Faculty of Engineering and Information Technology
University of Technology Sydney

Friend Recommendation in Social Multimedia Networks

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Shangrong Huang

June 2017

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by myself. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgments

Foremost, I would like to express my sincere gratitude to my supervisors Associate Prof. Jian Zhang for continued support of my PhD study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance has helped me in all aspects of research and writing this thesis. I could not have imagined having a better advisor and mentor for my PhD study.

I also would like to appreciate Associate Prof. Qiang Wu for providing me with continuous support throughout my PhD study and research.

Associate Prof. Lei Wang at University of Wollongong, Dr. Xiansheng Hua from Alibaba Group China, Dr. Tao Mei from Microsoft Asia, Dr. Shiyang Lu at NICTA Australia, and Dr. Xinwang Liu have also given me a lot of academic suggestions and supports. Without their professional guidance and persistent help, this thesis would not have been possible.

I thank my fellows in Global Big Data Technologies Centre: Jingsong Xu, Yucheng Wang, Hao Cheng for the stimulating discussions, and for all the fun we have had in the last several years.

I thank Dr. Jing Ren for her support during my PhD study.

Last but not the least, I would like to thank my parents, for their unconditional support, both financially and emotionally throughout my entire PhD journey.

Shangrong Huang
June 2017 @ UTS

Contents

List of Figures

List of Tables

List of Publications

Papers Published

- **Shangrong Huang**, Jian Zhang, Dan Schonfeld, Lei Wang, and Xian-Sheng Hua, Two-Stage Friend Recommendation Based on Network Alignment and Series-Expansion of Probabilistic Topic Model, *IEEE Transactions on Mutlimedia*, Volumn 19, Issue 6, pp.1314-1326.

- **Shangrong Huang**, Jian Zhang, Lei Wang, and Xian-Sheng Hua (2016), Social Friend Recommendation Based on Multiple Network Correlation. *IEEE Transactions on Mutlltimedia*, Volumn 18, Issue 2, pp. 287-299.

- **Shangrong Huang**, Jian Zhang, Shiyang Lu, and Xian-Sheng Hua (2015), Social Friend Recommendation Based on Network Correlation and Feature Co-Clustering. *in Proceedings of the 5th ACM International Conference on Multimedia Retrieval* (**ICMR'15**) , pp. 315-322.

- **Shangrong Huang**, Jian Zhang, Xinwang Liu, and Lei Wang (2014), A Method of Discriminative Information Preservation and In-Dimension Distance Minimization for Feature Selection. *in Proceedings of the International Conference on Pattern Recognition* (**ICPR'14**) , pp. 1615-1620

Abstract

With the rapid development of computer science and internet technologies, social media and social network has experienced explosive growth over the last decades. Social websites, such as Flickr, YouTube, and Twitter, have billions of users who share photos, videos and opinions, they also make friends on these websites. On-line friendship is an emerging topic that attracts the attentions from both economists and sociologists. The study of the on-line friendship, on one hand, can help the on-line merchants to find their potential customers, and thus make more precise recommendations; on the other hand, it helps to get a deep understanding of the relationships among different people. However, individuals' on-line friend making behaviour is relatively complex and may be affected by many different factors. For example, an individual might make on-line friends with others because they discuss a hard mathematical problem, or it is possible that he/she makes a friend because they both enjoy a film. The reasons for friend making behaviours are likely to be diverse. Traditional friend recommendations that have been widely applied by Facebook and Twitter are often based on common friends and similar profiles such as having the same hobbies or working on a similar topic, which usually can not make a precise recommendation, due to the complexity of the problem. In this thesis, I, with my collaborators, try to give some solutions of on-line social friend recommendation from several aspects. In general, I contribute more than 85% of this thesis.

One problem for social friend recommendation is that how shall we find the important social features that would highly influence individuals' friend making behaviours. Usually, the reason an individual A would make friends with another person B is not that A is satisfied with all the characteristics of B, but that he/she has interest in some factors that B has illustrated. These factors can be viewed as instructive social features for friend recommendation tasks. So in this thesis, we first discuss the important social features for friend recommendation.

Chapter 3 provides a general algorithm of important feature selection that can be applied in different fields such as biological and face image classification. The idea is to project the high dimensional data into lower dimensional space and select the important features that preserve both the global and local similarity structures of the datasets.

Chapter 4 extends the basic idea of Chapter 3 to the field of social networks, and consider the friend recommendation task from the view of the network structure. First we consider the tag features. The important tag features are chosen so that the Flickr tag similarity network looks similar to the Flickr contact network. In other words, Flickr tag similarity network is aligned to the contact network by selecting the important tag features. This network alignment method can also be applied to more than one networks.

In Chapter 5 we begin to take the image features into consideration. It would be relatively difficult to analyse the multi-domain data simultaneously. In this thesis we design a multi-stage scenario to consider the information from one domain in one stage. In this way, not only the complexity of the problem is reduced, but we can also make a deep analysis about the contributions of the information from different domains. For the algorithm proposed in Chapter 5, for the first stage we utilise the tag information similarly as the method suggested in Chapter 4, for the second stage we propose a co-clustering method that clusters the contact information, tag and image feature information simultaneously to refine the final recommendation result.

To further improve the recommendation accuracy, in Chapter 6 we apply a topic model based method in the second stage, instead of the co-clustering method proposed in Chapter 5. The reason for the improvement is that co-clustering method can not provide a precise rank of the recommendation list, but the topic model can give a quantitative analysis of the friendship between two individuals. In this chapter we also provide a new method to find the solution of the topic model, which is different from the widely applied Gibbs sampling, variational inference or the matrix factorization method. The idea is to analytically express the solution of the integral of two random variables, in a series form. In this way we can determine the solution of the probabilistic model precisely, which is better than the traditional Gibbs sampling, variational inference or matrix factorization methods.

In Chapter 7, with the help of widely discussed Deep Learning (DL) Framework, we develop a staged DL-based friend recommendation method. In the first stage, the text and image information is correlated to learn some features via convlutional neural network. In the second stage, the features are refined by the users' clustering information via another deep neural network.

The methods mentioned in Chapter 4, 5, 6 and 7 are applied in a dataset that collected from the widely used image sharing website Flickr. It contains tens of thousands of users, hundreds of thousands tags and millions of images to predict the on-line friendship between users. The performance of these recommendation methods is examined by precision, recall and F-measure. These methods give some insightful knowledge about individuals' online relationship and we hope these methods can help social websites to design their recommendation algorithms.

Chapter 1

Introduction

1.1 Background and Motivation

The online social relationship is a recently emerging topic with the rapid development of the social media. Online social communication platform such as Facebook[1], Twitter[2], and Wechat[3] give good support for individuals to share experiences, images and videos as well as to communicate with friends. Multimedia platform such as Youtube[4], Flickr[5], and Instagram[6] are also providing more convenient methods for the interactions between users. There are plenty of commercial opportunities when taking the online friendships into consideration: an individual might recommend some good products to his/her friend circle, and it is relatively easier for an individual to try some new products, with his/her friends' recommendations. So it is quite meaningful to study further the friendship in the online social platform.

Individuals' online social friendship is to some extent different from the traditional offline friendships. Traditional friends, on one hand, usually have face-to-face interactions frequently because they live near each other, or work

[1] www.facebook.com
[2] www.twitter.com
[3] www.wechat.com
[4] www.youtube.com
[5] www.flickr.com
[6] www.instagram.com

or play in the same place. On the other hand, when they have a chat, the topics are usually about things that happen in their surroundings, for example, the delicious food in a new restaurant nearby, or some new styles of clothing sold in the mall. etc. Also, traditional friends often share some similar social characteristics such as age, income, etc. The online friends can break most of these restrictions (age, gender, etc.)and have much wider topics to share: for instance, a European engineer can easily share his experiences and photos about travelling in Australia, with an American university lawyer student who plans to do so soon.

This brings some new problems to sociologists and online merchants: First, how do people find online friends, and how to make friend recommendation efficiently? Second, how to make the profit from the online friendships? (Tao & Rui 2006). In this thesis we concentrate on the first problem. The main topic of this thesis is about the online friend recommendation algorithms.

As mentioned, compared with traditional friend searching methods, the online friend recommendations meet some new problem. Firstly, the searching datasets are much larger: traditional friend circles are often quite limited by physical situations (location, age, etc.), while the range of online friends can be the billions of users on the internet. Secondly, the reasons that two individuals are to be online friends can be quite diverse, compared with traditional friendships.

Though the online friendships have few limitations compared with traditional friendship, the online individuals do follow some rules to find a friend (Carullo, Castiglione & Santis 2014)(Ghorbani & Ganjali 2012). Similar with the traditional friendships, some kinds of physical constrains also exist for online friendship. A simple example is that it is quite meaningless to recommend a young French student to a Korean old man, because they do not understand each other's language. For some time-sensitive situations (for example, online game), it is also not wise to recommend a man who lives in China to one who locates in America, because of the 12-hour time gap between China and America. These rules can help us to narrow the range of friend recommendation.

In our opinion, one of the important social reasons for individuals to surf the internet is that they want to find online friends that share similar personal interests, no matter who they are and where they are (Jebabli, Cherifi, Cherifi & Hamouda 2015)(Ahmed, Rashid, Hasan & Mahmud 2015). People want to find someone to make discussions about certain topics, to bring new ideas about these topics, and to learn something from other parts of the world (Yin, Zhou, Cui, Wang, Zheng & Nguyen 2016)(Pipanmekaporn & kamolsantiroj 2016). For example, a man who has an interest in delicious food, no matter where he is, might make online friends with those who share many photos and comments about the menus and tasting experiences in restaurants from different areas of the world. As a consequence, we think the online friend recommendation should be based on individuals' interests.

Individuals' interests are related deeply to their social environments, social status and social behaviours, etc. (Zhao, Wang, Yu, Liu & Zhang 2013), which we summarise as "social role". Taking social environment for example, a man in mainland China seldom has an interest in horse-riding since it is very expensive there are no places for this activity. But a man in Hongkong might take care of such activity since horse race betting is legal in Hongkong. A man in Australia is possible to have a great interest in horse-riding, because in Australia there are plenty of land for horse and the price is relatively cheap. A further discussion of the relationship between individuals' social role and their interest are made in Chapter 4 . In this thesis, we will mainly discuss the relationships between individuals' interests and online friendships.

Traditional friend recommendation methods that have been widely applied by big social websites such as Twitter and Facebook are generally based on two rules: 1. content similarity 2. mutual friend recommendation (Hasan, Shaon, Marouf, Hasan, Mahmud & Khan 2015*a*) . The second rule means that if two individuals share the same friends, they can be recommended to each other. Or if the two are friends, then one's friends can be recommended to the other. These two rules are simple as well as effective in many situations, but the recommendation precision is often not satisfactory. The reason might come from the following aspects: In the first place, these two rules might lead to a list of many possible friends, and in reality individuals can not make many online friends, so the problem is how to choose further from the list; In the second place, the online friend making behaviour is a very complex task and might be affected by many different physical, psychological or sociological factors. For example, a student of computer science may have some online friends that are good at mathematical skills or programming and make frequently academic discussions with them. On the other hand, he/she may also has some friends that have no interests in academic problems, but share some interests such as tennis or wine with him/her. As a consequence, these simple methods can not give an ideal solution.

As the above example shown, one individual's friends can be made for diverse reasons and he/she can have different friend circles for various reasons. Consequently, to make a precise friend recommendation, we should take the information from different domains into consideration to make a comprehensive recommendation. This makes the whole friend recommendation task to be relatively complex. In this thesis, we find a solution to this problem step by step: First we consider the friend recommendation utilising the information from one domain, then we utilise the information from other domains to refine the results. This idea has the following advantages, compared with some methods that take the cross-domain information into consideration simultaneously: 1. the whole problem becomes simpler to deal with. 2. the contribution of each domain in the task of friend recommendation becomes much clearer and easier for further analysis.

In this thesis, we will make an analysis of a recommendation system utilising the information from three different domains: the text information, the image information, and the friendship information itself.

Text is an essential and widely analysed source for recommendation problems (Wang, Liao, Cao, Qi & Wang 2015). From the text that individual posts, we can to some extent infer the personal interest of that person, and thus make friend recommendation to the ones that have similar interests. There are also some problems for text-based friend recommendation. One of the main problems is that texts often contain much noise and redundancy that are helpless for friendship. Another problem is that the text similarity does not automatically lead to a friendship between individuals. Chapter 3 and Chapter 4 of this thesis deal with this issue by important feature selection.

Image has ample information showing individuals' interest. However, the information the image provides is vaguer compared with text information (Yao, Ngo & Mei 2011*a*). For example, we can roughly say that the young children like pictures with vivid colours, and the traveller prefers photos with nature scenery. But these arguments are not hundred percent certain and there might exist many exceptional cases. For the above reasons, in this thesis, we utilise the image as additional information to improve the performance of the friend recommendation. Chapter 5 and Chapter 6 of the thesis give more details of the refinement methods.

1.2 Research Issues Summary

Based on the above discussions, we present the following important research issues in this thesis that may lead to good friend recommendation performance. We first study the pure text-based recommendation, and then extend it to multimedia recommendation.

1.2.1 Text-Based Friend Recommendation

In the first step, we consider the text information. As mentioned previously, we apply the feature selection method for text-based friend recommendation. In this approach, words are considered as features, and the goal is to find the useful words that are important for individuals' friend making behaviours.

We explain the idea in details as follows: when an individual intends to find some online friends, he/she is not likely, and in fact impossible, to have a thorough search over the internet, nor will he/she make a deep and complete investigation of another person online to make the decision. It is typical that if an individual finds some interesting posts online that he/she wants to get more knowledge about it(text, image, video, etc.), he/she might contact the authors and make further discussions, and an online friendship might start from these discussions. So when people make online friends, they will not take all the aspects into consideration, but only concentrate on some important aspects that interest them.

So it is important to find a good algorithm to select important features according to specific requirements. We first propose a general feature selection method for traditional tasks such as classification and clustering in Chapter 3. Then we will extend it to the social network for friend recommendation in Chapter 4.

A General Feature Selection Method: Discriminative Information Preservation and In-Dimension Distance Minimization (DIP-IDM)

We first propose a general feature selection method: DIP-IDM, for traditional tasks such as classification and clustering. If we have a dataset contains plenty of items and the number of features in each item is large (For example, a large text or image collections, or a gene pool, etc.), and the task is to classify or cluster the dataset. A large number of features not only increase the complexity of the classification/clustering task, but also it is common that most of the dimensions of the data are redundant or even have negative effects for a correct classification/clustering. Only some features in the feature space, or some combinations of different features are helpful and the similarity of these features or feature combinations can lead to a good classification/clustering result (Ding, Zhu, Tang, Lin, Xiao & Dong 2016*a*). So feature selection and feature extraction are two general methods for many different machine learning problems. (Shah & Patel 2016)

In general, we should select/extract the features that have the following property: If two items are in one class, then their similarity on these selected features should be high, otherwise it should be low. This property is defined as “ discriminative information preservation” in Chapter 3. Based on this property, we design an optimization problem similar as (Zhou, Liu, Zhu, Liu & Yin 2014).

On the other hand, if the selected/extracted features classify/cluster the training data perfectly, it might lead to an over-fitting problems that though the selected features works well on the training datasets, it might fail on the testing datasets (Eleftheriadis, Rudovic & Pantic 2016). To overcome this problem, we add a term in the optimization problem. This term ensures that before and after the feature selection, the similarity between two nodes in the dataset does not change much. This can be viewed as a method that minimises the in-dimension distance, which leads to a better performance.

With the idea of global and local structure preservation we propose a feature selection algorithm in Chapter 3, which illustrates good performance on some widely-used image and biological datasets. Then we extend the idea to social friend recommendation.

Network Correlation Based Social Friend Recommendation (NC-Based SFR)

In the classification/clustering tasks, the aim is to put the items that are similar to the same classes/groups. In friend recommendation, the aim is to find the pairs that have similar interest. We apply the idea of general feature selection method DIP-IDM to select text features.

For text-based friend recommendation, two assumptions can be made as the basis of friend recommendation. In the first place, some important words are more helpful than other words in the task of finding friends. For example, when a traveller posts an article about a nature scenery, the readers might be attracted by some of his/her specific descriptions such as “red maple leaves”, “big golden fish”, “rainbow”, etc., and may require photos, videos or detailed travel diaries for a more thorough and deeper understanding. But the readers are not likely to have an interest in some words such as “beautiful flowers” and “blue mountains” because they are relatively common. In the second place, an individual is likely to choose friends that post similar texts. For example, a professional cook who posts many articles and photos about food might choose online friends who often share their local traditional dishes, so that he/she has more chances to develop new menus from the discussions with these friends. So the problem is, how to choose the important text features (words) for friend recommendation?

Feature selection methods are previously proposed as an approach for recommendation systems (Sivakumar, Balaganesh & Muneeswaran 2015). In this thesis, we get the approach of this feature selection problem from a network view.

