak{r}} R^{\mathfrak{r}T} C^T - 2U^{\mathfrak{r}} C^{\mathfrak{r}} R^{\mathfrak{r}} R^{\mathfrak{r}T} C^T + \\ & S^{\mathfrak{r}T} S^{\mathfrak{r}} U^{\mathfrak{r}} - S^{\mathfrak{r}T} S^2 U^2 + 2\mathcal{L} U^{\mathfrak{r}} + \Gamma^{\mathfrak{r}} \end{aligned} \quad (4.27)$$

Applying the Hadamard multiplication of $U^{\mathfrak{r}}$ on both side of $\frac{\mathbb{L}}{\partial U^{\mathfrak{r}}}$ and expanding laplacian $\mathcal{L}$ to $D - \mathcal{W}$, we get:

$$\begin{aligned} & 2X^{\mathfrak{r}} R^{\mathfrak{r}T} C^T \odot U^{\mathfrak{r}} + 2\mathcal{D} U^{\mathfrak{r}} \odot U^{\mathfrak{r}} + \\ & \Gamma^{\mathfrak{r}} \odot U^{\mathfrak{r}} + S^{\mathfrak{r}T} S^{\mathfrak{r}} U^{\mathfrak{r}} \odot U^{\mathfrak{r}} = \\ & S^{\mathfrak{r}T} S^2 U^2 \odot U^{\mathfrak{r}} + 2\mathcal{W} U^{\mathfrak{r}} \odot U^{\mathfrak{r}} + 2U^{\mathfrak{r}} C^{\mathfrak{r}} R^{\mathfrak{r}} R^{\mathfrak{r}T} C^T \odot U^{\mathfrak{r}} \end{aligned} \quad (4.28)$$

Then by using Equation 4.26, we can get the update rule of $U^{\mathfrak{r}}$.

We propose the following update rules:

$$U_{ij}^{\mathfrak{r}} \leftarrow U_{ij}^{\mathfrak{r}} \frac{(2X^{\mathfrak{r}} R^{\mathfrak{r}T} C^T + 2\mathcal{D} U^{\mathfrak{r}} + S^{\mathfrak{r}T} S^{\mathfrak{r}} U^{\mathfrak{r}})_{ij}}{(S^{\mathfrak{r}T} S^2 U^2 + 2\mathcal{W} U^{\mathfrak{r}} + 2U^{\mathfrak{r}} C^{\mathfrak{r}} R^{\mathfrak{r}} R^{\mathfrak{r}T} C^T)_{ij}} \quad (4.29)$$

$$C_{ij}^{\mathfrak{r}} \leftarrow C_{ij}^{\mathfrak{r}} \frac{(U^{\mathfrak{r}} X^{\mathfrak{r}} R^{\mathfrak{r}T})_{ij}}{(U^{\mathfrak{r}T} U^{\mathfrak{r}} C^{\mathfrak{r}} R^{\mathfrak{r}} R^{\mathfrak{r}T})_{ij}} \quad (4.30)$$

$$R_{ij}^1 \leftarrow R_{ij}^1 \frac{(C^T U^1 X^1)_{ij}}{(C^T U^1 U^1 C^1 R^1)_{ij}} \quad (4.31)$$

$$U_{ij}^2 \leftarrow U_{ij}^2 \frac{(2X^2 R^2 C^2 T + 2\mathcal{X} U^2 + S^2 T S^2 U^2)_{ij}}{(S^2 T S^2 U^2 + 2\mathcal{W} U^2 + 2U^2 C^2 R^2 R^2 T C^2 T)_{ij}} \quad (4.32)$$

$$C_{ij}^2 \leftarrow C_{ij}^2 \frac{(U^2 X^2 R^2 T)_{ij}}{(U^2 T U^2 C^2 R^2 R^2 T)_{ij}} \quad (4.33)$$

$$R_{ij}^2 \leftarrow R_{ij}^2 \frac{(C^2 T U^2 X^2)_{ij}}{(C^2 T U^2 T U^2 C^2 R^2)_{ij}} \quad (4.34)$$

With these update rules, we present our common space embedding algorithm in Fig. 3. In this algorithm, when one matrix is being updated, all other matrices are kept fixed. We repeat optimizing all the matrices until the change of loss function is small enough or the maximum iteration is reached. After this optimization, we can get the common space for inference new associations.

Algorithm 3 Algorithm

- 1: procedure CS-TRIFACTORIZATION(Dimension K^1, K^2 , Parameter p, q, λ, α and β , Hint Set H)
 - 2: Randomly initialize $U^1, C^1, R^1, U^2, C^2, R^2 \geq 0$
 - 3: Build selection matrix S^1 and S^2 according to hint set H
 - 4: Build graph laplacian $\mathcal{L}^1$ and $\mathcal{L}^2$
 - 5: while not converge do
 - 6: Update R^1 and R^2 by Equation 4.31 and Equation 4.34
 - 7: Update U^1 and U^2 by Equation 4.29 and Equation 4.32
 - 8: Update C^1 and C^2 by Equation 4.30 and Equation 4.31
 - 9: end while
 - 10: end procedure
-

4.6 LINEAR ALGEBRAIC INFERENCE

In Section 4.1, we have already discussed the basic idea of the inference. Because the common space is built based on user's hints, we can say that other data are decomposed into the axes which comes from user's hints. Thus, as the shown in Figure 4.1, if two data points have the similar weights on hints, we can say they are similar one. Even though the real axes of this common space is not hints themselves, this will not affect the above discuss. Suppose we have two data points d_1 and d_2 :

$$d_1 = \sum_i w_i h_i \quad (4.35)$$

$$d_2 = \sum_j k_j h_j \quad (4.36)$$

Here, w_i and k_i is the weights when we decompose d_1 and d_2 to each hints. Now let's consider the "real" axes a_i in the common space, we can write these "real" axes a_i as the combination of hints:

$$a_i = \sum_v l_v h_v + \sum_u m_u^i p_u \quad (4.37)$$

In this equation, a_i is decomposed two parts. The first part is the combination of hints. However, according to Equation 4.23, during our optimization, all the data are used at the mean time. Thus, we suppose all the other terms which affect the axes are summarized as the second part. In the common space, if two data point d_1 and d_2 have the same vector in

the common space (i.e. same weights on the axes), which means that:

$$\begin{aligned}
d_1 &= \sum_i w_i a_i \\
&\quad \sum_i w_i (\sum_v \dot{l}_v h_v + \sum_u m_u^i p_u) \\
&\quad \sum_i (\sum_v w_i \dot{l}_v h_v + \sum_u w_i m_u^i p_u)
\end{aligned} \tag{4.38}$$

$$\begin{aligned}
d_2 &= \sum_i k_i a_i \\
&\quad \sum_i k_i (\sum_v \dot{l}_v h_v + \sum_u m_u^i p_u) \\
&\quad \sum_i (\sum_v k_i \dot{l}_v h_v + \sum_u k_i m_u^i p_u)
\end{aligned} \tag{4.39}$$

Because $w_i = k_i$, it is easy to know that:

$$w_i \dot{l}_v h_v = k_i \dot{l}_v h_v \tag{4.40}$$

This means that d_1 and d_2 will also have exactly the same weights on the hints. Thus, we can say that if two data points is near in the common space, they will be associated in the meaning which is discussed in Section 4.1. As the result, our inference can be done actually based on finding nearest vectors.