If we define one individual as a node and some kinds of relationships between two individuals as edges, then we have many nodes and edges to form a network (Pan & Lin 2011). If the edges stand for the friendships between individuals, then we have a friendship network; if the edges stand for that two individuals buy the same product, then we have a co-shopping network; Or if the edges stand for that two individuals post similar texts, then we have a text similarity network. In this way, we can have the same nodes to form different social networks. These different networks, whose edges are defined by the data from different domains, have the same nodes but different edges, or topologies. For example, two individuals travelling in the same city might buy similar local specialties, so they have an edge in the co-shopping network; but the two visit different attractions in the city: one prefers to stay in the historical museum, and the other enjoys the modern Disneyland, so they do not have an edge in the co-visiting network.

The main reason we deal with the social feature selection problem from a network view is that the friendships between individuals can naturally form a network, and the fundamental conceptions of the network, such as the distance between nodes, can be easily defined and adjusted when we choose different social features. In this way, the concept of the network helps us to have a deep understanding of the (Online) social relationships.

As mentioned, different networks have different topologies. On the other hand, these different networks are not entirely independent from each other. This is because the data from different domains are usually related to each other. For example, if two individuals have the similar hobbies of jogging, then there is a higher probability that both of them prefer soft, comfortable, durable shoes. This tells us that with the known topology of one network we can to some extent infer the topologies of other networks.

Based on this investigation, we design a network alignment method to correlate different networks. Network alignment here can be defined as to map one network to another with some constraints/rules (Sun, Yang, Liao, Xu & Luo 2015). For text-based friend recommendation, we align the text-similarity network to the friendship network by choosing some feature words, so the topology of the modified text-similarity network is similar with to friendship network.

Consequently, these chosen text words are considered to be more instructive in the friend recommendation task. By comparing the text-similarity on these important text word, we can make more precise friend recommendation.

1.2.2 Staged Recommendation with Co-Clustering (SRCC)

Till now we only consider the text information for the recommendation task. The humans' friend making behaviours are in fact very sophisticated and more factors should be considered. There is ample multimedia information on the Internet and it also helps us to find the right people as friends. In Chapter 5 we take the image information into consideration.

Individuals that prefer similar images are likely to have some similar interests and common things, and thus have higher possibilities to be online friends. As a consequence, the image information is to some extent instructive for friend recommendation. For example, history lovers prefer photos about ancient sculptures, temples and old paintings, and children may prefer photos that are vivid and colourful rather than the old paintings. On the other hand, image information is quite vague and noisy for friend recommendation, and the relationship between image and friendship is a challenging topic in psychology (Geng, Zhang, Bian & Chua 2015). For example, it is difficult to tell how possible the two individuals to be friends, if they both enjoy a picture of a young, beautiful lady. As a consequence, in Chapter 5 we design a staged recommendation method and introduce the image information as an additional materials to refine the results of the text-based recommendation.

Specifically, in Chapter 5 we apply a two-stage method for friend recommendation. Compared with the methods that combine the knowledge from different domains in one step (Min, Bao, Xu & Hossain 2015), there are two advantages of applying the knowledge from one domain in each stage: Firstly, the complexity of the system is reduced. Secondly, the staged method can give a clearer explanation of how the knowledge in each stage contributes to the final recommendation result, as well as an in-depth understanding of the different knowledge for friend recommendation in each stage.

The procedure goes as follows. In the first stage we apply directly the method introduced in Chapter 4 to generate a possible friend list.

Before the second stage, we apply a friend circle enlargement step. That is, for all the individuals in the possible friend list that has been generated in the first stage, we also add their friends into the possible friend list. The reason for this enlargement is two folds: firstly, it is a common rule that friends' friends are more likely also to become friends; secondly, the text-based method might filter some friends that are made due to other factors.

After the enlargement step a relatively long possible friend list is obtained. Then, information from three domains are concerned: individuals' friendship information, their text information, and their image information. To utilise the information from three different domains, intuitively we adopt a three-way co-clustering method which clusters the information from different domains simultaneously (Wang, Lin & Yu 2016). The reason we apply a co-clustering based approach is not only that it is conceptually simple, but also because that it is an efficient way to combine the knowledge from three different domains. Also some group information of individuals, texts and images can be obtained through co-clustering. In this way we can refine the recommendation precision further, compared with the first stage performance.

1.2.3 Probabilistic Topic Model with a Series Expansion Solution (PTM-SE)

The co-clustering method provides a conceptually clear solution of friend recommendation task. However, it lacks the ability to determine exactly how close two individuals are. It only tells if they belong to the same cluster of not. So our next task is to find a way to determine the friendships mathematically and more precisely.

Probabilistic topic model (Blei 2012*a*) provides the ability that we require. The basic idea of the probabilistic topic model is that the data we observe are generated from some random variables, and the random variables follow some probabilistic distributions and the distributions are controlled by some latent parameters. Based on the Bayes' rule (Gelman, Carlin, Stern & Rubin 2003), the value of the latent parameters can be inferred from the observed data, and the value of random variables can be determined from the distributions.

Taking a simple example, an article can be viewed as a combination of several latent topics such as "history", "travel", "landscape", etc. Each topic can be regarded as a combination of words: For the topic "history", it may consists of following words "Rome, 50%", "tomb, 30%", and "Egypt, 40%". The percent that appears after each word indicates the probability of the word belonging to the topic. In this way, an article can be calculated that the possibility it belongs to the topic "history" is 30%, "travel" is 20%, and "landscape" is 50%. With this kind of topic summary it is easy to do the further text classification or retrieval task by comparing the similarity on topics (Blei, Ng & Jordan 2003*a*).

The main advantage of the probabilistic topic model is that it provides a generative model to express the relative complex relationships of data from different domains (Min et al. 2015). In our case, the probability expression of data can give a more accurate method to illustrate the closeness between individuals.

One issue for the probabilistic topic model is how to find its solution, or, specifically, how to determine the value of the parameters of the latent random variables. One critical step to solve a topic model involves evaluating the integral of the complex combination of different random variables. This is usually a difficult task and mathematicians can not give a general solution today. To approximate it, the widely used methods include Gibbs sampling (Chen & Li 1995), variational inference (Blei et al. 2003*a*), and matrix factorization (Sun & Luo 2016). All of these methods have some disadvantages. Gibbs sampling method samples the distribution from the real data to provide relatively precise approximations, but it requires large computational resources both in time and memory if the dataset is large. Variational inference requires less computational resources but the accuracy of its approximation can not be ensured. It tries to approximate the real distribution of the data with some typical simple distributions, which can not guarantee the precision. Matrix factorization methods, which have relatively simple expressions, usually apply some gradient-descent based methods to find the solution, and is relatively easy to be trapped into a local optimum.

In Chapter 6 of this thesis we develop a novel method to evaluate the integral of the correlation of different random variables. The essential idea is that the integral can be expressed in the form of a series, and accuracy and complexity of the solution can be traded off by controlling the length of the series. The study of the series expansion for a complicated integral can be traced back in the 1960s (Springer 1979)(Pruett 1972), when the computer hardware was not well developed, and the numerical evaluation methods are not intensely discussed. Based on the contributions of the researchers in recent years, we find a series expression as the solution of the integral, with which we can determine the values of the parameters in the probabilistic topic model.

The probabilistic topic model is applied in the second stage of the system as a substitution of the co-clustering method. It gives a numerical expression for the degree of intimacy between two individuals, which the co-clustering method can not provide. Consequently, this approach can recommend friends more accurately.

1.2.4 Social Friend Recommendation via Deep Learning Framework (DL-SFR)

The deep learning framework, which has been proposed in the first decade of 21st century, has attracted much attention from different fields and has made significant achievements in image recognition (Ciresan, Meier & Schmidhuber 2012), natural language processing (Sarikaya, Hinton & Deoras 2014), bioinformatics (Chicco, Sadowski & Baldi 2014), and recommendation systems (Oord, Dieleman & Schrauwen 2013), etc. Before the proposal of the deep learning architecture in 2006 (Hinton & Salakhutdinov 2006), most of the machine learning methods are of shallow architectures, which limits their representation learning capacity, and these methods have difficulties in modelling complicated data such as texts, images and videos. The reason for the shallow architecture is due to the difficulties in the training process. In (Hinton, Osindero & Teh 2006), it shows that with a layer-wise training strategy, the complexity of training a deep architecture can be greatly reduced and the prediction performance can be greatly improved. It provides a way to extract much more effective features for different tasks.

Researchers have introduced the deep learning framework into the tasks of both collaborative filtering (Wang, Wang & Yeung 2015) and content based filtering (van den Oord, Dieleman & Schrauwen 2013). On the other hand, as far as we know, there are no deep-learning-based frameworks that are developed for social friendship discovery. In Chapter 7 of this thesis, we propose a new deep-learning-based feature extraction method that aims to predict the friendships between individuals. The method contains two stages. In the first stage, a traditional convolutional neural network is applied and the features are extracted from the last several layers of the network. In the second stage, the extracted features from the first stage are fed into a new neural network. Also, the individuals are assigned to different communities via some community detection method. The output of the new neural network is the community label. In this way, the features that are useful for building social relationships are extracted for further friend recommendation.

1.3 Contributions and Thesis structure

1.3.1 Research Contributions Highlight

This thesis mainly provides the following academic contributions.

- propose a general feature selection method that can be applied in different fields, by considering preserving the global and local structure of the data. (Chapter 3)

- develop an accurate friend recommendation algorithm that can be used to automatically find online friends , based on the alignment of different kinds of social networks. (Chapter 4)

- build a two stage friend recommendation framework to both utilise the information from different domains, as well as to reduce the complexity (Chapter 5 and Chapter 6)

- present a three-way co-clustering method to make a combinational recommendation based on image and text data (Chapter 5)

- introduce a novel series-expansion method to determine the solution of a complex integral, in order to find the solution of a probabilistic topic model, and make more precise recommendations. (Chapter 6)

- apply a staged deep learning based framework for precise friend recommendation. (Chapter 7)

- make comprehensive experiments to evaluate the performance of all the proposed methods, compared with different state-of-the-art friend recommendation systems. The self-collected Flickr datasets are used as the main evaluation dataset(Chapter 4, 5, 6 and 7)

1.3.2 Thesis Structure

Figure 1.1 illustrates the profile of the works in this thesis. The rest of the thesis is organised as follows.

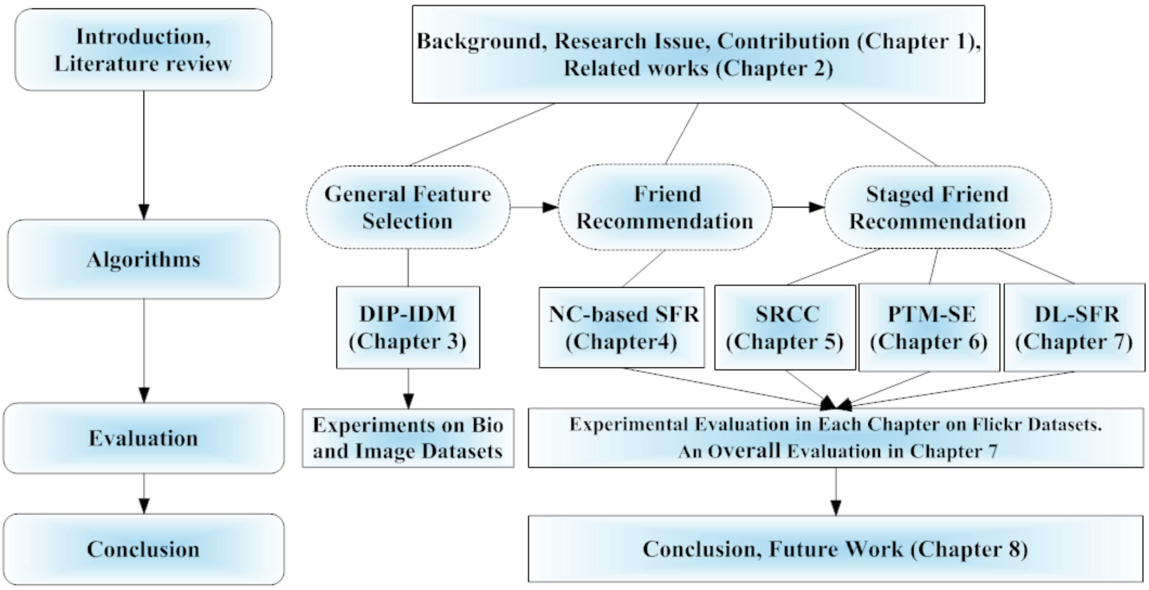


Figure 1.1: The profile of works in this thesis

In Chapter 2 we comprehensively review the different research fields that are related to this thesis. We mainly review the following topics:

- Recommender system, friend recommendation methods and cross domain recommendation
- Feature selection technologies
- Probabilistic topic model and some basis of algebra and random variables.

Chapter 3 gives the detailed explanation of a general feature selection method based on global and local structure preservation. The method is evaluated on different bio and image datasets.

Chapter 4 extends the feature selection idea in Chapter 3 to the social network. By aligning the text network to the friendship network, the important text features are selected as the instructive features for friend recommendation.

By introducing a second stage on the basis of Chapter 4, in Chapter 5 a co-clustering method is applied to refine the result of friend recommendation. Image features are utilised in the second stage to provide a more accurate recommendation.

To further increase the recommendation precision, a probabilistic topic model is applied in Chapter 6 to calculate the intimacy between individuals. We propose a novel series expansion method to find the solution of the integral of twisted random variables. This method has better performance compared with tradition Gibbs sample or variational inference. We get a further improvement in the recommendation accuracy.

In Chapter 7, a new friend recommendation method based on deep learning framework is proposed. This chapter tries to extract features that are highly related to individuals' friend making behaviours. A summary of the comprehensive experimental results of the previously proposed methods is also given in this chapter.

Chapter 8 concludes the thesis and points some directions for our future research.

Chapter 2

Literature Review

This chapter reviews the related works, in regarding with all the general important fields related to this thesis. For some works that relate to a particular chapter of this thesis, please check the specific related work part in that chapter. The literature related to general recommender system and friend recommendation is listed in Section 2.1. In Section 2.2 we review the previous works about feature selection and extraction. Some deep learning based feature extraction methods are also presented in this section. The last section presents the studies related to the field of the probabilistic topic model.

2.1 Recommendation System

In the beginning, we review the literature about the main topic of this thesis: recommendation system. Firstly we go through the researches about the general recommendation. Then the recent works about friend recommendation are mentioned. In the following, cross-domain recommendation is discussed since in the later chapters of this thesis the recommendation between different domains becomes an important topic. In the end, the applications of the deep learning framework in the recommendation tasks are listed.

2.1.1 General Recommendation System

Recommendation system can be viewed as an information filtering system that predicts the "rating" or "preference" that a user would give to an item/product/person (Ricci, Rokach, Shapira & Kantor 2010). With the rapid development of Internet social media and E-commercial activities, it has been widely used in this decade. It helps individuals to find their favourites without others' suggestions, and it has been applied in different areas: items (Ma, Yang, Lyu & King 2008*a*), movies (Hu, Wang, Wu, Guo & Zhang 2010), music (Dolatkia & Azimzadeh 2016), videos (Roy, Mei, Zeng & Li 2012), travel and tourism (Chen, Cheng & Hsu 2013)(Borrís, Moreno & Valls 2014), jobs (Siting, Wenxing, Ning & Fan 2012), experts (Chen, II & Giles 2015), and community (Yao, Ngo & Mei 2011*b*) or friend recommendation (Hasan, Shaon, Marouf, Hasan, Mahmud & Khan 2015*b*), etc.

There are two main methods to provide a list of recommendations: collaborative filtering (CF) (Breese, Heckerman & Kadie 1998) and contend-based filtering (CBF) (Brusilovsky, Kobsa & Nejdl 2007). The CF method builds a model from a user's past behaviour(purchases or ratings), the model is then compared with other users. A group of users who have similar models are chosen, and the recommendation is made based on the behaviours of the users in this group. CBF method, on the other hand, utilises a series of discrete characteristics of an item to recommend additional items with similar properties. The recommendation system problem is a developing research topic and all the existing methods have some drawbacks that need to be studied further. For example, for CF based method, it requires a large amount of information on a user for accurate recommendation, which is usually called "cold start" problem (Schein, Popescul, Ungar & Pennock 2002). For the CBF based method, though little information is required to get started, its scope can be quite limited and can not filter items on some assessment of quality, style or viewpoint.