Additional, we have point out that if two data points are far from each other, we can not say they are not associated. The reason is that these two data may have similar weights on

hints, but totally different weights on the other parts (i.e. $m_u^i p_u$ in the axes).

4.7 EXPERIMENT

4.7.1 DATASET

In the experiments, we use the news articles from Reuter Multilingual Corpus (RCV2)[†] [Lew+04]. These corpus contains the Reuters News stories from 1996-08-20 to 1997-08-19 in 13 languages. The stories are not parallel, which means that they are written by local reporters in each language. Thus, we can treat the article sets in different languages as different datasets. In the experiment, we use the news articles in Italian and Spanish. Firstly, we do morphological analysis by using tree-tagger [Sch13]. Furthermore, we found that the dataset contains many article which only have one picture or a table. To remove these type of news articles, we only keep the articles that contains more than 5 nouns. Because of the hardware and time limitations, not all articles are used. We randomly sample 6,000 articles from both two languages as well as their nouns. We use the bag-of-words to construct the word vectors as features. To build the hint set, we use the Google Cloud Translation API[‡]. Both spanish and italian nouns are translated to each other. If both the original nouns and translated word exist in the dataset, the pair of them is used as a hint. Finally, we use the algorithm in Fig. 3 to embed all the nouns into the common space. Detail statistics of the data is shown in Table 4.1. The parameters used in this experiment are shown in Table 4.2. λ controls the effect level of hints. α and β is the weight of the local structure in each datasets. If λ is too large, we can only find trivial associations. However, α and β should not be too

[†]<http://trec.nist.gov/data/reuters/reuters.html>

[‡]<https://cloud.google.com/translate/>

larger than λ to avoid noises in the common space. We performed several preliminary experiments and select the best combination of α , β and λ .

	Italian	Spanish
# of Nouns	4,906	4,502
# of Articles	6,000	6,000
# of Total Hints	4,502	4,502

Table 4.1: Statistics of Dataset

Parameter	Value
# of Used Hint	500
Dim. of Word Common Space	300
Dim. of Article Common Space	300
α	10.0
β	10.0
λ	80.0
p	0.3
q	0.1

Table 4.2: Parameters

4.7.2 PERFORMANCE AND ACCURACY

The change of error (i.e. value of cost function) is shown in Figure 4.4. Similar to other machine learning algorithms, error reduces fastly at the first 100 steps. With the optimization, it finally becomes stable and almost does not change after 200 steps. This means that the distances between points only change slightly at each step. Furthermore, for our inference, we only require the neighborhood information about the points. We do not care about the real distance. In the real-world application, we can just stop the optimization at 200 steps and get an acceptable result. In this case, the whole optimization will just take about

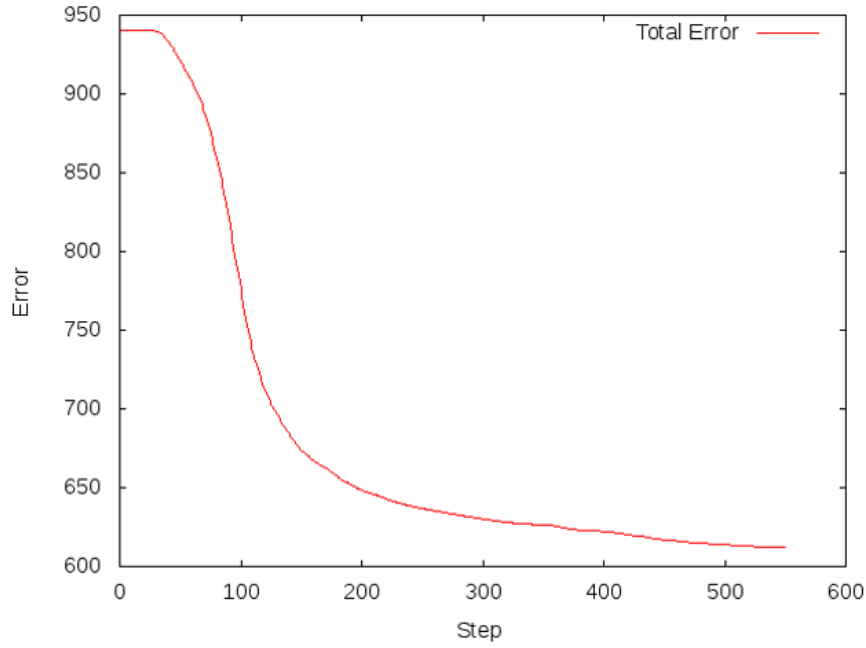


Figure 4.4: Change of Cost Function

5 minutes on a normal workstation. We can expect this algorithm will also work on larger datasets.

Furthermore, we also evaluated the accuracy of our inference algorithm. The details of the experiment is shown as follows:

- Term Level Accuracy (TLA)

For each term, we take the Top- k nearest terms in the other languages for this term.

If the top- k contains a correct answer (i.e. translated term except the hints), they are marked as correct. This process is done for all Italian terms and the final accuracy is calculated by $TLA = \frac{\#ofcorrectterms}{\#ofterms}$.

- Cluster Level Accuracy (CLA)

As we have discussed in the technical parts, the Tri-factorization can also be considered as a clustering algorithm. In this experiment, for each term, we take the index of the largest value in their vector as their cluster to build several common clusters. After this, we check if there exists more than k correct pairs (except the hints, e.g. an Italian term and its translated Spanish term) in a common cluster. If a common cluster contains more than k correct pairs, we mark this cluster as correct. This process is done for all common clusters and the final accuracy is calculated by

$$CLA = \frac{\#ofcorrectclusters}{\#ofcommonclusters}.$$

The experiment results is shown in Table 4.3 and Table 4.4. We can notice that the TLA does not achieve a good result. There are two reasons to cause this fact: (1) In this experiment, we only consider single terms. However, in real language, the translated term may contain several words. Our algorithm currently can not handle this case. (2) Our algorithm is not designed to find the translated terms. Because the number of hints is much smaller than the total terms, it is impossible to catch the meaning of all the terms. In many cases, our algorithm can find the terms which are very related but not the translation. We will also talk about this in the following sections. In contrast, we can also notice that the CLA has a much better performance. This means that for almost all the common clusters, there exists at least one correct pairs. This fact shows the potential that our algorithm may work better on TLA by future optimization. In the mean time, to validate the effect of Laplacian Constraints, we also measure the accuracy with and without laplacian constraints. It is clearly that the version with laplacian constraint outperforms the version without laplacian constraint. This shows that the laplacian constraint does have the effect to reduce the noises.

Top k	With Laplacian Constraint	Without Laplacian Constraint
10	14.2%	8.0%
20	14.4%	8.0%
30	18.0%	8.1%

Table 4.3: Term Level Accuracy

At least k	With Laplacian Constraint	Without Laplacian Constraint
1	81.0%	75.0%
3	73.2%	61.1%
5	57.8%	43.0%

Table 4.4: Cluster Level Accuracy

4.7.3 ASSOCIATIONS DISCOVERY

To discover new associations in the common space, for each Italian word, we take the nearest 4 Spanish nouns as the associated words. As shown in Table 4.5, it is easy to see that many meaningful associations are discovered. For example, around StatUnitense (United States in English), we can find the name of states in America, such as IOWA and ILLINOIS as well as the agency NASA, which are closely related America. Coalizione means coalition in English. We can find many coalitions appears around the term coalizione, such as GSM (Global System for Mobile communications) and LatinoAmérica (Latin America). For the Italian polmonite, which means pneumonia in English, our method finds the associations containing the transmission route (mosquito), symptom (calambre, i.e. muscle cramps) and the infect source (herpes virus). It is clear that none of the detected terms appears in the hints. This fact shows that our algorithm is able to find potential associations across two languages.

Italian	Spanish
StatUnitense	IOWA, ILLINOIS, COLISIÓN, NASA
Coalizione	IBERIA, GSM, LATINOAMÉRICA, USA
Polmonite	MOSQUITO, CALAMBRE, HERPES, VIRUS
Fibra	AAA, GREGORIO, CONC, LUX
SEMICONDUTTORE	BOSCH, ESTRECHEZ, ATISBO, ESTRANGULAMIENTO
MOBILIA	NTT, MOTIVACIÓN, CONSPIRACIÓN, VEJEZ

Table 4.5: Example of Associated Nouns

4.8 CONCLUSIONS

In this chapter, we proposed a common space embedding algorithm for discovering associations. Given a small set of associated feature pairs, our method construct a common space where the associated features have the same vector representation. Moreover, by preserving the distance relation in the original data sets, remaining features can be embedded in this common space. As the result, from the view point of hints, similar features will be embedded near to each other. Thus, we can discover new associations by looking into the neighborhood of features. The experiment on news articles shows that this method are able to find meaningful associations.

5

Conclusions and Future Work

In this thesis, we have investigated discovery and exploration systems considering the user's intention. To compress the data and retain various user's intentions about the sizes of discovery targets, we focused on pseudo-cliques in Chapter 1 and Chapter 2.

In Chapter 1, we defined a MCS as a family of overlapping cliques in a graph. MCS must be a complete pseudo-clique in the original graph. To find MCS, we first make the standard

clique enumerator run under the constraint of a lower-bound size, so MCS appears as a pseudo-clique formed by cliques in the clique graph. Our algorithm can prune any clique from the clique graph if it fails the test of pseudo-cliqueness in the original graph. Thus, we can design a very effective MCS enumerator based on the techniques for fast clique enumeration.

In Chapter 2, we designed an efficient and complete algorithm for enumerating (j, k) -MPCs that are densely connected pseudo-cliques satisfying a lower-bound connection constraint (LBC). We proved that any tentative vertex set that can be extended to (j, k) -MPCs is included in the j -core of the tentative set together with its candidate vertices. We can, therefore, implement a pruning rule to cut off undesirable vertex sets that have no chance of being extended to the (j, k) -MPCs. Because j -coreness can be checked in linear time, we succeeded in designing a very fast and complete enumerator that discards undesirable vertex sets in its early stages of enumeration. Experimental results showed that our algorithms can be a practical tool for large real-world datasets.

The user's intention discussed in Chapter 1 and Chapter 2 is concerned with only the size and density of discovery targets. On the other hand, it is generally agreed that users may not know their intentions clearly at the beginning of the retrieval and discovery process. Some associative search engines are often used to help the user realize their real intention. In Chapter 3, we described a new framework of associative search. A remarkable point of the framework is that the associative search is controlled by selecting people, called mediators, who can guide the user to various discourses while preserving the relevance to the user's initial query. Based on the techniques for seeking formal concepts, we designed a system of depth-first algorithms for forming appropriate bridge concepts under which potential

target concepts obtained by modifying the original query concept can be computed and shown to the user. The bridge concepts are dependent on the selection of mediators by the user. In this sense, our associative search preserves the user’s intention by referring to mediators with the same or similar intentions as the user.

In Chapter 4, we introduced a linear algebraic inference algorithm for finding feature associations based on the user’s intention given in the form of hints. We proposed a common-space embedding algorithm using the nonnegative trifactORIZATION framework for this purpose. We validated both the performance of the algorithm and the quality of the results through experiments on real-world datasets.

However, there are several problems remaining to be explored and solved.

- The integrated whole system with pseudo-clique enumerators as its subsystems must be designed and tested.

It is obviously possible to utilize the pseudo-clique methods as preprocessing algorithms for discovery. For example, in the associative search in Chapter 3, we have used the extended the k -means method for local clusters. Because such local clusters will appear as pseudo-cliques and the pseudo-clique enumerator can find more potential clusters than the extended k -means technique can, we can simply replace k -means with pseudo-cliques for the integration.

In addition to the direct integration mentioned in the above, we believe that more elegant ways exist for building the final system for better performance. One possible direction is that the pseudo-cliques are designed to preserve the user’s possible intentions as much as possible. However, during the discovery process, we can clearly filter out many intentions (i.e., clusters/pseudo-cliques) according to additional in-

formation provided by users. This means that there is no need to consider all the possibilities. Instead, we could dynamically generate the intentions in the form of clusters during the search process.

- Many k -plexes are highly overlapped.

As we have pointed out in Chapter 1 Section 1.5, *MetaClique* may detect many highly overlapped communities. These communities usually differ by only one or two vertices. This is because many primitive components (i.e., cliques) are highly overlapped. If these highly overlapped components cannot form one metaclique in total, according to the property of the k -plex, multiple combinations of these components may exist to form metacliques. Isolated cliques and pseudo-cliques [H09] provide a good solution to reduce these highly overlapped components. A clique is said to be c -isolated if it contains no more than c outgoing edges. Highly overlapped cliques usually have many outgoing edges that are connected to those different vertices. Thus, by restricting the primitive components to the isolated pseudo-cliques, the highly overlapped components will be merged into one big component. j - k -MPC also suffers from similar problems. Many j - k -MPCs are highly overlapped and differ by only one vertex. We can consider isolation as a constraint based on a notion of *variable- j -coreness* [OHT16] as with the MCS. Because the number of isolated pseudo-cliques is usually much smaller than the number of cliques in real-world graphs, a performance improvement can be expected by using isolated pseudo-cliques as primitive components.