In some cases, other kinds of recommendation methods can also be applied. One is called demographic filtering (Pazzani 1999), which follows the principle that individuals with certain common personal attributes (age, sex, position, etc.) are likely to have the common preference. Another method, Knowledge-based recommendation (Carrer-Neto, Hernández-Alcaraz, Valencia-García & García-Sánchez 2012) is also applied when the explicit knowledge about the item assortment, user preference or the recommendation criteria are available. It is obvious that these methods can not be applied when there are knowledge acquisition bottlenecks. As all the above methods have their advantages and disadvantages, hybrid methods are also proposed to achieve better recommendation performance (Bobadilla, Ortega, Hernando & GutiéRrez 2013*a*)(Isinkaye, Folajimi & Ojokoh 2015). It usually combines different methods based on some bio-inspired or probabilistic method. (Bobadilla, Ortega, Hernando & GutiéRrez 2013*b*)

It would be quite tedious to give a thorough review of the current research on the recommendation system in this thesis. Some useful surveys have been presented recently (Bobadilla et al. 2013*b*)(Su & Khoshgoftaar 2009). In our study, we are particularly interested in individuals' friendship on the internet and the related recommendation methods: how can two people attract each other online, and in what situations will they become online friends. This topic attracts much research interest recently, for it can widely improve the user experience and enhance user stickiness (Ding, Zhu, Tang, Lin, Xiao & Dong 2016*b*). In the following, some important or recent researches about friend recommendation are reviewed.

2.1.2 Friend Recommendation

With the development of online community, individuals are no more satisfied with being friends only with their schoolmates, colleagues or neighbours (Ding et al. 2016*b*), but want to find some online friends to share their interests and experiences. Studying the online friendship helps us to have a deeper understanding of the social network and individuals' online behaviours (Verma & Pal 2015). Also, there are great commercial opportunities in discovering the online friendships. (Liang, Ho, Li & Turban 2011) gives a good report of the correlation between the online friendship and e-commercial activities: the friendship can positively influence the user's intention to use social commerce.

Big social websites have designed some relatively simple friend recommendation strategies to strength individuals' online friendships (Sivakumar et al. 2015): Facebook explores users' communication interaction to give the possible knowing person, Weibo[1] finds the most likely friends who are not already associated, according to some content similarity measurements. This problem also has been studied by researchers. Different kinds of data are applied for online friend recommendation or community detection. Some early works mine the location similarity of different individuals (Li, Zheng, Xie, Chen, Liu & Ma 2008), whose applications are quite limited and the performance in general cases is unsatisfactory. (Wu, Jiang & Huang 2009) develops a friend recommendation scenario based on the online appearance of individuals, by assuming that the appearances are somehow attractive for potential friends. (Yao, Ngo & Mei 2011*c*) utilises both the location information and the photo information, and develops an optimisation method to effectively combine the two factors. (Li, Nie, Lee, Giles & Wen 2008) develops a method for community discovery and friend recommendation based on the text information. (He, Li, Fei, Tang & Zhu 2015) also deals with the community detection task with a matrix factorization method by combing the link and content information.

For friend recommendation, different methods and applications has been published: (Hannon, Bennett & Smyth 2010) considers user generated content on Twitter and develops a fast algorithm for real-time user-to-user similarity for follower/followee recommendation. It is developed only for text similarity. (Li & Chen 2009) concentrates on mobile applications and makes friend recommendation considering the influence of different social aspects such as locations and common interesting words. It defines a transition probability for friendship recommendation based on location neighbourhood

[1] www.weibo.com

and interesting word co-occurrence.

No matter what kinds of information is utilised (text, visual, link, location, etc.), one key problem is to find the key features that are helpful in the task of friend recommendation. As mentioned in Chapter 1, some features are more instructive and some features can be viewed as redundancy. In (Shalforoushan & Jalali 2015), the authors extract the important attributes for social link prediction from the social environment via a Bayesian network. (Zhang, Fang, Ng & Zhang 2011) proposes an algorithm that determines the trust factors that makes the recommended friendships to be more solid. In (Kacchi & Deorankar 2016), it recommends friends based on the features that can well represent the individuals' lifestyles. These selected features can be similar interest, similar bloody group, etc. In Section 2.2 some important aspects of feature selection are to be reviewed and in some later chapters of this thesis, we will develop some feature selection/extraction methods for friend recommendation.

2.1.3 Cross-Domain Recommendation

As mentioned in Chapter **??**, many different factors might influence individuals' friend making behaviours. To make precise friend recommendation, the information from different domains should be considered. The problem of general cross domain recommendation has been studied in recent years (Fernández-Tobías, Cantador, Kaminskas & Ricci 2012). It aims to generate or enhance recommendations in a target domain by exploiting knowledge from other domains, and mitigate the cold-start and sparsity problems that frequently occur in the single domain recommendation tasks. (Cantador, Fernández-Tobías, Berkovsky & Cremonesi 2015). According to (Cantador et al. 2015), the cross domain recommendation has two essential functions: the first is to aggregate knowledge, and the second is to link and transfer knowledge for a better recommendation.

For the first function of aggregating knowledge, (Berkovsky, Kuflik & Ricci 2007) proposes a method by normalising users' preference of different domains into the same type and representation, and thus the user preference of different domains can be merged. This method requires a significant amount of user preference records in multiple domains. Except for the users' preference, other data which is related to the recommendation tasks such as neighbourhoods (Tiroshi & Kuflik 2012) and latent features of the probabilistic model for each domain (Low, Agarwal & Smola 2011) can also be merged. Furthermore, the recommendation result of each single domain can also be merged in a well designed weighted manner (Givon & Lavrenko 2009).

For the second function, it links or transfers the knowledge between domains to enhance the useful information in the target domain. (Chung, Sundaram & Srinivasan 2007) utilises the overlapped item attributes in the source and target domain to deign personalised filtering strategy. If the overlapped item attributes are rare, (Fernández-Tobías, Cantador, Kaminskas & Ricci 2011) proposes a framework to extract a multi-domain semantic network, and links items and concepts in the source and target domains. Another way for knowledge transfer is to share latent features in different domains. (Hu, Cao, Xu, Cao, Gu & Zhu 2013) assumes that the users, items and domains share some common latent variables, and applies a user-item-domain tensor factorization to decompose the latent variables. At last, if the individuals' attributes are to some extent difficult to transfer, it is possible to transfer knowledge at a community/group level. In (Li, Yang & Xue 2009), to alleviate the sparsity problem, users and items are first co-clustered in the source domain to form a codebook, and the target domain obtains the knowledge through the codebook.

In this thesis, as mentioned, because the friend making behaviours are quite complex and can be affected by different factors, we design a two-staged algorithm to solve the cross-domain problems. The staged method can to some extent reduce the complexity of the algorithm and check the contributions of the data from different domains. Several recommendation algorithms have adopted the staged-based methods such as (Qinglin, Huifeng, Bo & Minghu 2010) and (Liu, Cao, Liang & Li 2015), which we will give some detailed summaries in the related chapters.

2.2 Feature Selection and Extraction

One important issue in recommendation problems is how to find the instructive features that are helpful for the recommendation task. Feature selection/extraction is itself an important problem in machine learning. In this section we will first review the general feature selection/extraction methods, then the social feature selection problem in the recommendation system, which is most related to this thesis. In the last, we will briefly refer some deep learning related feature extraction methods.

2.2.1 General Feature Selection and Extraction Methods

In the field of machine learning, the dimensions of the practical datasets we want to deal with, such as the text, image, social or biological datasets, are usually very high. They might contain more than tens of thousands dimensions, and many of the dimensions can be judged as noise or redundancy for specific tasks such as classification/clustering/retrieval. So dimensionality reduction is widely applied in the preprocessing of high dimensional data. One of the simplest ways for dimensionality reduction is feature selection: to select important features that contain high-relevant information for particular tasks. Feature extraction, on the other hand, is a more general method to transform the input feature space onto a lower dimensional subspace that preserves most of the relevant information (Khalid, Khalil & Nasreen 2014).

The feature selection/extraction methods can be developed for supervised, semi-supervised and unsupervised tasks. The availability of label information allows supervised feature selection/extraction algorithms to effectively select discriminative features (Li, Chen, Wei, Xu & Kou 2007). If only a small portion of features are labelled, the semi-supervised methods can be applied which take the advantages of both labelled and unlabelled data (Xu, King, Lyu & Jin 2010). In the case that there is an absence of label information, unsupervised feature selection methods are applied and they are considered to be much harder problems (Dy & Brodley 2004). Most chapters of this thesis deal with the supervised cases. For some detailed references related to the unsupervised cases, we list some in the related work part in Chapter 3.

First we make a discussion about feature selection. According to the search strategies, feature selection methods can be distinguished into three categories: filters, wrappers, and embedded methods. Filtering methods, which require relatively light computational load and can to some extent avoid overfitting problems, are to utilise variable ranking techniques as the principal criteria for variable selection by ordering (Stoppiglia, Dreyfus, Dubois & Oussar 2003), and the feature selection procedure is independent of the training process. Wrapper methods, on the other hand, consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations (Kohavi & John 1997). This kind of methods leads to better prediction performance but higher computational costs. The embedded methods reduce the computation time taken up for reclassifying different subsets which are done in wrapper methods. The embedded methods incorporate the feature selection as part of the training process (Wang, Tang & Liu 2015). In Chapter 3, we adopt the filtering methods because of its simplicity and efficiency in dealing with the big data problems.

Different from feature selection, the feature extraction method performs some transformation of the original features to generate new features that are more significant for specific tasks. One classical and widely used method is the Principal Component Analysis (PCA) (Johnson & Wichern 1988). It turns various feature indicators to a small number of effective indicators. In brief, PCA method projects the high-dimensional data to low-dimensional master subspaces that are relevant with several largest eigenvalues of the covariance matrix of the data. If the data is highly non-linear, then a kernel based improvement, kernel PCA can be applied. It maps the input data via a nonlinear transform to the feature space, and then executes the linear PCA in the feature space (Yang, Frangi, Yang, Zhang & Jin 2005). Entropy-based methods, which analyse the uncertainty of features are also proposed (Peng, Zhang, Tian & Zhang 2015). It extracts features with small entropies/uncertainties and obtains good classification results. The combination of the ideas from PCA and information theory leads to some powerful algorithms such as Local Linear Embedding (LLE) (Roweis & Saul 2000), which maintains the local geometric nature while reducing the data dimensionality.

One class of feature extraction methods are the artificial neural network (McCulloch & Pitts 1943). It is based on intellective computing that using computer network system to simulated a biologic neural network, which is a nonlinear and adapting information processing system which contains mass processing units. To find a solution of the network, traditional ANN met some serious computational issues by applying the gradient descent methods in the 80s and 90s of the last century. In recent years, with the development of deep learning technologies, which adopt some layer-wise training methods (Hinton, Deng, Yu, Dahl, r. Mohamed, Jaitly, Senior, Vanhoucke, Nguyen, Sainath & Kingsbury 2012), the neural network has been widely discussed and applied in different fields, and has made significant achievements. Successful deep models contain Deep Belief Network (DBN) (Hinton et al. 2006), Deep Neural Network (DNN) (Hinton et al. 2012), Deep Auto Encoder (DAE) (Hinton & Salakhutdinov 2006), and Convolutional Neural Network (CNN) (Krizhevsky, Sutskever & Hinton 2012*a*). etc. A detailed literature review of deep learning framework related to our work is to be given in Chapter 7.

2.2.2 Social Feature Selection

The initial motivation for social feature selection/extraction is that the social data often contain high dimensional features that are difficult to deal with, and most of the features are redundant except for specific tasks (Bellman 2003). To address this problem, we usually apply feature extraction (Jolliffe 1986) or feature selection (Liu & Mototda 2008) methods. Feature selection is often preferred over extraction, because the selected features have more obvious physical meanings. Later we will discuss the feature extraction methods based on deep learning framework.

(Hatammimi & Sharif 2014) gives an interesting social survey in an Indonesia University about the relationship between individuals' social media features and their behaviour patterns. It illustrates some positive influence of the social media features on certain social behaviours. To select the important social features, (Azadifar & Monadjemi 2015) proposes a graph based method which first builds a social feature graph whose edges stand the correlation between features. Then the features are clustered and the important features are selected from each cluster. This method concentrates on the relationship between features. Different from the above work, (Tang & Liu 2014) utilises the link information to form a user-relation graph, and extracts the pseudo-class label from the graph. Then the important features are selected based on the constraints of the pseudo-class labels. This method can be applied for both supervised and unsupervised cases. In some situations, the labels of social media data are difficult to acquire, so unsupervised selection methods are required in different cases. In (Li, Hu, Tang & Liu 2015), based on the work of (Tang & Liu 2014), the authors develop a method that can update the social features with the change of the social environment, for the unsupervised case. The idea is to design a computational efficient derivative calculation method in accordance with the newly arrived social features.

Different from (Li, Hu, Tang & Liu 2015), in this thesis, we do not take much care about the real-time recommendation, partly because that the friendship usually does not have a strict real-time requirement. (Ding et al. 2016*b*) selects the important social features and treat the friend recommendation problem as a classification problem. The selection is done by removing the redundant features step by step. (Dao, Rangamani, Chin, Nguyen & Tran 2015) presents a dot product representation of the social network, and selects the features that minimise the gap between this representation and the real social network. This method takes the missing data into account.

Sometimes, simply selecting features can not find a good representation of the dataset. In the following, we discuss some methods of social feature extraction, as well as the deep learning framework.

2.2.3 Social Feature Extraction and Deep Learning Framework in Recommendation System

Feature extraction can also be applied to the study of the social network. However, the existing works tend to extract the manually-designed social features. (Thi & Hoang 2013) proposes a link prediction methods by extracting the features that related to their jobs and locations. (Liu, Rui, Zhang & Jia 2016) designs more than ten kinds of features to predict the real social relationship between users (son, lover, friend, teacher, etc.). (Alsaedi & Burnap 2015) applies temporal and textual features for disruptive event identification.

Deep learning framework provides a good way to extract the social features automatically and makes further recommendations. (x. Lv, Yu, y. Tian & Wu 2014) applies a DBN model to extract hundreds of individuals' social attributes such as gender, age, region to predict their behaviours on the market. (Wang, Wang & Yeung 2015) integrates a Bayesian denoise auto encoder and a collaborative topic regression model in a deep learning model for the recommendation task. This can be viewed as a deep collaborative filtering model. For the content-based filtering model, (Oord et al. 2013) introduces a deep CNN model to learn the latent factors from music signals for further music recommendation. (Deng, Huang, Xu, Wu & Wu 2016) proposes a deep autocoder model to factorise the user-item-rating and user-trust matrices to catch the preferences of users. This model helps to alleviate the sparsity and cold-start problems in the recommendation system. This work shows that the social relationship can be extracted via a deep neural network.

In Chapter 7 of this thesis, we will give more detailed discussions about the application of the deep learning in the social network and recommendation systems.

2.3 Probabilistic Topic Model

In Chapter 6, a topic model is applied to get a more precise recommendation. There are many interesting problems and researches about the topic model. Here we will give a review of the basic idea and the state-of-the-art methods and algorithms.

A probabilistic topic model is a type of statistical model that is used to discover the latent topics in data. It is originally developed for finding topics in textual data (Blei 2012*b*). Then it has been applied in the fields of image annotation (Tian, Huang, Guo, Qi, Chen & Huang 2015), audio/video analysis (McCaffery & Maida 2013), and bioinformatics (Chen, Hu, Lim, Shen, Park & Rosen 2012). In text processing, topics can be viewed as semantically related probabilistic clusters of words in text corpora, and the process for finding these topics is called topic modelling. (Daud, Li, Zhou & Muhammad 2010).

In general, a topic model unveils the latent semantic structure (topic) of the data, it is a mechanism for discovering low-dimensional, multi-facet summaries of documents or other discrete data. In a topic model, observable data and latent random variables are represented as nodes, and the relationships between nodes are represented as edges. If there are strong casual relationships between two nodes, then we have directed topic models, otherwise an undirected topic model can be applied (Daud et al. 2010). In this thesis we concentrate on the directed topic model since it can be described as a generative process and thus enjoy modelling and computational benefits (Wang, Crammer & McCallum 2007).

A generative model assumes the observed data is generated from a specific probability distribution. Some directed generative topic models have been proposed and applied successfully in different fields. Probabilistic Latent Semantic Analysis (PLSA) (Hofmann 1999) proposes a statistical latent variable model for co-occurrence data which associates an unobservable class. It provides a generative model at words level but not at documents level. Latent Dirichlet Allocation (LDA) (Blei et al. 2003*a*) makes the improvement by providing a model that is generative both at word and document levels. The essential idea of LDA is that documents are represented as random mixtures over latent topics, where each topic is defined by a distribution over words. It captures the implicit correlation between words via the topic representation. One drawback of PLSA and LDA is that they can not explicitly model the correlation between topics. Correlated Topic Model (CTM) (Blei & Lafferty 2006) uses flexible distribution for topic proportions that allows for considering direct correlation between topics.

In the case of link based relationship (e.g., online friendship, citations, etc.), it is straightforward to assume that if there is an edge between two nodes, they can be topically related. Based on this assumption, link-PLSA-LDA method (Nallapati & Cohen 2008) is proposed to exploit the relationships between nodes, as well as the influence of each node. (Erosheva, Fienberg & Lafferty 2004) develops a joint probabilistic model to discover the latent topics and correlations between documents and the influence of the links.

In this thesis, we propose a compact topic model to deal with the image-based recommendation problem. Some more related works are referred in Chapter 6.