- A deeper theoretical analysis is necessary.

In Chapter 1, the theoretical analysis of computational complexity for MCS remains a future work. Because others [TTTo6] have already investigated the worst-case time complexity for CLIQUES, the analysis may be similar. In the research on associative search, we proposed a novel k -means algorithm called extended k -means. Our extended k -means accepts a parameter σ . Clusters with variances larger than σ will be divided into smaller clusters. The computational complexity of the extended k -means is not shown in this thesis. However, as the time complexity of traditional k -means has been proved in [AVo6] and [Vato9], a similar analysis is possible for the extended k -means.

- The performance of MCS can be improved with recent techniques.

[Con+17] presents a maximal k -plex enumerator with several new filtering criteria, such as Cliqueness (any node of a k -plex of size m must be included in a clique of size at least $\frac{m}{k} \in \mathbb{N}$). Our MCS algorithm can take advantage of these criteria. Suppose we have a clique set $\mathcal{M}$ and a candidate clique C ; based on the cliqueness criterion, we can filter out the clique safely if $|C| < \frac{|\mathcal{M} \cup C|}{k}$. This criterion allows our MCS algorithm to cut off less desirable cliques before constructing the k -clique graph to achieve better performance. The j -core constraint introduced in [Con+17] and j - k -MPC can also be applied to MCS. The j -cored MCS notion will be much denser than the normal MCS, so this notion may work even better on the Protein–Protein Interaction Network.

- Parallel algorithms could be considered.

Recently, parallel or distributed algorithms have attracted much attention in the re-

search areas of pseudo-clique and formal concept analysis [FNo4; Du+o6; Sch+o9; WPo7; KOVo8; Xu+12]. We would like to investigate the development of parallel algorithms for our proposals in Chapters 1, 2, and 3, especially for the MCS in Chapter 1. Because the MCS algorithms are based on clique enumeration, it should be easy to extend them to parallel versions based on existing parallel clique enumeration algorithms.

In the study of feature association discovery in Chapter 4, our experiments showed that nonnegative trifactORIZATION works well with $\sim 10,000$ features. However, the algorithms of the NMF family require all the matrices, including the input data and factorized matrices, to be kept in memory. If data matrices become really big (e.g., more than $50,000 \sim$ features), it is difficult to perform the nonnegative trifactORIZATION because of the limitation of memory resources. Fortunately, an online version of NMF has been proposed in the literature [LBFI]. This online NMF can split the input dataset into smaller groups. Such an online algorithm would provide more efficient common-space embedding algorithms for bigger data matrices.

- Feature association discovery may make use of more language information.

As we discussed in Chapter 0 and 4, the feature association discovery algorithm can be used to solve the bilingual word-embedding problem. However, the experiments on Reuters news articles showed that the accuracy of the algorithm is still not adequate for real-world applications. One major reason is that we did not use any information other than the word co-occurrences (bag-of-documents representation). This representation does not contain context information, which is extremely important in the translation task. Thus, our next step will be to introduce more language infor-

mation into the feature association discovery algorithm.

Recently, deep learning based on neural networks has achieved some success in word embedding, such as the neural probabilistic language model [Ben+03] and word2vec [Mik+13].

These methods learn to produce word vectors considering their contexts by analyzing large-scale data. There are two kinds of contexts. The first is the general context, which simply catches the general meaning of the words. “word2vec” [Mik+13] is a famous algorithm for learning these vectors. A word will always have only one vector representation regardless of the meaning of the sentence containing it. The second is the local context. Vectors with local context are not unique. The learning algorithm will generate the vector for a word based on the current topics or sentence. The algorithms for learning local context are usually based on a recurrent neural network, such as the long short-term memory network [HS97] and the RNN-RBM [BBV12].

One straightforward way to enjoy the advantages of deep learning techniques is to replace the input matrix of our original algorithm with vectors containing general context information. With these vectors, we can perform common-space embedding with the context information without modifying our NMF algorithm. Unfortunately, current NMF algorithms cannot handle local context information. We must extend the NMF framework itself for deeper integration between deep learning and common-space embedding.

- Multiple feature associations could be considered.

The proposed algorithm for feature association handles all the associations in the same way. In the current version of the algorithm, even if two features “A” and “B”

have the association “R” while “A” and “C” have the association “K,” both (“A,” “B”) and (“A,” “C”) will be mapped in the same manner. As a result, “B” and “C” will both appear as neighbors to “A,” and we cannot distinguish the different associations from the embedded vectors. To solve this problem, we would like to consider multiple feature associations in the algorithm. In fact, a machine learning technique called knowledge-graph embedding [Wan+14; Lin+15] can handle multiple associations. Unfortunately, these knowledge-graph embedding techniques take totally different approaches that cannot be applied to NMF. However, by using the ideas of knowledge-graph embedding, we can use the distances to distinguish the relations. That is, in the common space, “A” and “B” will never be mapped to the same place. Instead, they will have a distance d_R , which depends on the relation “R.” “A” and “C” will also have a distance d_K and $d_R \neq d_K$. When we infer the words that have relation R to a word $\mathcal{A}$, we can simply find all the words with distances d_R from $\mathcal{A}$. Technically, in the original algorithm, the constraint for constructing the common space is $\|S^1 U^1 - S^2 U^2\|_F^2$ (in Chapter 4 Equation 4.19), where $S^1 U^1$ and $S^2 U^2$ are the hints vectors. To represent relations R^i ($i \in \{1, 2, \dots, n\}$), we consider the following constraint instead of those above:

$$\sum_i \|S_i^1 U_i^1 - S_i^2 U_i^2 + \xi i\|_F^2 \quad (5.1)$$

Here, $S_i^1 U_i^1$ and $S_i^2 U_i^2$ are the hints that have relation R^i ($i \in \{1, 2, \dots, n\}$). ξ is a pre-defined parameter that controls the margins between the different relations. This approach divides the relations into hyperspheres with different diameters. For exam-

ple, for a word w , if we would like to find the associated words with relation R^1 , we can search for the words u for which $1 \geq |u - w| \geq 0$. Similarly, the words u for which $5 \geq |u - w| \geq 4$ will have the relation R^4 with w .

The biggest challenge of this approach may be noise. In contrast to knowledge-graph embedding, which can use negative samples, we cannot use negative information in NMF without breaking its good properties. Handling rare relations will be our next task.

- Dynamic (interactive) hints for feature association discovery.

The feature association discovery algorithm uses a fixed set of associated features as the hints. However, according to the discussion of the association search (Chapter 3), if users can interactively control the direction of the search, the algorithm will be able to find targets that are more related to the users' real intention. One way to allow users to control the search direction for feature association is to allow the hints to be dynamically modified. That is, we first find some associations according to a predefined hint set. Given these found associations, users select others that they are interested in and add them to the hint set. This process can be repeated several times until users finally find their targets.