2.4 Summary

The chapter lists the related research fields in our friend recommendation task. Section 2.1 introduces the general recommender system and the friend recommendation algorithms, as well as the state-of-the-art cross domain recommendation frameworks. Section 2.2 reviews some important feature selection and extraction methods and their applications in the social environments. Deep learning as a special and widely applied feature extraction method is also briefly discussed. Then we make a detailed review of the probabilistic topic model, including the concepts and modern applications. The algebra of random variables is also important for this thesis since it is the basis of the series expansion method proposed in Chapter 6.

Chapter 3

A Method of Discriminative Information Preservation and In-Dimension Distance Minimization for Feature Selection

In the beginning we first study the general feature selection problem. Preserving sample's pairwise similarity is essential for feature selection. In supervised learning, labels can be used as a direct measure to check whether two samples are similar with each other. In unsupervised learning, however, such similarity information is usually unavailable. In this chapter, we propose a new feature selection method through spectral clustering based on discriminative information as an underlying data structure. Laplacian matrix is used to obtain more partitioning information than other previously proposed structures such as the eigenspace of original data. The high dimension of sample data is projected into a low dimensional space. The in-dimension distance is also considered to get a better compact clustering result. The proposed method can be solved efficiently by updating the projection matrix and its inverse normalized diagonal matrix. A comprehensive experimental study has demonstrated that the proposed method outperforms many state-of-the-art feature selection algorithms with different criterion including the accuracy of clustering/classification and Jaccard score.

3.1 Introduction

There are many classification and clustering tasks to process large scale data or higher dimensional data, i.e. in the fields of bio-informatics, text processing and image processing. The high dimensional data raises two problems: firstly, it might be that only a small amount of dimensions of the overall spaces are useful for classification/clustering, and the rest of features are irrelevant for our expected classification/clustering result, these redundant features are likely to disturb our decision, and thus is harmful for correct classification/clustering; secondly, high-dimensional data leads to great computational cost both in time and memory.

A direct solution to this problem is provided by reducing dimensionality. The idea is to find a lower subspace representation of the original data while retaining some data properties. Basically there are three techniques to reduce dimensionality of the original data: One is called feature extraction. It creates a new lower dimension feature space by transforming or combining of the original features (Jain, Duin & Mao 2000). A classical unsupervised feature extraction method is Principal Component Analysis (PCA) (Johnson & Wichern 1988), which projects the original data into a lower subspace by maximizing the variance among data. Linear Discriminant Analysis (LDA) (Swets & Weng 1996) provides a supervised learning that attempts to best discriminate data classes. The problem of feature extraction is that the newly created features may not have a clear physical meaning and thus it is difficult to interpret the results.

Another approach to dimensionality reduction is to select a subset from the original features that are mostly informative. After the feature selection, different classification/clustering methods are applied. Traditional feature selection methods include SBFS and SFFS (Pudil, Novovičová & Kittler 1994), which iteratively select best features in each round by maximizing certain criterion functions. Compared with feature extraction technology, the feature selection method retains the original physical interpretation of each selected feature. However, the best subset of selected features may not provide a better discriminative ability than feature extraction.

The third approach is the feature selection combined with feature extraction to overcome the weakness of the above two techniques. Many of the recent feature selection methods are highly related to the extraction methods such as PCA in the sense that they choose representative features according to the data properties after projecting the original data into a lower space (Nie, Huang, Cai & Ding 2010)(Zhao, Wang, Liu & Ye 2013)(Gu, Li & Han 2011). In last decades this kind of feature selection became a hot topic in machine learning and in this chapter we focus on it. A lot of algorithms have been proposed and we name a few in the following.

Based on whether the label information is available, there are three kinds of feature selection methods. Firstly, when class labels of the training data are available, supervised feature selection algorithms are suitable for these cases. Typical supervised methods includes: Efficient and Robust Feature Selection via Joint $l_{2,1}$-norm Minimization(L21RFS) (Nie et al. 2010), Minimum Redundancy Maximum Relevance(mRMR) (Peng, Long & Ding 2005), Fish Score(FScore) (Liu & Motoda 2007), and Similarity Preserving Feature Selection(SPFS) (Zhao, Wang, Liu & Ye 2013). Secondly, sometimes only a small amount of class labels are available, which are not enough for correct classification. In such cases semi-supervised feature selection methods utilize both labelled and unlabelled data to obtain a series of representative features. Recent semi-supervised algorithms contain Locality sensitive semi-supervised feature selection (Zhao, Lu & He 2008) and Spectral Analysis for Semi-supervised Feature Selection (Pierre 2007). Lastly, for many tasks such as text clustering, the class labels are not available or not reliable, and unsupervised feature selection is applied. Because of the lack of label information, this is usually a more challenging task and many algorithms are developed to catch the more essential structure of data. These include: Trace Ratio (Nie, Xiang, Jia, Zhang & Yan 2008), Spectral Feature Selection (Zhao & Liu 2007), Laplacian Score (He, Cai & Niyogi 2005), SPFS, and Global and Local Structure Preservation for Feature Selection(GLSPFS) (Liu, Wang, Zhang, Yin & Liu 2014*a*).

In this chapter, we concentrate on the unsupervised learning. As no label information is provided, we may cluster samples according to their pairwise similarity (Nie et al. 2008)(Zhao & Liu 2007)(He et al. 2005). We propose a new approach for pairwise similarity preserving projection. In addition, we also add a regularization term in our formulation that minimizes the in-dimension distance. Details are given in the following sections. Experimental results show that our algorithm outperforms the state-of-the-art methods for clustering/classification.

The rest of the chapter is organized as follow: we propose our algorithm in section 3.3. Experiment results show the good performance of our algorithms in section 3.4, and we also make some analysis according the results. Finally we make our conclusion in section 3.5.

3.2 Related Work

For unsupervised feature selection, many previous works consider two aspects: first, the global pairwise similarity should be kept before and after selection; second, the neighbourhood relationship of the data should be maintained.

Pairwise similarity preservation follows the idea that two similar samples should have higher probability to be assigned into one group. As a result, the pairwise similarity relationship of the data should be kept during the projection from high dimensional space to lower dimensional space. (Zhao, Wang, Liu & Ye 2013) selects features that preserves pairwise similarity defined between high-dimensional samples. Unlike (Nie et al. 2010), which serves for supervised case and maximally preserves the class structure, (Zhao, Wang, Liu & Ye 2013) extends it into unsupervised case by applying pairwise similarity matrix instead of class label matrix. (He et al. 2005) seeks features that best present the underlying manifold structure of original data, by constructing a nearest neighbour graph and important features are chosen according this graph. (Nie et al. 2008) provides a more general form of (He et al. 2005) by considering minimizing the distances between certain pairs of nodes and maximizing between some other pairs according to different requirements. In general, all these works extract a low dimensional representation from the original data and achieve good performances in certain datasets. The problem lies on how to obtain a representative low dimensional subspace to make the clustering/classification precise. In this chapter, we propose a new selection method from the original data space based on spectral clustering.

All the above works do not consider the neighbourhood relationship, or local geometric structure of the data. (Yang, Shen, Ma, Huang & Zhou 2011) selects the most discriminative features by assuming that the class label can be predicted by linear classifier. It exploits local discriminative information from local total scatter matrix and local between class scatter matrix, and defines a discriminative score for all features. (Gu et al. 2011) preserves the relationships among data points by graph embedding. The purpose is to find the optimal low dimensional vector representation for vertices in a graph that best preserves local relationships. The above listed works don't take the global similarity preservation into account. In addition, these researches don't provide a clear explanation why local geometric structure works well for clustering/classification. Here we propose a deep interpretation of local structure from the aspect of in-dimension distance.

One Algorithm based on the combination of pairwise similarity and local geometric structure has been proposed in (Wei, Zeng, Wang, Wang & Wen 2014), which forms LDA as a least-square problem considering both global and local structures and develops a model selection scheme that balances the trade-off between global and local similarity. But this work does not select features. Recently, (Liu et al. 2014a) makes feature selections based on both global and local preservation. However, it simply uses eigenspaces of similarity matrix as the objective of the projection. Following the ideas of (Liu et al. 2014a), we propose a more discriminative subspace for clustering/classification.

In summary, we have two main contributions in this chapter:

1. We propose a new low dimensional representation of the original data which is more discriminative for classification/clustering.

2. We add a new regularization term to the optimization problem that represents the in-dimension distance minimization, and proves that this term is mathematically similar with the local geometric structure preservation, and thus can be solved efficiently by existing iterative methods.

3.3 Proposed Algorithm for Feature Selection

3.3.1 Previous Pairwise Similarity Preservation Methods

Similar with the steps of (Liu et al. 2014a), let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the data matrix with n samples, each has d dimensions of features. Directly using all features for clustering is in many cases a computationally expensive task because d is high. One way to overcome this problem is to project the d dimensional data to a lower dimensional space, while the useful information for clustering or classification preserves. It can be expressed mathematically as follows (Zhao, Wang, Liu & Ye 2013):

$$\min_{\mathbf{W}} \parallel \mathbf{XW} - \mathbf{V} \parallel_F^2 + \lambda \parallel \mathbf{W} \parallel_{2,1} \tag{3.1}$$

where $\mathbf{W} \in \mathbb{R}^{d \times r}$ stands for the projection matrix and r is the dimension of the projected space. For dimension reduction purpose we have $d > r$. The regularization term $\| \mathbf{W} \|_{2,1}$ forces $\mathbf{W}$ to be sparse and discards redundant features. $\mathbf{V}$ stands for a matrix that contains clustering or classification information of $\mathbf{X}$, and $\|.\|_F$ stands for Frobenious norm. In the supervised case, as in (Nie et al. 2010), it is the class label matrix of which each element in the matrix is 1 for that a sample belongs to one class or 0 for not. In the case that the class label is unavailable, different approach is required. One idea is that two samples that are similar with each other are likely to be assigned in one cluster, so the pairwise similarity relationship should be preserved during projection. (Liu et al. 2014*a*) preserves this relationship by an eigendecomposition of the pairwise similarity matrix K of original data, and r eigenvectors corresponding to the r largest eigenvalues compose the column of $\mathbf{V}$.

Though eigenvectors contain some information of the pairwise similarity of the samples, the method in (Liu et al. 2014*a*) is essentially only a simple extension of traditional PCA, which uses eigenvectors as indicative vectors. As we know, PCA is not originally designed for clustering/classification but for low dimensional data representation. Furthermore, it only fits certain kinds of datasets (Bishop 2006). This limitation triggers us to find another way to determine the objective matrix $\mathbf{V}$ that catches more information for clustering/classification.

3.3.2 Discriminative Information Preserving Projection

In this section we propose a new discriminative information preserving projection based on spectral clustering. Spectral clustering is a powerful clustering method in many fields such as image segmentation (Shi & Malik 2000). The basic idea of spectral clustering is to cut a graph so that connections between clusters are minimized, compared with the in-cluster connections.

It has been shown that the solution of spectral clustering is related to Laplacian matrix of the pairwise similarity matrix (Shi & Malik 2000). This fact tells us that Laplacian matrix may contain ample discriminative information for clustering than other relationship matrix of a graph. Inspired by this fact, we use Laplacian matrix as the indicator matrix.

Traditional spectral clustering algorithm solves the following optimization problem for partition:

$$\max_{\mathbf{U}} \quad trace(\mathbf{U}^T \mathcal{P} \mathbf{U}), \quad \text{s.t.} \quad \mathbf{U}^T\mathbf{U} = \mathbf{I} \tag{3.2}$$

where the normalized graph Laplacian is defined by:$\mathcal{P} = \mathbf{D}^{-1/2}\mathbf{K}\mathbf{D}^{-1/2}$, $\mathbf{D}$ stands for diagonal matrix whose entries are the sum of each row of $\mathbf{K}$. The solution of $\mathbf{U}$ is simply the eigendecomposition of $\mathcal{P}$.

For the above reasons, different from (Liu et al. 2014*a*) that simply uses eigendecomposition of original similarity matrix, we propose to use the eigenvectors corresponding to the first r largest eigenvalues of graph Laplacian $\mathcal{P}$ as the columns of $\mathbf{V}$ in Eq. (3.2).

3.3.3 In-Dimension Distance Minimization

Except for the clustering Information preserving projection, we also want to make the samples in each projected dimensions close to each other, if they are neighbours in the original data space. In other words, two close samples should also stay near each other in each dimension after projection. Thus after the projection, neighbour samples of the original data space are more likely to be arranged in one cluster.

To minimize the pairwise distances in one projected dimension, we set the data after projection as $\mathbf{A} = \mathbf{W}^T\mathbf{X}^T$, Also we set a pairwise distance matrix of all sample as $\mathbf{L}$, and then we want to minimize the in-dimension distance of all the r dimensions, here the in-dimension distance is defined as the sum of all pairwise distances in each dimension:

$$\begin{aligned} \min_{\mathbf{W}} \sum_{j=1}^{n}\sum_{i=1}^{n} \mathbf{A}_{1i}\mathbf{A}_{1j}\mathbf{L}_{ij} + \sum_{j=1}^{n}\sum_{i=1}^{n} \mathbf{A}_{2i}\mathbf{A}_{2j}\mathbf{L}_{ij} + \\ \cdots + \sum_{j=1}^{n}\sum_{i=1}^{n} \mathbf{A}_{ri}\mathbf{A}_{rj}\mathbf{L}_{ij} \end{aligned} \tag{3.3}$$

The first term of above equation can be written in a matrix form as follows:

$$\sum_{j=1}^{n}\sum_{i=1}^{n}\mathbf{A}_{1i}\mathbf{A}_{1j}\mathbf{L}_{ij} \tag{3.4}$$
$$\begin{aligned}
&= \sum_{i=1}^{n}\mathbf{A}_{1i}\mathbf{L}_{i1}\mathbf{A}_{11} + \mathbf{A}_{1i}\mathbf{L}_{i2}\mathbf{A}_{12} + \cdots + \mathbf{A}_{1i}\mathbf{L}_{in}\mathbf{A}_{1n} \\
&= [\sum_{i=1}^{n}\mathbf{A}_{1i}\mathbf{L}_{i1}, \sum_{i=1}^{n}\mathbf{A}_{1i}\mathbf{L}_{i2}, \cdots, \sum_{i=1}^{n}\mathbf{A}_{1i}\mathbf{L}_{in}](\mathbf{A}_{row1})^T \\
&= (\mathbf{AL})_{row1}(\mathbf{A}_{row1})^T = (\mathbf{AL})_{row1}(\mathbf{A}^T)_{colunm1} \\
&= (\mathbf{ALA})^T_{11}
\end{aligned}$$

Similarly, we have:

$$\sum_{j=1}^{n}\sum_{i=1}^{n}\mathbf{A}_{2i}\mathbf{A}_{2j}\mathbf{L}_{ij} = (\mathbf{ALA})^T_{22} \tag{3.5}$$
$$\cdots$$
$$\cdots$$
$$\sum_{j=1}^{n}\sum_{i=1}^{n}\mathbf{A}_{ri}\mathbf{A}_{rj}\mathbf{L}_{ij} = (\mathbf{ALA})^T_{rr}$$

By adding the r terms listed above together, we show that formulation (3.5) can be expressed as the trace of a matrix:

$$(3.5) = \mathrm{tr}(\mathbf{ALA}^T) = \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{LXW}) \tag{3.6}$$

As stated above, $\mathbf{L}$ is a distance matrix of all pairs of samples. We notice this form is mathematically equivalent to the local geometric preservation (Liu et al. 2014*a*). This equivalence lays on that by summing up the distance of all the low dimensions between each two pair of users we get the distance between the two in the whole low dimensions. So here we get the local structure preservation from the aspect of in-dimension distance.

Similar as (He et al. 2005) , we use the Gaussian kernel to determine the similarity between two samples from their feature space as follows:

$$\mathbf{S}_{ij} = \exp(\frac{\|x_i - x_j\|^2}{-2\sigma^2}) \tag{3.7}$$

where σ stands for the width of the Gaussian function. The distance matrix $\mathbf{L}$ is defined as the Laplacian matrix of $\mathbf{S}$: $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is the diagonal matrix with $\mathbf{D} = \sum_{j=1}^{n}\mathbf{S}_{ij}$.

3.3.4 Proposed Problem and Solution

By considering clustering information preserving projection and the minimization of in-cluster distance, the transform matrix $\mathbf{W}$ can be determined by solving the following optimization problem:

$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{V}\|_F^2 + \mu \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{LXW}) + \lambda\|\mathbf{W}\|_{2,1} \tag{3.8}$$

where λ and μ are regularization parameters. The $l_{2,1}$-norm constraint makes the problem hard to solve, according to (Liu et al. 2014*a*), it can be approximated by $\mathbf{W}^T\mathbf{BW}$, where $\mathbf{B}_{ii} = \frac{1}{2\|\mathbf{W}_{row-i}\|_2}$. So equation (3.8) is formulated as follows:

$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{V}\|_F^2 + \mu \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{LXW}) + \lambda \mathrm{tr}(\mathbf{W}^T\mathbf{BW}) \tag{3.9}$$

This problem has the same form as in paper (Liu et al. 2014*a*), it can be solved efficiently by iteratively solving $\mathbf{W}$ and $\mathbf{B}$. When $\mathbf{B}$ is fixed, $\mathbf{W}$ can be obtained as follows:

$$\mathbf{W} = (\mathbf{X}^T(\mathbf{I} + \mu\mathbf{L})\mathbf{W} + \lambda\mathbf{B})^{-1}\mathbf{X}^T\mathbf{V} \tag{3.10}$$

The iteration goes until convergence. (Liu et al. 2014*a*) have proved that this iteration converge to a global optimum. By ranking the rows of $\mathbf{W}$ according to their norms in descending order, we choose representative features from the top.

3.4 Experimental Results

3.4.1 Experimental Settings and Data Sets

In the experimental part, we evaluate our proposed algorithm with some state-of-the-art algorithms on different datasets. We show that our algorithm selects the most important feature for classification and clustering.

We compare our algorithm with the following unsupervised feature selection algorithms: Laplacian Score (He et al. 2005), SPEC(Zhao & Liu 2007), and GLSPFS(Liu et al. 2014*a*). All the codes for these algorithms are open and can be downloaded freely[1],[2].

Following (Liu et al. 2014*a*), We use eleven real-world data sets for all the tests, including image face datasets and some bio datasets for different diseases. All the datasets are public[3],[4]. We summarize the details of these datasets in table 3.1. For each dataset, we split the dataset by half randomly, one part is for training and the other part is for testing. We do such training and testing for 20 times to get the average results.

Table 3.1: Datasets Information

Data Set	Instance	Features	Classes	Keywords
AR10P	130	2400	10	Face
PIE10P	210	2420	10	Face
PIX10P	100	10000	10	Face
ORL10P	100	10304	10	Face
TOX-171	171	5748	4	Bio
CLL-SUB-111	111	11340	4	Bio
ALLAML	72	7129	2	Bio
GLIOMA	50	4433	4	Bio
LUNG	203	3312	5	Bio
Carcinomas	174	9182	11	Bio
Prostate-GE	102	5966	2	Bio

3.4.2 Evaluation Criteria

To show the performance improvements of our proposed feature selection algorithm, we apply the following criteria for tests: classification accuracy, Jac-

[1]featureselection.asu.edu
[2]https://sites.google.com/site/feipingnie/publications
[3]http://featureselection.asu.edu/
[4]https://sites.google.com/site/feipingnie/publications

card score, clustering accuracy, and Normalized Mutual Information(NMI).

We first evaluate the classification accuracy on the features we select from different algorithms. We use a Gaussian kernel Support Vector Machine(SVM) for classification.

We also test the Jaccard score of different algorithms, which is defined as the similarity between the nearest neighbours of samples calculated with all features and the nearest neighbours calculated with selected features. The higher the Jaccard score is, the more the selected features preserve original data structure.

Clustering accuracy is third criterion for evaluation. After feature selection, we run K-means clustering algorithm on selected features, and compare the results with the predefined class label.

Finally, another widely used criteria for clustering is NMI (Wagner & Wagner 2007). It describes how much we can reduce the uncertainty about the cluster obtained from the selected features when knowing its predefined class label. Details are given in the corresponding subsection.

In all of our experiments for the proposed algorithm and GLSPFS (Liu et al. 2014*a*) we choose regularization parameters λ and μ from $10^{[-2:1:3]}$ via four-fold cross-validation on the training set. We choose the dimension of the projected subspace r by experience and always shows the best r in both algorithms. This might be the reason that our experimental results are sometimes different from presented in (Liu et al. 2014*a*) because they fix r in their experiments.

3.4.3 Experimental Results

SVM Classification Accuracy

After choosing the representative features by proposed unsupervised algorithm, SVM classifications is applied on features chosen by different algorithms. From Figure 3.1 we see that the proposed algorithms has the average highest classification accuracy on bio data set TOX-171, compared with three reference methods. When we select few features (10 among 5748 features), the proposed method has a little worse performance than GLSPFS, but when we select reasonable enough features, the proposed methods outperforms all the reference methods. Table 3.2 lists the average SVM classification accu-

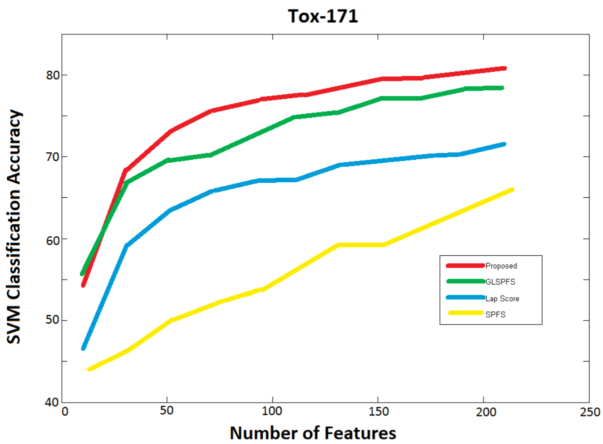


Figure 3.1: SVM Classification Accuracy: Tox-171

racy on different datasets as well as its standard derivation. From the table we see that our proposed algorithm outperforms other reference methods in all the datasets. Among the reference methods, GLSPES has better performance than others and is closest to the proposed method. This is what we expect because GLSPES has the similar form with our algorithm, the difference of the choice of the indicator matrix V. Because we choose a matrix with more discriminative information for clustering/classification, it leads to a better result.

Jaccard Score

Next we test the Jaccard score of different algorithm. The Jaccrad score of a user i is given as follows:

$$\mathbf{J}_i = \frac{A_i \cap B_i}{A_i \cup B_i} \tag{3.11}$$

Here A means the set of k nearest neighbours of user i measured on selected features, and B means the set of users measured on all features. The Jaccard score of all users is obtained by summing up of all $\mathbf{J}_i$.

The result on dataset TOX-171 is shown in Figure 3.2. In this table, we use the k=5 users as its neighbors for calculation. Again, the proposed algorithm achieves better performance over all reference methods, which means that the proposed method has stronger ability to preserve the data structure. The result on all data set is given in Table 3.3.

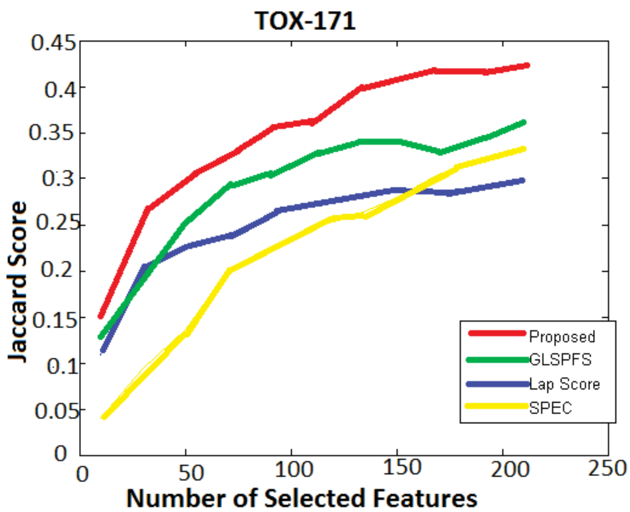


Figure 3.2: Jaccard Score: Tox-171

Clustering Accuracy

Jaccard score is a criterion of the feature for preserving data structure after projection, but not a direct measurement of whether the selection is right. Clustering Accuracy is in many cases a more important measurement for unsupervised feature selection. After feature selection, k-means clustering algorithm runs on selected features for rounds to get an average clustering result. The result is compared with the predefined class label. Figure 3.3 shows the clustering result on data set TOX-171. We find that the proposed method has the highest accuracy of all the proposed algorithms.

Table 3.4 lists the performance the proposed an reference algorithms over different datasets. We notice that SPEC algorithm has better performance on face datasets AR10P and PIE10P. The reason might be that SPEC considers the weights of the eigenvalues and thus performs better on certain datasets. However. our algorithm combines pairwise similarity and in-cluster distance and has highest clustering accuracy on most datasets.

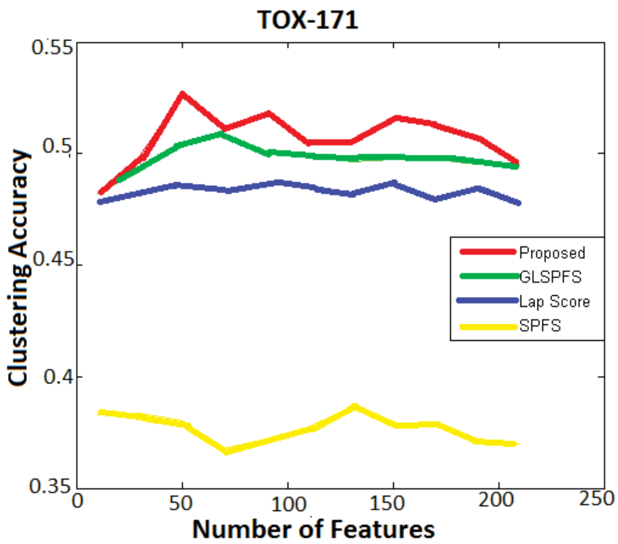


Figure 3.3: Clustering Accuracy: Tox-171

NMI

Formally, the mutual information between two clusters $\mathcal{T}$ and $\mathcal{C}$ that both have K clusters is given by

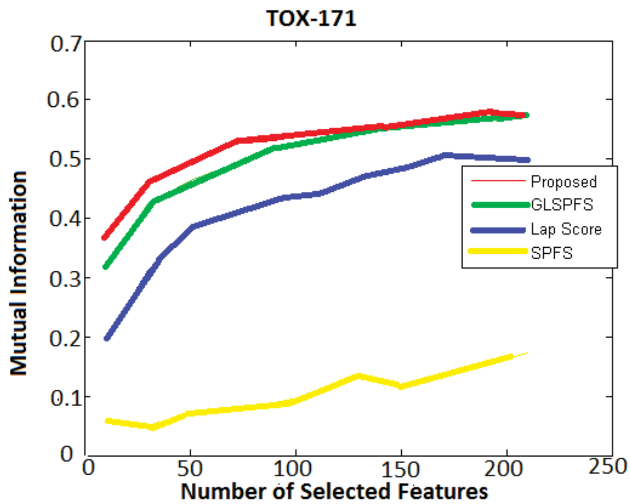
$$\mathbf{I}(\mathcal{T},\mathcal{C}) = \sum_{i=1}^{K}\sum_{j=1}^{K} P(\mathcal{T}=i,\mathcal{C}=j)\log_2 \frac{P(\mathcal{T}=i,\mathcal{C}=j)}{P(\mathcal{T}=i)P(\mathcal{C}=j)} \tag{3.12}$$

Where $\mathcal{T}$ and $\mathcal{C}$ are two different clustering labels of one dataset. The normalization is given as follows:

$$\mathrm{NMI}(\mathcal{C},\mathcal{T}) = \frac{\mathbf{I}(\mathcal{C},\mathcal{T})}{\sqrt{\mathbf{I}(\mathcal{C},\mathcal{C}),\mathbf{I}(\mathcal{T},\mathcal{T})}} \tag{3.13}$$

As stated above, it measures the reduction of clustering uncertainty by selected features. This measurement on dataset TOX-171 is given in Figure 3.4. And the results on different datasets are given in Table 3.5. The proposed algorithm again shows better performance than all reference methods.

Figure 3.4: NMI: Tox-171



3.5 Conclusion

Based on the pairwise similarity preservation and spectral clustering, this work has proposed a new approach for feature selection to get a more accurate clustering method, by utilizing the discriminative information contained in Laplacian matrix. In addition, the in-dimension distance minimization is considered to make the clustering result more compact. These two ideas are integrated as a optimization problem that has global optimum and can be solved by updating different variables iteratively. To verify the proposed algorithm, we conduct extensive experiments to show that the proposed method outperforms the existing feature selection methods under different criterion.

Table 3.2: SVM Classification Accuracy: TOX-171

Data Set	Proposed	GLSPFS	LapScore	SPEC
ALLAML	98.99 ± 2.71	96.46 ± 2.43	86.36 ± 4.42	81.82 ± 2.99
AR10P	91.54 ± 2.98	89.37 ± 3.29	80.24 ± 3.52	79.45 ± 4.79
Carcinom	85.27 ± 1.49	84.85 ± 1.94	80.25 ± 3.05	81.09 ± 1.8
CLL-SUB	78.39 ± 3.76	75.28 ± 3.4	69.91 ± 3.69	69.15 ± 4.03
GLIOMA	77.4 ± 4.77	74.8 ± 5.21	71.4 ± 5.3	70.0 ± 5.68
LUNG	95.64 ± 2.03	95.19 ± 1.45	92.4 ± 2.17	87.75 ± 3.42
ORL10P	87.7 ± 3.69	85.4 ± 4.09	73.6 ± 6.59	71.3 ± 5.48
PIE10P	98.48 ± 0.61	97.05 ± 1.34	89.86 ± 2.62	95.24 ± 1.72
PIX10P	96.2 ± 2.58	95.6 ± 1.75	89.6 ± 3.13	82.1 ± 4.54
Prostate	89.31 ± 2.05	89.02 ± 2.73	81.76 ± 3.68	85.49 ± 4.08
TOX-171	80.24 ± 4.25	78.29 ± 3.58	72.21 ± 4.7	66.23 ± 5.32

Table 3.3: Jaccard Score

Data Set	Proposed	GLSPFS	LapScore	SPEC
ALLAML	35.12 $\pm$ 2.56	32.92 $\pm$ 3.88	27.41 $\pm$ 2.99	20.26 $\pm$ 3.32
AR10P	37.11 $\pm$ 2.97	34.31 $\pm$ 3.37	26.14 $\pm$ 3.85	25.83 $\pm$ 3.34
Carcinom	29.95 $\pm$ 3.41	27.56 $\pm$ 3.34	20.58 $\pm$ 2.8	24.56 $\pm$ 2.85
CLL-SUB	22.68 $\pm$ 3.03	20.59 $\pm$ 3.51	14.73 $\pm$ 1.85	18.74 $\pm$ 1.03
GLIOMA	59.61 $\pm$ 3.73	57.46 $\pm$ 3.03	47.69 $\pm$ 3.64	46.43 $\pm$ 4.28
LUNG	28.32 $\pm$ 2.28	25.43 $\pm$ 1.86	24.81 $\pm$ 1.82	15.05 $\pm$ 2.93
ORL10P	65.27 $\pm$ 4.78	60.24 $\pm$ 5.28	46.17 $\pm$ 3.25	39.38 $\pm$ 4.62
PIE10P	44.62 $\pm$ 3.46	42.30 $\pm$ 4.13	26.82 $\pm$ 4.09	31.56 $\pm$ 2.97
PIX10P	96.2 $\pm$ 2.58	95.6 $\pm$ 1.75	89.6 $\pm$ 3.13	82.1 $\pm$ 4.54
Prostate	25.81 $\pm$ 1.36	22.78 $\pm$ 1.95	15.8 $\pm$ 2.48	12.85 $\pm$ 1.89
TOX-171	34.6 $\pm$ 4.2	29.13 $\pm$ 3.36	24.99 $\pm$ 4.59	22.18 $\pm$ 4.14

Table 3.4: Clustering Accuracy

Data Set	Proposed	GLSPFS	LapScore	SPEC
ALLAML	73.31 ± 8.39	72.01 ± 8.25	69.96 ± 6.27	69.32 ± 5.53
AR10P	38.81 ± 3.65	37.03 ± 5.35	38.44 ± 3.18	43.68 ± 3.11
Carcinom	71.38 ± 5.93	63.16 ± 3.68	62.81 ± 4.46	63.08 ± 4.59
CLL-SUB	22.68 ± 3.03	20.59 ± 3.51	14.73 ± 1.85	18.74 ± 1.03
GLIOMA	70.33 ± 4.1	66.96 ± 3.61	67.36 ± 3.7	68.11 ± 3.69
LUNG	69.53 ± 4.06	61.82 ± 4.86	55.58 ± 3.33	42.42 ± 4.08
ORL10P	53.01 ± 3.96	48.41 ± 1.94	48.57 ± 2.23	36.68 ± 3.29
PIE10P	40.67 ± 2.75	34.54 ± 2.67	33.42 ± 2.19	51.91 ± 4.19
PIX10P	79.63 ± 4.09	78.3 ± 3.28	77.15 ± 4.28	67.05 ± 4.11
Prostate	62.21 ± 4.12	61.95 ± 4.26	61.74 ± 4.7	62.04 ± 4.08
TOX-171	50.72 ± 2.09	48.3 ± 3.17	49.7 ± 2.58	37.72 ± 2.14

Table 3.5: NMI

Data Set	Proposed	GLSPFS	LapScore	SPEC
ALLAML	54.76 $\pm$ 6.09	49.15 $\pm$ 5.94	38.01 $\pm$ 4.3	36.6 $\pm$ 4.14
AR10P	71.29 $\pm$ 4.23	61.36 $\pm$ 3.58	47.08 $\pm$ 3.26	62.43 $\pm$ 3.79
Carcinom	74.15 $\pm$ 4.38	73.95 $\pm$ 4.52	68.33 $\pm$ 3.69	71.17 $\pm$ 3.03
CLL-SUB	18.97 $\pm$ 4.31	18.43 $\pm$ 5.24	15.31 $\pm$ 3.04	6.37 $\pm$ 1.58
GLIOMA	63.46 $\pm$ 6.12	62.12 $\pm$ 5.66	51.22 $\pm$ 6.2	53.15 $\pm$ 7.84
LUNG	78.06 $\pm$ 1.28	72.91 $\pm$ 1.68	61.36 $\pm$ 1.75	63.96 $\pm$ 1.17
ORL10P	84.51 $\pm$ 4.52	83.05 $\pm$ 3.98	71.04 $\pm$ 4.59	64.03 $\pm$ 5.74
PIE10P	97.4 $\pm$ 1.72	92.423 $\pm$ 3.5	76.76 $\pm$ 5.69	87.34 $\pm$ 2.36
PIX10P	93.45 $\pm$ 2.63	92.61 $\pm$ 2.2	88.88 $\pm$ 5.42	74.84 $\pm$ 4.98
Prostate	44.97 $\pm$ 6.49	41.88 $\pm$ 5.28	27.52 $\pm$ 5.13	30.78 $\pm$ 5.39
TOX-171	51.4 $\pm$ 2.75	51.02 $\pm$ 2.37	39.73 $\pm$ 4.7	10.26 $\pm$ 2.35

Chapter 4

Social Friend Recommendation Based on Multiple Network Correlation

From this chapter we start to study the friend recommendation task. It is an important recommender application in social edia. Major social websites such as Tweet and Facebook are all capable of recommending friends to individuals. However, most of these websites use simple friend recommendation algorithms such as similarity, popularity, or "friend's friends are friends", which do not satisfy the majority of users. In this chapter we investigate the structure of social networks and develop an algorithm for Network Correlation-based Social Friend Recommendation (NC-based SFR). To accomplish this goal, we correlate different "social role" networks, find their relationships and make friend recommendation. NC-based SFR is characterized by two key components: 1) We align related networks by selecting important features of each network. 2) Network structure should be maximally preserved before and after network alignment. After important feature selection we recommend friends based on these features. We conduct experiments on the Flickr network, which contains more than ten thousand nodes and over 30 thousand tags covering half million photos, to show that the proposed algorithm recommends friends more precisely than reference methods.