For our feature association discovery algorithm, we know that the constraint for the common space is $\sum_{ij} \|u_i - u_j\|$, where u_i and u_j are the vectors of the associated features. This can also be written as $\sum_{ij} w_{ij} \|u_i - u_j\|$, $w_{ij} = 1$ if (u_i, u_j) is a hint, otherwise $w_{ij} = 0$. [Cai+11] showed that besides the selection matrix, this type of constraint can also be formalized as a Laplacian constraint. The Laplacian is con-

structed from a bipartite graph in which vertices are the features and two vertices are only connected if they form a hint. From the definition of the Laplacian, it is easy to show that adding new hints is equivalent to adding new edges to the bipartite graph. Thus, the optimization algorithm for the dynamic hints is exactly the same as that for the fixed hints. However, because the Laplacian (constraint for the common space) changes during the optimization, the convergence of the algorithm should be investigated.

- Guidelines for selecting mediators and hints are necessary.

In the discovery tasks (Chapter 3 and Chapter 4), the mediators and hints will directly affect the final results. A good set of mediators (or hints) will cover the user's intention while a bad one may provide no information to the system. Thus, the selection of mediators and hints is very important. We would like to develop guidelines for their selection. Of course, if there are too many constraints on the selection, the process will become hard to use and many interesting mediators and hints may be ruled out. Thus, the guidelines should be made as simple as possible.



Permission to Use

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Feb 14, 2018

This Agreement between Hokkaido University ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4286930037913
License date	Feb 13, 2018
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Shifting Concepts to Their Associative Concepts via Bridges
Licensed Content Author	Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo et al
Licensed Content Date	Jan 1, 2013
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	<501
Author of this Springer Nature content	yes
Title	Study on Discovery and Exploration Systems Considering User's Intention
Instructor name	Makoto Haraguchi
Institution name	Graduate School of Information Science and Technology, Hokkaido University
Expected presentation date	Feb 2018
Requestor Location	Knowledge-base Laboratory Graduate School of IST Hokkaido University Kita 14, Nishi 9, Kita-ku, Sapporo, Hokkaido 001-0031 Japan Attn: Hongjie Zhai
Billing Type	Invoice
Billing Address	Hokkaido University Kita 14, Nishi 9, Kita-ku, Graduate School of IST, Hokkaido University Sapporo, Japan 001-0031 Attn: Hongjie Zhai
Total	0 JPY
Terms and Conditions	

Springer Nature Terms and Conditions for RightsLink Permissions

Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However,

you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where **print only** permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.
3. Permission granted **free of charge** for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.
4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.
5. Where '**reuse in a dissertation/thesis**' has been selected the following terms apply: Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).
6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines <http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/>), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.
7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com for these rights.
8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.
9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.
10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

For Advance Online Publication papers:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)]

For Adaptations/Translations:

Adapted/Translated by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [the Licensor]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s)

Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication** papers:

Reprinted by permission from The [the Licensors]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:

Reprinted/adapted by permission from [the Licensors]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Feb 14, 2018

This Agreement between Hokkaido University ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4286930255741
License date	Feb 13, 2018
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Enumerating Maximal Clique Sets with Pseudo-Clique Constraint
Licensed Content Author	Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo et al
Licensed Content Date	Jan 1, 2015
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	<501
Author of this Springer Nature content	yes
Title	Study on Discovery and Exploration Systems Considering User's Intention
Instructor name	Makoto Haraguchi
Institution name	Graduate School of Information Science and Technology, Hokkaido University
Expected presentation date	Feb 2018
Requestor Location	Knowledge-base Laboratory Graduate School of IST Hokkaido University Kita 14, Nishi 9, Kita-ku Sapporo, Hokkaido 001-0031 Japan Attn: Hongjie Zhai
Billing Type	Invoice
Billing Address	Hokkaido University Kita 14, Nishi 9, Kita-ku, Graduate School of IST, Hokkaido University Sapporo, Japan 001-0031 Attn: Hongjie Zhai
Total	0 JPY
Terms and Conditions	

Springer Nature Terms and Conditions for RightsLink Permissions

Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However,

you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where **print only** permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.
3. Permission granted **free of charge** for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.
4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.
5. Where '**reuse in a dissertation/thesis**' has been selected the following terms apply: Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).
6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines <http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/>), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.
7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com for these rights.
8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.
9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.
10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

For Advance Online Publication papers:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)]

For Adaptations/Translations:

Adapted/Translated by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [the Licensor]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s)

Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication** papers:

Reprinted by permission from The [the Licensors]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:

Reprinted/adapted by permission from [the Licensors]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

SPRINGER NATURE LICENSE TERMS AND CONDITIONS

Feb 14, 2018

This Agreement between Hokkaido University ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	4286930420734
License date	Feb 13, 2018
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	A Fast and Complete Enumeration of Pseudo-Cliques for Large Graphs
Licensed Content Author	Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo et al
Licensed Content Date	Jan 1, 2016
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	print and electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	<501
Author of this Springer Nature content	yes
Title	Study on Discovery and Exploration Systems Considering User's Intention
Instructor name	Makoto Haraguchi
Institution name	Graduate School of Information Science and Technology, Hokkaido University
Expected presentation date	Feb 2018
Requestor Location	Knowledge-base Laboratory Graduate School of IST Hokkaido University Kita 14, Nishi 9, Kita-ku Sapporo, Hokkaido 001-0031 Japan Attn: Hongjie Zhai
Billing Type	Invoice
Billing Address	Hokkaido University Kita 14, Nishi 9, Kita-ku, Graduate School of IST, Hokkaido University Sapporo, Japan 001-0031 Attn: Hongjie Zhai
Total	0 JPY
Terms and Conditions	

Springer Nature Terms and Conditions for RightsLink Permissions

Springer Customer Service Centre GmbH (the Licensor) hereby grants you a non-exclusive, world-wide licence to reproduce the material and for the purpose and requirements specified in the attached copy of your order form, and for no other use, subject to the conditions below:

1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of this material. However,

you should ensure that the material you are requesting is original to the Licensor and does not carry the copyright of another entity (as credited in the published version).

If the credit line on any part of the material you have requested indicates that it was reprinted or adapted with permission from another source, then you should also seek permission from that source to reuse the material.