4.1 Introduction

Social networks have experienced explosive growth in the last decade. Social websites such as Twitter, YouTube and Flickr have billions of users who share opinions, photos and videos every day. Users make on-line friends through these social networks. One challenging issue is how to help these users to efficiently find new social friends. Social friend recommendation has therefore become a new research topic and several methods have been proposed (Chen, Geyer, Dugan, Muller & Guy 2009)(Wan, Lan, Guo, Fan & Cheng 2013).

Friend recommendation is a primary function in social network services and aims to recommend new social links for each user. Today when we lodge on the main social website such as Facebook, Twitter, and LinkedIn etc., we receive many recommendations of online friends. Seeing and hearing what the friends look at and listen to, or sharing our experience with our friends is an unparalleled experience. However, the decision of making friends is a complex human behaviour and can be affected by many different factors such as age, gender, location, interest (Alex 2012), etc. As a consequence, similar to real life, finding a good on-line friend is not easy without the help of good recommendations. Traditional friend recommendations widely applied by Facebook and Twitter are often based on common friends and similar profiles such as having the same hobbies or studying in the same fields. These methods usually provide a long ranked possible friend list, but the recommendation precision is usually not satisfactory due to its complexity.

Content similarity (such as image visual similarity) has been a primary clue for friend recommendation (Chen et al. 2009). However, we argue that many other social aspects need to be explored to systematically build high-performance social friend recommendation, other than basing recommendation purely on content similarity matching. People making friends often based on the following social aspects: 1) Social environment, including where one lives and works (Barnett & Casper 2001); 2) Social behaviours and actions, including one's working performance, shopping habits, hobbies, and, importantly, interactions with one another (Rummel 1991)(Weber 1991). 3) Social status, such as gender, age, position, etc. (Brym 2009) We summarize all these aspects as an individual's "social role". Here the term "social role" is the part that a person plays as a member of a particular society (Zhao, Wang, Yu, Liu & Zhang 2013). As stated in (Zhao, Wang, Yu, Liu & Zhang 2013): *"In on-line social networks, people behave differently in social situations because they carry different latent social roles, which entail various expectations that society puts on them."*. From our point of view, we believe that utilizing the individual's different social role information would be a new research component for recommendation tasks. In this chapter, We define network topology as the arrangement of edges of a network.

These different social roles can be perceived in different social networks, such as a basketball-fan network, football-fan network, etc. These networks have the same set of nodes(each node represents one individual) but with different edge connections between nodes, because the meaning of the edges are different. Although each network represents one kind of relationship, its topology is not independent of other networks. This is because an individual's various social roles are related to each other — a person's hobbies are usually related to gender and age, while his/her friend circle is related to hobbies/positions, and so on. We can also observe one's different social roles on web. For example, for an individual who uses the big image sharing website Flickr, he/she plays different on-line social roles such as a photo provider who shares his photos, as well as tags about his/her feelings about the photos, a photo connoisseur, or simply one who wants to find some friends who have some photos he/she also has interest. These individual's on-line social roles form different networks and these networks are related to each other. In this chapter we mine the correlations of these networks and propose a new approach for social friend recommendation. According to an individual's social role, we recommend friends through alignment between different networks.

To leverage correlations between different networks, we first present a social network as a graph in which the nodes of the graph are users and the edges stand for the relationships between users. Taking the contact and tag imformation on Flickr as an example, we build a contact graph in which the nodes are individuals and the edges represent their friendships. We then build a tag graph, in which the nodes are the same, but the edges represent the similarity of the tag set from each individual.

Figure 4.1 illustrates the tag and contact network of a group of Flickr users. The left hand side of Figure 4.1 is the Flickr tag similarity network of a small community with five people. The right hand side of Figure 4.1 is the Flickr contact network. We know the topologies of both networks except the edges connected with Phillip in the contact network. Phillip is new to the community and knows nobody else. He has already provided several tags that interest him via searching behavious and is seeking new friends on Flickr. No correlations between the two networks have been built in Figure 4.1, thus only simple content similarity recommendation based on the co-occurrence of tag can be applied for friend recommendation, whose accuracy is usually not satisfactory. Our problem is, how to build the correlation of these two networks and make reliable friend recommendation.

"Correlation" between networks means that the structures of different networks share some similar properties. Here the "structure" of a network is to some extent similar with "topology" but the meaning is broader: we define the structure of a network as the property of how the network is formed and organized. To determine the structure correlation, we propose to use the network alignment methods. It is defined as to find approximate isomorphisms between similar networks (Bayati, Gerritsen, Gleich, Saberi & Wang 2009), and have been widely applied in the fields of bio-informatics (Klau 2009) and computer vision (Conte, Foggia, Sansone & Vento 2004). In this chapter we take advantage of the study about network alignment in other fields such as bio-informatics into social media as a new approach.

To model the network correlations, in this chapter, we propose to align

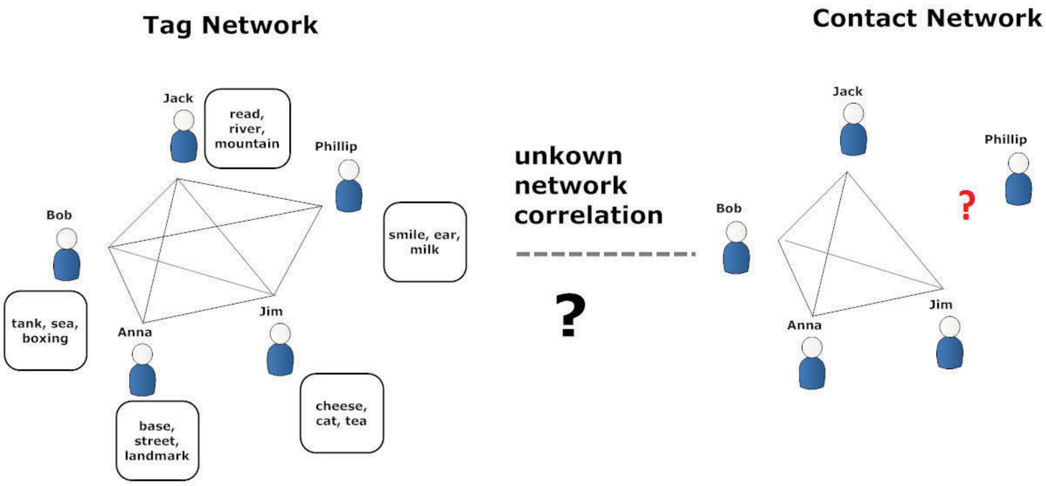


Figure 4.1: Problem Illustration: how to correlate the two networks and recommend friends to Phillip

tag and contact networks through important tag feature selection. Here an "important" feature is decided by if a feature contributes much in correlating the tag network to contact network, or in other words, makes the topologies of the two networks more similar. The reason we select important features is that a person usually presents different social signals in different social networks, which may have different importance in mining the network correlations. To give a more specific example, a photographer uploads images to Flickr tags such as "natural animals", "historical buildings", "street views" and "people". We view these tags as different feature words. In Flickr network, he may find that most of his friends contact him because of the photos tagged with "natural animals" and "historical buildings", rather than "street views" and "people". This indicates that the first two feature words are more important than the last two for friend recommendation.

In addition to network alignment, to make more precise friend recommendation, we also consider network structure preservation in our algorithm. Here "preservation" means that we don't change much the tag network structure before and after alignment. By preserving tag network structure on Flickr, we reduce the over-fitting risk of our algorithm. A number of previous works have applied this concept for classification/clustering (Liu, Wang, Zhang, Yin & Liu 2014*b*). In this chapter we analyse its correctness mathematically and extend the idea to social media.

Roughly speaking, the algorithm goes as follows. We align the tag and contact network by projecting the two networks in the lower dimensional spaces, in order to correlate them: we first project the contact network to its eigen-subspace, because eigenspace usually carries important information of the original space. Then we project the tag network to another lower-dimensional subspace. The two subspaces of tag and contact network should, to some extent, match each other for the more precise friend recommendation, compared to pure content similarity matching. One key point of our approach is on important feature selection for the network matching. Details are given in section 4.3 and 4.4.

In summary, this chapter makes the following contributions:

1. We have proposed a new friend recommendation method, based on network correlation by considering the effect of different social roles.

2. To model the correlation between different networks, we have developed a method by aligning these networks through important feature selection.

3. We also consider preserving the network structure for a more precise recommendation.

4. We have conducted comprehensive experiments to show that the proposed method significantly improves the accuracy of friend-recommendation. To reduce the problem of biased data, we choose a very large dataset that is randomly crawled from Flickr.

The rest of the chapter is organized as follows. Section 4.2 outlines related work. Section 4.3 introduces our framework and system model. Section 4.4 gives the details of our algorithm. Section 4.5 shows the performance of our method and analysis is made according to the result. Finally, Section 4.6 concludes our work.

4.2 Related Work

In this section we introduce several research fields that directly relate to our work.

4.2.1 Social Network Correlation

In this chapter we study the correlations between different networks. Network or graph matching/correlation problems have been widely studied in some fields such as image retrieval (Gori, Maggini & Sarti 2005), bio-informatics (Goodfellow, Wilson & Hunt 2010), etc. However, algorithms run well on image and biological datasets need modifications for social media problems: 1). Social networks deal with large scale complex networks. 2). As discussed, different social networks are formed when people play different social roles. The correlations of these different social networks are not well studied. So unlike the case in image retrieval or bio-informatics, the network correlation in the field of social media has its own properties and requires further study. (Zhuang, Yang & Wu 2008) is a pilot paper that studies the correlations among heterogeneous multimedia data. It studies the co-occurrence of the low-level features in different modalities and reinforce each other for information retrieval. But it doesn't study the correlation from a network view. (won You, won Hwang, Nie & Wen 2011) studies the matching of people's name and their social network identities such as their Twitter account with the help of common friends and co-occurrence of words. It illustrates that pure text similarity matching has poor matching performance. By synthetically considering the text, the popularity and the relationship among people, the correctly matching rate increases. This paper gives support to our idea that different social roles should be synthetically considered for a better recommendation. We further develop this idea in a way that we consider the structure of different networks and apply it for friend recommendation.

A more recent and related work for social media is given in (Liu, Ye, Chen, Yan & Chang 2012). In (Liu et al. 2012), first three networks are formed:

user friendship network, tag network and image content network. Different relations are then defined within each network and between different networks. According to these relations the transition probability is defined and a random walk-based algorithm is developed to calculate the relative score among different nodes. This propagation algorithm can be used for multi-purpose recommendation such as item/query/friendship, because the links among different networks can be inferred. Compared with (Liu et al. 2012), our algorithm focuses on the use of new network alignment method. Though both of the two use multi networks for link prediction, our proposed algorithm correlates networks with same nodes and provides a mechanism to choose important features. It gives a new point of view to interpret the property of the social network, compared with the pure propagation algorithm in (Liu et al. 2012).

4.2.2 Network Alignment

To find the correlations between different social networks, we propose to use network alignment methods. Some previous researches consider the combination of different social networks for user behaviour prediction. (Zhong, Fan & Yang 2014) considers different behaviours such as music listening and booking reading for a composite behaviour prediction. It uses graphical model to build the relations of different networks. (Pan, Aharony & Pentland 2011) utilizes the different application installation information from mobile devices. It would be better if these researches utilize the ample topological information of different network for a better result. In this chapter, we consider the different social roles of individuals and use the topological information of different networks by alignment. The concept of network alignment is applied in different fields that studies the relationship of big networks. In addition to the fields of bio-informatics and image processing problem as mentioned in the introduction part of this chapter, it has also been applied to different fields that deals with problem with large networks such as internet network management (Kreibirch & Crowcroft 2006). In (Kreibirch & Crowcroft 2006), alignment is used to find the co-occurrence of elements of different networks and reduce the traffic of internet data. In this work, we extend network alignment concept to social network, by considering different social roles of individuals for a better recommendation.

4.2.3 Feature Selection

In this chapter, we align different networks together through important feature selection. The initial motivation for feature selection is that the dimension of many social data is very high (Bellman 2003). To deal with this problem we usually apply feature extraction (Jolliffe 1986) or feature selection (Liu & Mototda 2008) methods. Feature selection is often preferred over extraction, for the selected features have more understandable physical meanings. Feature selection has been successfully applied in the fields of biology and image processing (Liu et al. 2014*b*). In this chapter we concentrate on unsupervised feature selection method.

(Cai, Xhang & He 2010) provides a clustering method based on spectral embedding. It projects the data on a subspace, chooses features that minimize the distance in each cluster on the projected subspace. However, it doesn't consider the pairwise structure of the original data. (Liu et al. 2014*b*) provides a model that considers both the local and global structure preservation during projection. (Liu et al. 2014*b*) induces the global preservation from linear kernel functions that can be applied for both supervised and unsupervised case. It is applied to image and bio datasets for clustering. Feature selection can also be applied in multimedia analysis that selects features from different domains (Yang, Ma, Hauptmann & Sebe 2013). In this chapter we extend the traditional feature selection algorithm to the field of social media. By carefully analysis, we apply the concepts of structure preservation for a better use of social features for recommendation.

Some previous works have combined the concept of feature selection and similarity network(kernels) alignment together for different purpose. (Guo, Man-Wai & Kung 2006) studies the use of profile alignment and support vector machine for cellular localization. (Kosir, DeWall & Mitchell 1996) has applied network alignment to overcome the problem that the locations of features varies from measurement to measurement for image matching. Our work differs from these previous works in two aspects: First, these works are mostly supervised and concentrate on image processing and bio-informatics. In our proposed network, we extend the concept to unsupervised cases and deal with more complex and bigger social network for social recommendation purpose. Second, most of the previous works are based on kernels that only applied the similarity information between users. Different from these kernel-based methods, we utilize more detailed information that is not only about the relationships between individuals, but also their social roles. These information is relatively easy to obtain in social media and so we expect better result than pure kernel methods.

4.3 System Model and Framework

In this section we present our framework. Details of the algorithm are given in section 4.4.

4.3.1 Problem Statement and Notations

In NC-based SFR, there are different networks including a contact network, $\mathcal{C}$ and $\mathcal{T}$(Taking a real world example, $\mathcal{C}$ stands for the contact network and $\mathcal{T}$ for the tag similarity matrix on Flickr). $\mathcal{C}$ and $\mathcal{T}$ have exactly the same nodes but different topologies. As mentioned in Chapter **??**, the different social roles of individuals are related to each other. $\mathcal{T}$ shows individual's interests and $\mathcal{C}$ shows the friendship. So it is reasonable to assume that the topologies of tag and contact networks are correlated. In this chapter, we propose a method to make more precise friend recommendations based on the correlations of different networks through their alignments.