2. Where **print only** permission has been granted for a fee, separate permission must be obtained for any additional electronic re-use.
3. Permission granted **free of charge** for material in print is also usually granted for any electronic version of that work, provided that the material is incidental to your work as a whole and that the electronic version is essentially equivalent to, or substitutes for, the print version.
4. A licence for 'post on a website' is valid for 12 months from the licence date. This licence does not cover use of full text articles on websites.
5. Where '**reuse in a dissertation/thesis**' has been selected the following terms apply: Print rights for up to 100 copies, electronic rights for use only on a personal website or institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/).
6. Permission granted for books and journals is granted for the lifetime of the first edition and does not apply to second and subsequent editions (except where the first edition permission was granted free of charge or for signatories to the STM Permissions Guidelines <http://www.stm-assoc.org/copyright-legal-affairs/permissions/permissions-guidelines/>), and does not apply for editions in other languages unless additional translation rights have been granted separately in the licence.
7. Rights for additional components such as custom editions and derivatives require additional permission and may be subject to an additional fee. Please apply to Journalpermissions@springernature.com for these rights.
8. The Licensor's permission must be acknowledged next to the licensed material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract, and must be hyperlinked to the journal/book's homepage. Our required acknowledgement format is in the Appendix below.
9. Use of the material for incidental promotional use, minor editing privileges (this does not include cropping, adapting, omitting material or any other changes that affect the meaning, intention or moral rights of the author) and copies for the disabled are permitted under this licence.
10. Minor adaptations of single figures (changes of format, colour and style) do not require the Licensor's approval. However, the adaptation should be credited as shown in Appendix below.

Appendix — Acknowledgements:

For Journal Content:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

For Advance Online Publication papers:

Reprinted by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM].)]

For Adaptations/Translations:

Adapted/Translated by permission from [the Licensor]: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication)]

Note: For any republication from the British Journal of Cancer, the following credit line style applies:

Reprinted/adapted/translated by permission from [the Licensor]: on behalf of Cancer Research UK: : [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s)

Name), [COPYRIGHT] (year of publication)

For **Advance Online Publication** papers:

Reprinted by permission from The [the Licensors]: on behalf of Cancer Research UK: [Journal Publisher (e.g. Nature/Springer/Palgrave)] [JOURNAL NAME] [REFERENCE CITATION (Article name, Author(s) Name), [COPYRIGHT] (year of publication), advance online publication, day month year (doi: 10.1038/sj.[JOURNAL ACRONYM])

For Book content:

Reprinted/adapted by permission from [the Licensors]: [Book Publisher (e.g. Palgrave Macmillan, Springer etc) [Book Title] by [Book author(s)] [COPYRIGHT] (year of publication)

Other Conditions:

Version 1.0

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Bibliography

- [Luc50] R Duncan Luce. “Connectivity and generalized cliques in sociometric group structure”. In: *Psychometrika* 15.2 (1950), pp. 169–190.
- [Hes66] Mary Hesse. *Models and Analogies in Science, 2nd edition*. Notre Dame University Press, 1966.
- [Mac67] James MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.
- [Ham68] Ronald C Hamelink. “A partial characterization of clique graphs”. In: *Journal of Combinatorial Theory* 5.2 (1968), pp. 192–197.
- [Alb73] Richard D Alba. “A graph-theoretic definition of a sociometric clique”. In: *Journal of Mathematical Sociology* 3.1 (1973), pp. 113–126.
- [BK73] Coen Bron and Joep Kerbosch. “Algorithm 457: finding all cliques of an undirected graph”. In: *Communications of the ACM* 16.9 (1973), pp. 575–577.
- [SF78] Stephen B Seidman and Brian L Foster. “A graph-theoretic generalization of the clique concept*”. In: *Journal of Mathematical sociology* 6.1 (1978), pp. 139–154.
- [Mok79] Robert J Mokken. “Cliques, clubs and clans”. In: *Quality & Quantity* 13.2 (1979), pp. 161–173.
- [Gen83] Dedre Gentner. “Structure-mapping: A theoretical framework for analogy”. In: *Cognitive science* 7.2 (1983), pp. 155–170.
- [Sei83] Stephen B Seidman. “Network structure and minimum degree”. In: *Social networks* 5.3 (1983), pp. 269–287.
- [Gre85] Russell Greiner. “Learning by understanding analogies”. Doctoral Dissertation. Stanford University, 1985.

- [Jol86] Ian T Jolliffe. “Principal Component Analysis and Factor Analysis”. In: *Principal component analysis*. Springer, 1986, pp. 115–128.
- [Rym92] Ron Rymon. “Search through systematic set enumeration”. In: *Technical Reports (CIS)* (1992), p. 297.
- [PT94] Pentti Paatero and Unto Tapper. “Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values”. In: *Environmetrics* 5.2 (1994), pp. 111–126.
- [HS97] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.
- [BHK98] John S Breese, David Heckerman, and Carl Kadie. “Empirical analysis of predictive algorithms for collaborative filtering”. In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc. 1998, pp. 43–52.
- [WS98] Duncan J Watts and Steven H Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *nature* 393.6684 (1998), pp. 440–442.
- [BA99] Albert-László Barabási and Réka Albert. “Emergence of scaling in random networks”. In: *science* 286.5439 (1999), pp. 509–512.
- [LS99] Daniel D Lee and H Sebastian Seung. “Learning the parts of objects by non-negative matrix factorization”. In: *Nature* 401.6755 (1999), pp. 788–791.
- [Tak+00] Akihiko Takano, Yoshiki Niwa, Shingo Nishioka, Makoto Iwayama, Toru Hisamitsu, Osamu Imaichi, and Hirofumi Sakurai. “Information access based on associative calculation”. In: *SOFSEM*. Springer. 2000, pp. 187–201.
- [BNo1] Mikhail Belkin and Partha Niyogi. “Laplacian eigenmaps and spectral techniques for embedding and clustering”. In: *NIPS*. Vol. 14. 14. 2001, pp. 585–591.
- [LS01] Daniel D Lee and H Sebastian Seung. “Algorithms for non-negative matrix factorization”. In: *Advances in neural information processing systems*. 2001, pp. 556–562.
- [ARSo2] James Abello, Mauricio Resende, and Sandra Sudarsky. “Massive quasi-clique detection”. In: *LATIN 2002: Theoretical Informatics* (2002), pp. 598–612.
- [GNo2] Michelle Girvan and Mark EJ Newman. “Community structure in social and biological networks”. In: *Proc. of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826.

- [NJV02] Andrew Y Ng, Michael I Jordan, and Yair Weiss. “On spectral clustering: Analysis and an algorithm”. In: *Advances in neural information processing systems*. 2002, pp. 849–856.
- [BZ03] Vladimir Batagelj and Matjaz Zaversnik. “An $O(m)$ algorithm for cores decomposition of networks”. In: *arXiv preprint cs/0310049* (2003).
- [Ben+03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. “A neural probabilistic language model”. In: *Journal of machine learning research* 3.Feb (2003), pp. 1137–1155.
- [Nag03] Anna Nagurney. *Innovations in financial and economic networks*. Edward Elgar Publishing, 2003.
- [Omio3] Edward R Omiecinski. “Alternative interest measures for mining associations in databases”. In: *IEEE Transactions on Knowledge and Data Engineering* 15.1 (2003), pp. 57–69.
- [Chu+04] Moody Chu, Fasma Diele, Robert Plemmons, and Stefania Ragni. “Optimality, computation, and interpretation of nonnegative matrix factorizations”. In: *SIAM Journal on Matrix Analysis*. Citeseer. 2004.
- [FNo4] Huaiguo Fu and Engelbert Nguifo. “A parallel algorithm to generate formal concepts for large data”. In: *Concept Lattices* (2004), pp. 141–142.
- [Lew+04] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. “Rcv1: A new benchmark collection for text categorization research”. In: *Journal of machine learning research* 5.Apr (2004), pp. 361–397.
- [CSW05] Peter J Carrington, John Scott, and Stanley Wasserman. *Models and methods in social network analysis*. Vol. 28. Cambridge university press, 2005.
- [DHS05] Chris Ding, Xiaofeng He, and Horst D Simon. “On the equivalence of non-negative matrix factorization and spectral clustering”. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM. 2005, pp. 606–610.
- [LZY05] Bo Long, Zhongfei Mark Zhang, and Philip S Yu. “Co-clustering by block value decomposition”. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM. 2005, pp. 635–640.
- [VLo5] Fabien Viger and Matthieu Latapy. “Efficient and simple generation of random simple connected graphs with prescribed degree sequence”. In: *Computing and Combinatorics*. Springer, 2005, pp. 440–449.

- [ZPo5] Lihi Zelnik-Manor and Pietro Perona. “Self-tuning spectral clustering”. In: *Advances in neural information processing systems*. 2005, pp. 1601–1608.
- [AVo6] David Arthur and Sergei Vassilvitskii. “How Slow is the K-means Method?”. In: *Proceedings of the Twenty-second Annual Symposium on Computational Geometry*. SCG ’06. Sedona, Arizona, USA: ACM, 2006, pp. 144–153. ISBN: 1-59593-340-9.
- [BMo6] Vladimir Batagelj and Andrej Mrvar. *Pajek datasets*. <http://vlado.fmf.uni-lj.si/pub/networks/data/>. 2006.
- [Din+o6] Chris Ding, Tao Li, Wei Peng, and Haesun Park. “Orthogonal nonnegative matrix t-factorizations for clustering”. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2006, pp. 126–135.
- [Du+o6] Nan Du, Bin Wu, Liutong Xu, Bai Wang, and Xin Pei. “A parallel algorithm for enumerating all maximal cliques in complex network”. In: *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE. 2006, pp. 320–324.
- [HOo6] Makoto Haraguchi and Yoshiaki Okubo. “A method for pinpoint clustering of web pages with pseudo-clique search”. In: *Federation over the Web*. Springer, 2006, pp. 59–78.
- [Newo6] Mark EJ Newman. “Finding community structure in networks using the eigenvectors of matrices”. In: *Physical review E* 74.3 (2006), p. 036104.
- [TTTo6] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. “The worst-case time complexity for generating all maximal cliques and computational experiments”. In: *Theoretical Computer Science* 363.1 (2006), pp. 28–42.
- [AVo7] David Arthur and Sergei Vassilvitskii. “k-means++: The advantages of careful seeding”. In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics. 2007, pp. 1027–1035.
- [Col+o7] Sean R Collins, Patrick Kemmeren, Xue-Chu Zhao, Jack F Greenblatt, Forrest Spencer, Frank CP Holstege, Jonathan S Weissman, and Nevan J Krogan. “Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*”. In: *Molecular & Cellular Proteomics* 6.3 (2007), pp. 439–450.

- [Du+07] Nan Du, Bin Wu, Xin Pei, Bai Wang, and Liutong Xu. “Community detection in large-scale social networks”. In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM. 2007, pp. 16–25.
- [GMW07] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data clustering: theory, algorithms, and applications*. SIAM, 2007.
- [Lino7] Chih-Jen Lin. “Projected gradient methods for nonnegative matrix factorization”. In: *Neural computation* 19.10 (2007), pp. 2756–2779.
- [Von07] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and computing* 17.4 (2007), pp. 395–416.
- [WP07] Bin Wu and Xin Pei. “A parallel algorithm for enumerating all the maximal kplexes”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2007, pp. 476–483.
- [DDE08] Frithjof Dau, Jon Ducrou, and Peter Eklund. “Concept similarity and related categories in searchleuth”. In: *Lecture Notes in Computer Science* 5113 (2008), pp. 255–268.
- [KOV08] Petr Krajca, Jan Outrata, and Vilem Vychodil. “Parallel recursive algorithm for FCA”. In: *CLA*. Vol. 2008. 2008, pp. 71–82.
- [LHO08] Aixiang Li, Makoto Haraguchi, and Yoshiaki Okubo. “Implicit groups of web pages as constrained top N concepts”. In: *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society. 2008, pp. 190–194.
- [Car+09] Claudio Carpineto, Stanislaw Osinski, Giovanni Romano, and Dawid Weiss. “A survey of web clustering engines”. In: *ACM Computing Surveys (CSUR)* 41.3 (2009), p. 17.
- [Geo+09] Elisabeth Georgii, Sabine Dietmann, Takeaki Uno, Philipp Pagel, and Koji Tsuda. “Enumeration of condition-dependent dense modules in protein interaction networks”. In: *Bioinformatics* 25.7 (2009), pp. 933–940.
- [Ito9] Hiro Ito and Kazuo Iwama. “Enumeration of isolated cliques and pseudo-cliques”. In: *ACM Transactions on Algorithms (TALG)* 5.4 (2009), p. 40.
- [Pu+09] Shuye Pu, Jessica Wong, Brian Turner, Emerson Cho, and Shoshana J Wodak. “Up-to-date catalogues of yeast protein complexes”. In: *Nucleic acids research* 37.3 (2009), pp. 825–831.

- [Sch+09] Matthew C Schmidt, Nagiza F Samatova, Kevin Thomas, and Byung-Hoon Park. “A scalable, parallel algorithm for maximal clique enumeration”. In: *Journal of Parallel and Distributed Computing* 69.4 (2009), pp. 417–428.
- [Vat09] Andrea Vattani. “K-means Requires Exponentially Many Iterations Even in the Plane”. In: *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*. SCG ’09. Aarhus, Denmark: ACM, 2009, pp. 324–332. ISBN: 978-1-60558-501-7.
- [ABL10] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. “Link communities reveal multiscale complexity in networks”. In: *Nature* 466.7307 (2010), pp. 761–764.
- [For10] Santo Fortunato. “Community detection in graphs”. In: *Physics reports* 486.3 (2010), pp. 75–174.
- [Gre10] Steve Gregory. “Finding overlapping communities in networks by label propagation”. In: *New Journal of Physics* 12.10 (2010), p. 103018.
- [Lee+10] Conrad Lee, Fergal Reid, Aaron McDaid, and Neil Hurley. “Detecting highly overlapping community structure by greedy clique expansion”. In: *In Proc. of the 4th Workshop on Social Network Mining and Analysis held in Conjunction with the International Conference on Knowledge Discovery and Data Mining (SNA/KDD’10)* (2010), pp. 33–42.
- [AB11] Faris Alqadah and Raj Bhatnagar. “Similarity measures in formal concept analysis”. In: *Annals of Mathematics and Artificial Intelligence* 61.3 (2011), pp. 245–256.
- [BBH11] Balabhaskar Balasundaram, Sergiy Butenko, and Illya V Hicks. “Clique relaxations in social network analysis: The maximum k-plex problem”. In: *Operations Research* 59.1 (2011), pp. 133–142.
- [Cai+11] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. “Graph regularized nonnegative matrix factorization for data representation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011), pp. 1548–1560.
- [ES11] David Eppstein and Darren Strash. “Listing all maximal cliques in large sparse real-world graphs”. In: *Experimental Algorithms* (2011), pp. 364–375.
- [JS11] Björn H Junker and Falk Schreiber. *Analysis of biological networks*. Vol. 2. John Wiley & Sons, 2011.