Specifically, when a new node comes into network $\mathcal{T}$, we know its links with other nodes in $\mathcal{T}$, but we do not know its links in network $\mathcal{C}$. Our research seeks to predict its links in $\mathcal{C}$. A real world example for this scenario is that when a new user comes into a social network, he/she may provide interesting keywords. The system should make friend recommendations for the new user, but traditional content similarity recommendation methods do not take the different aspects of social roles into account. In our approach, the alignment between different social role networks is considered and thus a more comprehensive friend recommendation is obtained. We expect better performance using our algorithm.

Following are some of the nations used in this chapter. In total, there are N nodes in $\mathcal{C}$ and $\mathcal{T}$. The similarity matrix of network $\mathcal{C}$ is given by $\mathbf{K} \in \mathbb{R}^{N\times N}$. In the above Flickr example, $\mathbf{K}_{ij}$ is a binary number where "1" means $user_i$ and $user_j$ are online friends, while "0" means they are not. $\mathbf{X} \in \mathbb{R}^{N\times F}$ is the feature matrix of network $\mathcal{T}$, where F stands for the dimensions of features to represent each node. In Flickr, F stands for the length of the whole dictionary of tags and $\mathbf{X}_{ij}$ stands for whether $user_i$ uses tag_j. We also introduce the $N \times N$ matrix $\mathbf{L}$ according to the tag similarity of each pair of users.

4.3.2 Our Framework

To make a prediction of network links, according to the previous analysis, we propose to apply feature selection techniques to find the alignment of different networks that have same nodes and different topologies.

In Figure 4.2, we show the framework of our whole system. When we have the original tag and contact network as input(Fig 4.2a), we first project the contact network to its eigen-space and extract tag features(Fig 4.2b)– in our case, features are the tag words provided by the photo uploaders. Then we align the tag network $\mathcal{T}$ to the eigen-representation of the contact network $\mathcal{C}$(Fig 4.2c) by considering network correlation and structure preservation. In the last step we select some important word features from the whole feature set (which is composed of all the tag words). These important tag features illustrate the correlations between tag and contact network. In other words, these features make the tag network more similar to the contact network. So when a new user with some tags comes into the network, based on how his/her tag features matches to the pool of those important features that have been selected previously, we can map him/her to the existing contact network to see which users are closer to the new one, these closer users are more likely to be his/her potential friends. Details of each step are explained in Section 4.4.

4.4 NC-based SNC

In this section, we give the details of our algorithm by considering the correlation of two networks.

4.4.1 Approach Overview

We start to find the alignment of two networks. In this chapter we align different networks by selecting important features that catch the similarity of different networks. Taking the tag and contact networks as an example, we usually judge a person who might be our potential friend by only few words. For example, when we find a friend on Facebook, usually we don't read all posts of a person, which is too time consuming. Instead we only read titles of several of his/her posts, and then we get a rough but to some extent accurate understanding of what he/she likes. Triggered by this phenomenon, we assume that individual's friend making decision is determined by a small amount of features from a large feature set. The whole feature set may contain tags, photos, comments, geo information, etc. And according to the previous discussion, we assume that based on a relatively small amount of features, different networks should be aligned well. In this chapter we choose tag features for correlation with contact information, and the idea can be extended to different kinds of networks.

We choose features from two aspects: In the first aspect, features are chosen that correlate two networks well. We fix one network $\mathcal{C}$ and align the topologies of the other networks such as $\mathcal{T}$ in a subspace. Details for network correlation are given in 4.4.3.

In the second aspect, in addition to network alignment, we choose features that preserve the original structure of the modified network $\mathcal{C}$. In other words, nodes that are close to each other in the original network should also be close enough in the modified network. Thus the network alignment doesn't change the pairwise similarity among nodes. The effectiveness of pairwise similarity preservation has been shown in (Liu et al. 2014*b*)(Yang, Xu, Liu, Ma & Sebe 2013). By doing this we may predict links for new nodes for network $\mathcal{C}$ according to the existing links in modified network $\mathcal{T}$ more precisely. We will discuss it more carefully in 4.4.4.

4.4.2 shows some small but non-trivial methods to filter noise and redundancies. 4.4.3 and 4.4.4 illustrate the details of proposed algorithm, 4.4.5 gives the solution as well as complexity analysis of proposed algorithm.

4.4.2 Bag of Words and Feature Extraction

In the alignment of tag and contact network, we treat tag words as features. The tag data crawled from social website such as Flickr usually contains much noise and thus data refinement is required for a better recommendation result. After removing some explicit stop features such "a","the", as well as features that too often or too seldom appear in Flickr tags. After this we build the vocabulary of tags.

To calculate the tag feature matrix $\mathbf{X}$, we adopt the widely used TF-IDF method (Wu, Luk, Wong & Kwok 2008). Except for counting the numbers of words that each user has used as the tags of his/her photos, TF-IDF assumes that seldom appeared features carry more information. Under this assumption, TF-IDF diminishes the weight of features that occur frequently in dataset and increases the weight of features that occur rarely. By calculating the TF-IDF of each word, we build feature matrix $\mathbf{X}$.

4.4.3 Network Alignment

As mentioned in 4.4.1, first we consider minimizing the gap between graph $\mathcal{C}$ and $\mathcal{T}$ by selecting important features. Assume we have a feature selection matrix $\mathbf{W}$, then the selected features from the whole feature matrix can be expressed as $\mathbf{XW}$. To make the user-user similarity between $\mathcal{C}$ and $\mathcal{T}$ as small as possible, we have the following formulation:

$$\begin{aligned} &\min_{\mathbf{W}} \parallel \mathbf{X}\mathbf{W}\mathbf{W}^T\mathbf{X}^T - \mathbf{K} \parallel_F^2 \\ &\text{subject to} : \mathbf{W} \in \{0,1\}^{F\times r} \\ &\qquad \mathbf{W}^T\mathbf{1}_{F\times r} = \mathbf{1}_{r\times 1} \end{aligned} \tag{4.1}$$

Where $\mathbf{K}$ is the similarity matrix of $\mathcal{C}$ defined in 4.3.1. $\mathbf{W} \in \{0,1\}^{F\times r}$ means that $\mathbf{W}$ can only be chosen from $\{0,1\}$ and $\mathbf{W}^T\mathbf{1}_{F\times r} = \mathbf{1}_{r\times 1}$ means that the sum of each row in $\mathbf{W}$ is exactly 1. These two constraints ensure that each feature can only be chosen once so we don't get a linear combination of features. This is a discrete problem and is hard to solve. Also, the user similarity in (4.1) measured by inner product makes the dimension too high for optimization. In (Liu et al. 2014*b*), the problem is approximated by the following:

$$\min_{\mathbf{W}} \parallel \mathbf{XW} - \mathbf{V} \parallel_F^2 \tag{4.2}$$

Where $\mathbf{V} \in R^{N\times r}$ stands for the r eigenvectors corresponding to the largest eigenvalues of $\mathbf{K}$. r also stands for the dimension of the subspace the data projects in. (Liu et al. 2014*b*) shows that the upper bound of the solutions between Eq.(4.2) and Eq.(4.1) is small. By adding a regularization term we have the following optimization problem:

$$\min_{\mathbf{W}} \parallel \mathbf{XW} - \mathbf{V} \parallel_F^2 + \lambda \parallel \mathbf{W} \parallel_{2,p} \tag{4.3}$$

The regularization term $\parallel \mathbf{W} \parallel_{2,p}$ forces $\mathbf{W}$ to be sparse. This equation is similar as Eq. (3.1), except the regularisation parameter p. This is a improvement of Eq. (3.1) in that p controls the sparsity of $\mathbf{W}$. By applying $l_{2,p}$ norm it is possible to control how sparse the projection matrix $\mathbf{W}$ should be.

In this way, we make the gap between two networks as small as possible.

4.4.4 Structure Preservation

Only considering gap minimization between two networks might lead to a problem that the structure of the modified network to be changed greatly. (Liu et al. 2014*b*) have shown by experiments that by preserving the pairwise similarity of the original data structure, the clustering and classification performance is improved, compared with pure gap minimization. (Liu et al. 2014*b*) studies the problem in image and biological data. In this subsection we extend the idea of pairwise similarity relationship to the field of social network. First we calculate the structure of the original dataset.

Data Structure Representation

The structure of a network can be expressed as the pairwise similarity between each two different nodes in the graph. For tag feature matrix, it can be expressed as the semantic meaning of words between users.

After we obtain feature matrix $\mathbf{X}$, a simple way to calculate the similarity between two users is to count their number of co-occurrence features. However, this method is too simple and sometimes fails to catch the similarity among users. For example, a user with a tag "river" might have more similar topics with a user with "mountain" than another user with "basketball". The above mentioned simple method can't catch this similarity. Here we use Wordnet[1] to calculate the semantic similarity among different features (Nithiya, Vidhya & Ganesan 2010). The Wordnet groups words into sets of synonyms and different synonyms are connected with hypernyms/hyponyms (as a simple example, *apple* is a hyponym of $plant$ and $food$, $plant$ and $food$ are hypernyms of *apple*).According to (Jin, Khan, Wang & Awad 2005), different methods can be applied for feature similarity measurements. Two different features will get a relation score between 0 and 1.

[1]https://wordnet.princeton.edu/

Recently there are other tag feature similarity methods in the field of natural language processing such as DSSM (Huang, He, Gao, Deng, Acero & Heck 2013) and Google distance (Cilibrasi & Vitanyi 2007). As Wordnet is widely applied in many previously works, we apply Wordnet in our work for similairity calculation by following the method in (Nithiya et al. 2010).

After we get the relation score of different features, we calculate the similarity among different users to get L. for $user_i$ having a tag set of $word_{1i}, word_{2i}, ..word_{si}$, $user_j$ having a tag set of $word_{1j}, word_{2j}, ..word_{tj}$, we have the relation score matrix $\mathbf{E}ij \in s \times t$. and the similarity between i and j is given by:

$$\mathbf{L}_{ij} = \sum_{s} \sum_{t} \mathbf{E}ij_{st} \tag{4.4}$$

Structure Preservation

After we represent the data structure, we study how to preserve the network structure during the alignment.

To minimize the pairwise distances, we define the data after feature selection as $\mathbf{A} = \mathbf{W}^T \mathbf{X}^T$, Also we set a pairwise distance matrix of all samples as $\mathbf{L}$, and then we want to minimize the in-dimension distance of all the r dimensions:

$$\begin{aligned} \min_{\mathbf{W}} \sum_{j=1}^{n} \sum_{i=1}^{n} \mathbf{A}_{1i} \mathbf{A}_{1j} \mathbf{L}_{ij} + \sum_{j=1}^{n} \sum_{i=1}^{n} \mathbf{A}_{2i} \mathbf{A}_{2j} \mathbf{L}_{ij} + \\ \cdots + \sum_{j=1}^{n} \sum_{i=1}^{n} \mathbf{A}_{ri} \mathbf{A}_{rj} \mathbf{L}_{ij} \end{aligned} \tag{4.5}$$

The mathematical deducing procedure is similar as the procedure in Chapter 3. Finally a compact term is obtained as follows:

$$\mathrm{tr}(\mathbf{A}\mathbf{L}\mathbf{A}^T) = \mathrm{tr}(\mathbf{W}^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{W}) \tag{4.6}$$

4.4.5 Solutions and Complexity Analysis

Solutions

By mixing up optimization problems in 4.4.3 and 4.4.4 together, we have the following optimization problems:

$$\min_{\mathbf{W}} \|\mathbf{XW} - \mathbf{V}\|_F^2 + \mu \mathrm{tr}(\mathbf{W}^T\mathbf{X}^T\mathbf{LXW}) + \lambda\|\mathbf{W}\|_{2,p} \tag{4.7}$$

where λ and μ are regularization parameters. The equation is quite similar as Eq. (3.8), expect the parameter p. When the value of p is limited to be greater than 1, Eq. (4.7) is convex. And the solution has similar procedure as Eq. (3.8)

Feature Selection

The next is to choose important features according to the optimal $\mathbf{W}$. From Eq.(4.1) we see that each row of $\mathbf{W}$ corresponds to one feature. The larger the norm of this row, the more important role this feature plays in aligning network $\boldsymbol{\mathcal{T}}$ to network $\boldsymbol{\mathcal{C}}$. The norms of rows that are nearly zero mean that the corresponding features make nearly no contributions in alignment. So we rank the norm of rows of $\mathbf{W}$ according to their norms in a descending order, and choose features according to this order. The top features are considered to be most important for friend prediction. In the experimental part, we verify that by considering weight we get a slightly better performance.

Friend Recommendation

For a new user with some tag words coming into the network, how do we make recommendations according the important features? As mentioned before, we select these features based on the alignment from tag network to contact network, and these important features illustrate the correlations of contact and tag networks. So we calculate the similarities between tag features of the new user and those important features of the existing users. Because the important features pool reflects each existing user tag's contribution and the correlation between tag and contact networks, therefore, this similarity indicates the distance of the new user to those existing ones in the contact network. The more similar on the important tag set, the more closer the two uses should be. So by ranking the tag similarity of the new user and the members that already in the networks, we choose top K as recommending friends.

The whole process is shown in Algorithm 4.1.

Algorithm 4.1 Proposed NC based SFR

Require:

tag feature matrix $\mathbf{X}$, contact matrix $\mathbf{K}$, tag feature vector of the new user $\mathbf{x}$, number of friends K

Ensure:

Friend recommendation list

1: Determine λ, μ and p via cross validation on training set
2: Calculate the tag relationship matrix $\mathbf{L}$
3: Calculate $\mathbf{V}$ by eigen-decomposition of Laplacian of $\mathbf{K}$
4: Initial $\mathbf{B}$ with identity matrix $\mathbf{I}$
5: **repeat**
6:     Calculate $\mathbf{W}$
7:     Calculate $\mathbf{B}$
8: **until** *Convergence*
9: Calculate the norm of each row of $\mathbf{W}$. Rank the norms in a descending order.
10: Choose important features from top of the ranking list.
11: map the new user to the existing contact network by calculating the similarities between the important features of the new user and those of the existing users. This similarity indicates the distance of the new user to those existing ones in the contact network. So we choose the top K users according to the similarity as recommended friends to the new user.

Parameter Choice

The step 1 in Algorithm 4.1 determines the best value for parameters λ, μ and p. In practise, the best value of parameters μ, λ and p is determined by cross validation method on the training set. We choose the best value of the parameters so that we get the highest friend recommendation accuracy.

Complexity and Large Scale Data Suitability

The most time-consuming task in Algorithm 4.1 is the calculation of computing tag similarity matrix $\mathbf{L}$. Assuming the time for each similarity function of WordNet takes τ seconds, because $\mathbf{L}$ is symmetric, the time to calculating relation score matrix $\mathbf{E}$ is $F \times (F-1) \times \tau \,/\, 2$.

To calculate $\mathbf{L}$ from $\mathbf{E}$ by Eq.(4.4), it takes totally $N \times N \times s \times t$ sum operation.

The most time-consuming job in algorithm 4.1 is the inverse of matrix of sieze $N \times N$ or $F \times F$. So the complexity at each iteration is $\mathcal{O}(\min\{N, F\}^3)$.

From the above discussion, it is clear that the complexity of the algorithm is mainly determined by $(\min\{N, F\}^3)$, which means that when the number of features is fixed, the complexity doesn't increase with the increase of the scale of data. So our algorithm fits large scale social network that contains millions of users. This is a good property of our algorithm.

4.5 Experiments

In this section we make extensive experiments to show the effectiveness of our proposed method, as well as illustrate some interesting properties. First we give a brief introduction of our social media data set, and then we discuss our algorithm from different aspects.

4.5.1 Experimental Settings

Dataset

We crawled a social network from the big image sharing site Flickr. To reduce bias, we crawled groups randomly. As the data set is quite large, a relatively un-bias dataset can be obtained. A “group” in Flickr is a user-created album that relates to a topic, such as “Sydney”, “bike”, “autumn”, etc. Members of this group can upload photos to this group for sharing. Together we have crawled the data of 10000 users from 2000 groups. In our experiment, in each group we crawled 5 users.

For each user we crawled all their photos, and tags of each photo. For same users, we crawled their contact information to form the contact network. In Flickr, the contact information is obtained by if a user has added another user to his/her friend list, or vise versa. We crawled all the contacts between any of the two users in our dataset. A short summary of our dataset is given in Table 4.1 :

Table 4.1: Dataset Statistics

Users	10000
Groups	2000
Photos	543,754 photos from 10000 users
Contact	145,684 friend links among users
Tags	35,574 words after filtering

In our experiments, the features in table 4.1 are tag words. Other features such as image features and geo features might be integrated in future works.

Settings and Metrics

Our task is to make precise contact information prediction. In this way when a new user comes into the social network, by providing some key words that he/she has interest, we recommend new friends to him/her. By considering different aspects of social roles, we correlate the tag feature network and contact network through alignment for friend recommendation.