- [Lan+11] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato. “Finding statistically significant communities in networks”. In: *PloS one* 6.4 (2011), e18961.
- [LBF11] Augustin Lefevre, Francis Bach, and Cédric Févotte. “Online algorithms for nonnegative matrix factorization with the Itakura-Saito divergence”. In: *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2011 IEEE Workshop on*. IEEE. 2011, pp. 313–316.
- [SC11] John Scott and Peter J Carrington. *The SAGE handbook of social network analysis*. SAGE publications, 2011.
- [BBV12] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. “Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription”. In: *Proceedings of the 29th International Conference on International Conference on Machine Learning*. Omnipress. 2012, pp. 1881–1888.
- [GW12] Bernhard Ganter and Rudolf Wille. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.
- [MH12] Benjamin McClosky and Illya V Hicks. “Combinatorial algorithms for the maximum k-plex problem”. In: *Journal of combinatorial optimization* 23.1 (2012), pp. 29–49.
- [OHT12] Yoshiaki Okubo, Makoto Haraguchi, and Etsuji Tomita. “Structural change pattern mining based on constrained maximal k-plex search”. In: *International Conference on Discovery Science*. Springer. 2012, pp. 284–298.
- [PYB12] Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. “Clique relaxation models in social network analysis”. In: *Handbook of Optimization in Complex Networks* (2012), pp. 143–162.
- [Xu+12] Biao Xu, Ruairí de Fréin, Eric Robson, and Mícheál Ó Foghlú. “Distributed formal concept analysis algorithms based on an iterative mapreduce framework”. In: *International Conference on Formal Concept Analysis*. Springer. 2012, pp. 292–308.
- [Bez13] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.
- [Mik+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.

- [OHT13] Yoshiaki Okubo, Makoto Haraguchi, and Etsuji Tomita. “Relational change pattern mining based on modularity difference”. In: *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*. Springer. 2013, pp. 187–198.
- [Sch13] Helmut Schmid. “Probabilistic part-of-speech tagging using decision trees”. In: *New methods in language processing*. 2013, p. 154.
- [Zha+13] Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo, Kiyota Hashimoto, and Sachio Hirokawa. “Shifting concepts to their associative concepts via bridges”. In: *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer. 2013, pp. 586–600.
- [Zou+13] Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. “Bilingual word embeddings for phrase-based machine translation”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 1393–1398.
- [LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. <http://snap.stanford.edu/data>. June 2014.
- [SIL14] Noa Slater, Royi Itzhack, and Yoram Louzoun. “Mid size cliques are more common in real world networks than triangles”. In: *Network Science* 2.3 (2014), pp. 387–402.
- [Wan+14] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. “Knowledge Graph Embedding by Translating on Hyperplanes.” In: *AAAI*. 2014, pp. 1112–1119.
- [BCK15] Devora Berlowitz, Sara Cohen, and Benny Kimelfeld. “Efficient enumeration of maximal k-plexes”. In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM. 2015, pp. 431–444.
- [DP15] Nicoletta Del Buono and Gianvito Pio. “Non-negative Matrix Tri-Factorization for co-clustering: An analysis of the block matrix”. In: *Information Sciences* 301 (2015), pp. 13–26.
- [IFO15] F.O. Isinkaye, Y.O. Folajimi, and B.A. Ojokoh. “Recommendation systems: Principles, methods and evaluation”. In: *Egyptian Informatics Journal* 16.3 (2015), pp. 261–273. ISSN: 1110-8665.
- [LHW15] Jonathan Le Roux, John R Hershey, and Felix Weninger. “Deep NMF for speech separation”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE. 2015, pp. 66–70.

- [Lin+15] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. “Learning Entity and Relation Embeddings for Knowledge Graph Completion.” In: *AAAI*. 2015, pp. 2181–2187.
- [PCS15] Yulong Pei, Nilanjan Chakraborty, and Katia Sycara. “Nonnegative Matrix Tri-Factorization with Graph Regularization for Community Detection in Social Networks”. In: *IJCAI’15 Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 2083–2089.
- [VM15] Ivan Vulic and Marie-Francine Moens. “Bilingual word embeddings from non-parallel document-aligned data applied to bilingual lexicon induction”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL 2015)*. ACL. 2015, pp. 719–725.
- [YL15] Jaewon Yang and Jure Leskovec. “Defining and evaluating network communities based on ground-truth”. In: *Knowledge and Information Systems* 42.1 (2015), pp. 181–213.
- [Zha+15] Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo, and Etsuji Tomita. “Enumerating maximal clique sets with pseudo-clique constraint”. In: *International Conference on Discovery Science*. Springer. 2015, pp. 324–339.
- [Cao+16] Buwen Cao, Jiawei Luo, Cheng Liang, and Shulin Wang. “Detecting overlapping protein complexes in weighted protein-protein interaction networks using pseudo-clique extension based on fuzzy relation”. In: *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE. 2016, pp. 1244–1252.
- [OHT16] Yoshiaki Okubo, Makoto Haraguchi, and Etsuji Tomita. “Enumerating maximal isolated cliques based on vertex-dependent connection lower bound”. In: *Machine Learning and Data Mining in Pattern Recognition*. Springer, 2016, pp. 569–583.
- [Zha+16] Hongjie Zhai, Makoto Haraguchi, Yoshiaki Okubo, and Etsuji Tomita. “A fast and complete enumeration of pseudo-cliques for large graphs”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2016, pp. 423–435.
- [Con+17] Alessio Conte, Donatella Firmani, Caterina Mordente, Maurizio Patrignani, and Riccardo Torlone. “Fast Enumeration of Large k-Plexes”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’17. Halifax, NS, Canada: ACM, 2017, pp. 115–124. ISBN: 978-1-4503-4887-4.

THIS THESIS WAS TYPESET using L^AT_EX, originally developed by Leslie Lamport and based on Donald Knuth's T_EX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. A template that can be used to format a PhD thesis with this look and feel has been released under the permissive MIT (X11) license, and can be found online at github.com/suchow/Dissertate or from its author, Jordan Suchow, at suchow@post.harvard.edu. The template used in this dissertation is created based on Harvard Template. The logo of Hokkaido University is used under the [University Logo Guideline](#).