In friend recommendation, assume for each user we recommend K friends to him/her. We use the existing contact information as the ground truth for training and testing. Parameters λ, μ and p are determined on the training set by a fourfold cross validation to find the best. The ranges for these parameters are: $\lambda \in 10^{[-2:1:3]}$, $\mu \in 10^{[-2:1:3]}$, and $p \in [1, 1.5, 2, 2.5, 3]$.

We use the method stated in 4.4.5 to recommend friend to new users. We may use the precision and recall metrics to show the effectiveness of the proposed algorithm. In our experiment, precision is defined as the correctly recommended friends divided by all the recommended users, and recall is defined as the correctly recommended friends divided by the number of all truly existing friends.

For example, if we recommend 5 friends to 5000 users, the precision is calculated by all the truly recommended friends divided by 5×5000, and the recall is calculated by all the truly recommended friends divided by the number of all the friends of the 5000 users. In later chapters, the precision and recall are also calculated in this way.

One problem for precision and recall is that usually when one becomes large, the other becomes small, so we use F-measure to combine the two:

$$F = 2 \times \frac{pecision \times recall}{precision + recall} \tag{4.8}$$

In our experiment we choose 80% for training and 20% for testing, and run totally 20 times to calculate the average precision, recall, and F-measure.

Reference Methods

We use several reference methods to show the advance of our proposed algorithm in friend recommendation. They are: 1).pure tag similarity, 2).SVM, 3).on-line collaborative filtering(OLCF) (Rendle & Thieme 2008), and 4) Relational Domain Recommendation(RDR) (Jiang, Cui, Wang, Yang, Zhu & Yang 2012).

The first is the simplest tag similarity comparison. We recommend friends of each user purely on the tag similarity calculated in 4.4.2.

The second method is an SVM-based method. It implements the "one-against-one" approach for multi-class classification (Knerr, Personnaz & Dreyfus 1990). We choose SVM as reference method because in social media, friend recommendation can be viewed as a classification problem, where each user is classified as "recommended" or "not recommended". In our experiment we crawled the data of 10000 users from 2000 Flickr groups. Each group has 5 users and we assume each group as one labelled class. Because each group is very small, we assume the members in each group are friends with each other. For training we choose 4 users in each group and test if the last one can be classified correctly. If it is correctly classified, we assume we have right friend recommendation result.

The third method is OLCF. Collaborative filtering method is widely used in recommender system. It fills the blank entries of the user-item matrix. In our experiment we use a model-based collaborative filtering method to determine the votes of each user of each features of the whole feature space, and then calculate the cosine similarity between users. By ranking these similarities we recommend friends.

Some traditional model-based collaborative filtering methods face the problem that when a new user comes, the whole latent space has to be updated (Ma et al. 2008*a*). In this chapter, we apply an online-updating collaborative filtering method as reference method (Rendle & Thieme 2008).

At last we consider an multi-network based algorithm for comparison. When considering social multiple network problems, transition probability propagation is a method that is frequently used (Jiang et al. 2012)(Liu et al. 2012). We choose (Jiang et al. 2012) as a reference method for the following reasons: 1) It considers the relationships of different networks which is similar with our idea, though in (Jiang et al. 2012), different networks have different nodes; 2). It uses the information of other networks for recommendation, which again has some similarities with ours. (Jiang et al. 2012) enhances the links in one network and among different networks using a random walk propagation method. After enough round of walks it obtains the modified link weights between each user pair. And we use the weights for friend recommendation. (Liu et al. 2012) also uses a random walk base method but considers more kinds of relationships. Due to the limit of space we don't fulfil it here but leave it to further work.

4.5.2 The effect of number of chosen words

First we study the algorithm performance with different number of features. We change the selected number of features from 300 to 12000, and test first the precision for each number of features. We recommend the first 20 most similar users as friends for each user with certain number of features. We determine parameters in Eq.(4.7) with a five cross validation procedure for the best value. The result of precision is shown in Figure 4.3:

From Figure 4.3 we see that the algorithm performance increases quickly when the number of features is relatively small. There is a turning point when we choose the number of features around 4500. When the feature number exceeds 4500, the performance doesn't increase much. So according to the experiments, in practise there is a turning point for the number of features. The most efficient way is the determine the number of features around this turning point.

This experiments shows that when a user looks for friends, he/she only concentrates on some words or aspects of the characteristic of others but not all aspects.

4.5.3 Feature Selection vs. Selection plus Re-weighting

After the calculation of 4.4.5 we get the importance of each features. Then we rank the features according their importance, and we get a list of features, on the top of the list are those features with most importance. Two methods can be applied in the next step: First, we choose important features from the top of the list; Second, we also choose the features from the top of the list, and consider their weightings for friend recommendation. The following are the results for pure feature selection and selection with re-weighting.

We fix the number of selected features to be 4500 and the total number of the dataset to be 10000, and change the number of recommended friends. The result is shown in figure 4.4.

We see that there is only slightly improvement when considering the weights of each feature. It shows that in our algorithm, slightly changing the weight does not lead to great performance improvements. It is a trade-off between the effectiveness and complexity by changing the weights of the features.

4.5.4 Comparison of Different Methods for Precision

In this experiment we compare the proposed method with all reference methods mentioned in section 4.5.1.

We fix the number of features for proposed method to be 4500. Now we compare the Precision Measure performance of different methods with the reference methods mentioned in 4.5.1. We change the value of K from 5 to 30 for a more complete comparison. The resulting histogram is given in Figure 4.5: From Figure 4.5 we see that no matter which K we choose, the proposed method outperforms all other reference methods. Pure similarity method has the lowest recommendation precision. This coincide our statement in Chapter 1, that people make friends not only based on similarity.

Collaborative filtering method OLCF has slightly lower performance than SVM. The reason might be that by filling the user-feature matrix, it adds some noise and redundancy. So it doesn't have a good performance.

Propagation-based algorithm RDR has the second best performance. It enriches the user-to-user link by random walk in contact and other networks and thus has a better performance than SVM. However, it lacks a mechanism to consider what is important in friend making decision, which has been carefully considered in our proposed algorithm.

The proposed method has the best friend prediction accuracy. This is because we correlate the tag information with the contact network. We choose those most important features when people make friends with each other.

When we increase the number of selected feature to be 7500, the result is shown in Figure 4.6

Figure 4.6 shows that the performance of proposed method does not increase much with increase of features, when the number of features is relatively large. The reason is that our weighing matrix **W** is sparse, and for features that have a low order, their weights is very small or even becomes zero. So they don't have much influence on the recommendation accuracy.

4.5.5 Comparison for Recall and F-measure

As mentioned before, recall and F-measure are also common metrics to measure the effectiveness of recommendation algorithms. In the following, Figure 4.7 and Figure 4.8 give the results of our proposed method and the reference methods. Here we fix the number of recommended users to be 5, the number of features to be 4500, and change the total number of users from 5000 to 10000.

From Figure 4.7 and 4.8 we see that the system performance changes when the number of users are changed. And our proposed method always has the best performances.

4.5.6 Effect of Parameters

In this experiment we study the effect of parameter μ. It adjusts the weight between alignment of two networks and preservation of data structure. Still we fix the feature number to be 4500 and K to be 20, λ is fixed to be 1. The result is given in figure 4.9:

Figure 4.9 shows that with the increase of μ, first the prediction accuracy increase to a maximum point, then it decreases. This phenomenon tells us that both the network alignment and network structure preservation play important role in important feature selection. A maximum friend recommendation precision is reached when we balance both of the two well.

Different choice of p shows how sparse the projection matrix $\mathbf{W}$ is. the following is the experimental result.

Figure 4.10 tells us that the sparsity of $\mathbf{W}$ does have influence on friend recommendation accuracy. As p increases, the recommendation accuracy first keeps nearly unchanged, and then goes down quickly. The reason might be that as $\mathbf{W}$ becomes more dense, it lacks the ability to distinguish the most important features from others. As p doesn't influence much from 1 to 1.5, for simplicity reasons, in other parts of our experiments we fix p to be 1.

4.6 Conclusions and Further Works

In this work, we study the friend recommendation problem from the view of network correlation. A person has many different social roles on-line. For each social role, he/she makes different friends, and these different social roles form different social networks. To consider the effect of different social roles, we propose a network alignment method to find the correlations among different networks. The second aspect we take into account is the pairwise user similarity preservation to maintain the original data structure.

Experimental results by aligning tag and contact networks have shown that the proposed NC-based SFR outperforms other methods in friend recommendation: we achieve the highest precision in friend prediction. We found that a small number of features can align the tag network to contact network well and provide sufficient information for friend recommendation. Both network alignment and social network structure preservation play an important role in our task.

In future, we will further develop our algorithm in the following aspects: 1) In this chapter, we consider different social networks to have similar structures and we handle them using similar methods, And in experiments we align only two networks. We will extend the idea of network alignment to many networks, and consider the individual properties of these networks to make better recommendations. 2) We will apply the idea of network correlations for applications other than friend recommendation.

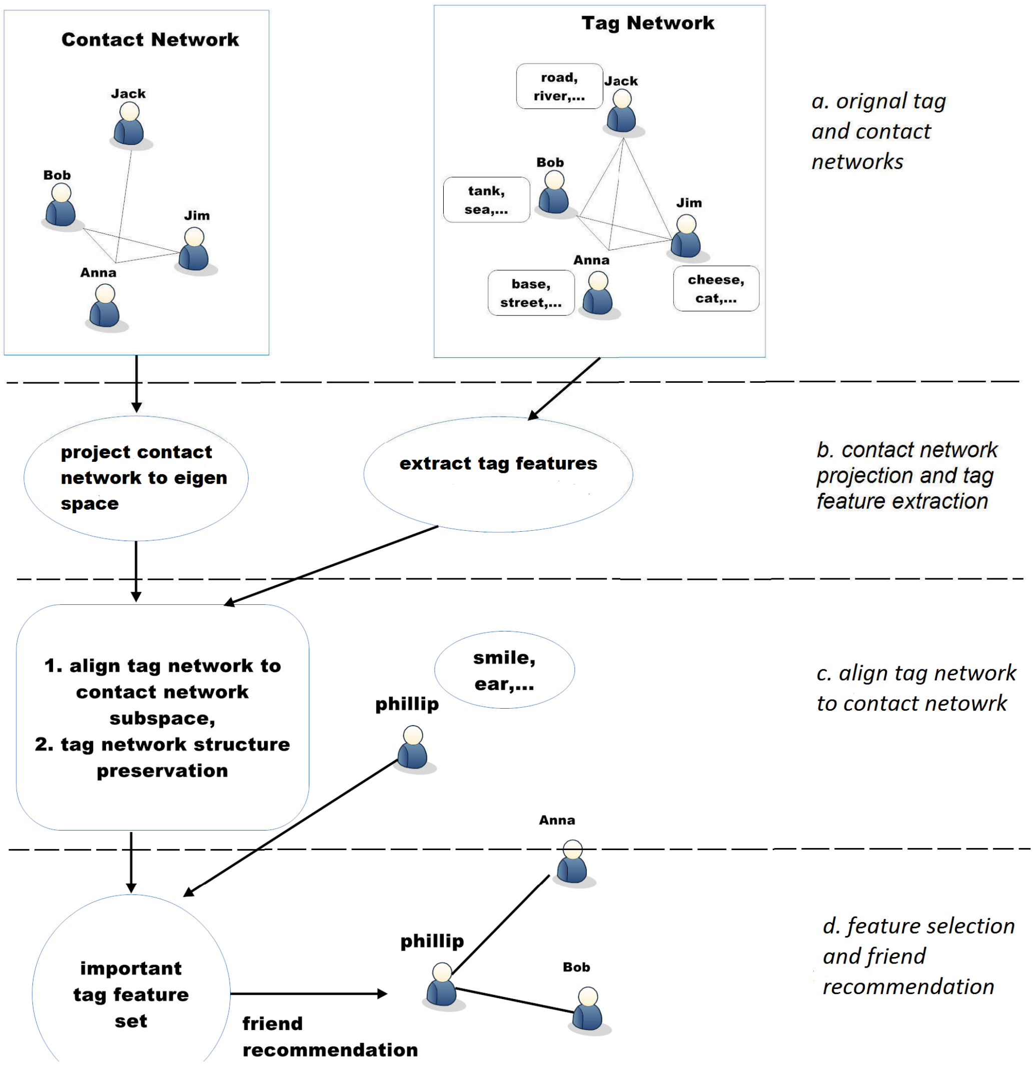


Figure 4.2: Framework for our algorithm: *a.* input original tag and contact network. *b.* project the contact network to its eigen-space, extract features from tag network. *c.* align tag network with contact network in their subspace through feature selection. *d.* new friends will be recommended based on selected important features

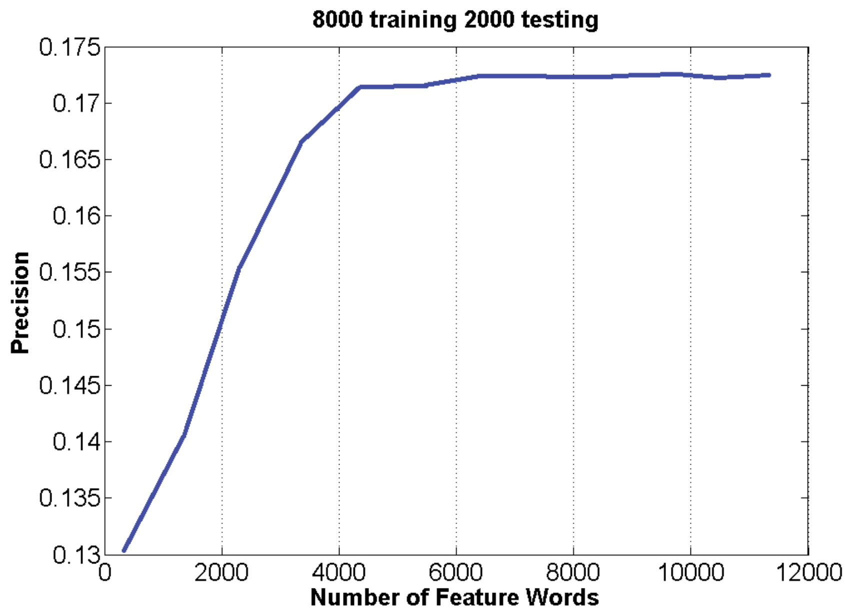


Figure 4.3: Precision@20 with the Increase of Features

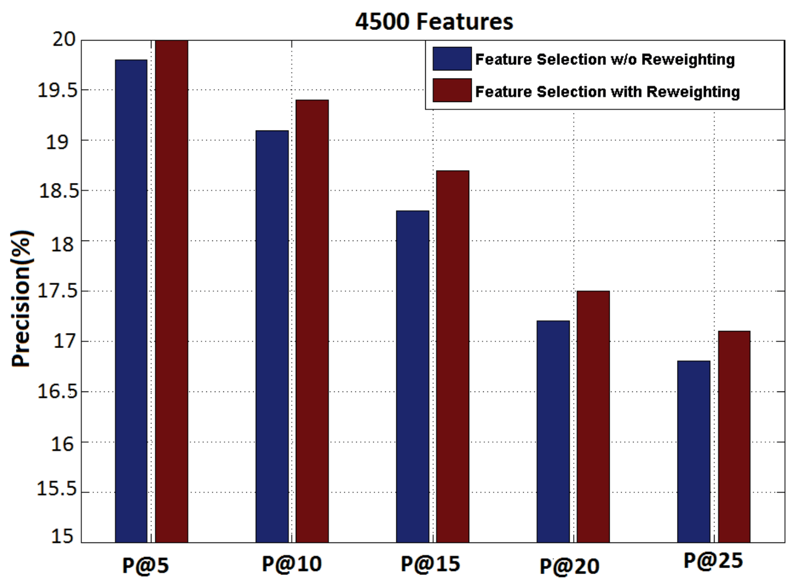


Figure 4.4: Feature Selection vs. Re-weighting

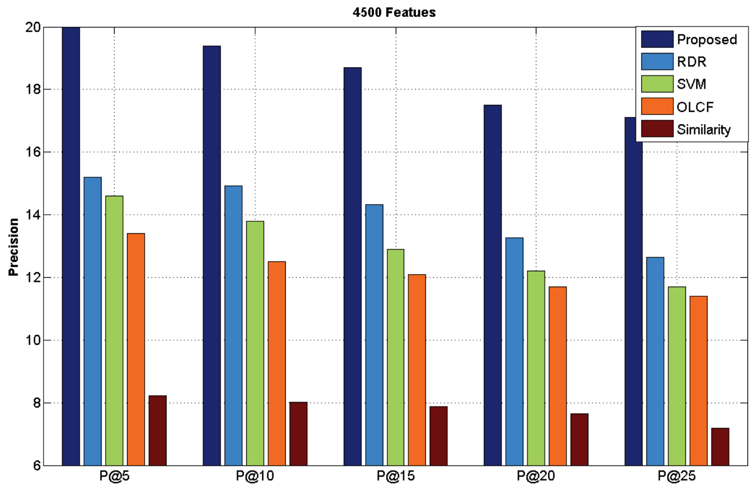


Figure 4.5: Precision with Different Algorithms for 4500 Features

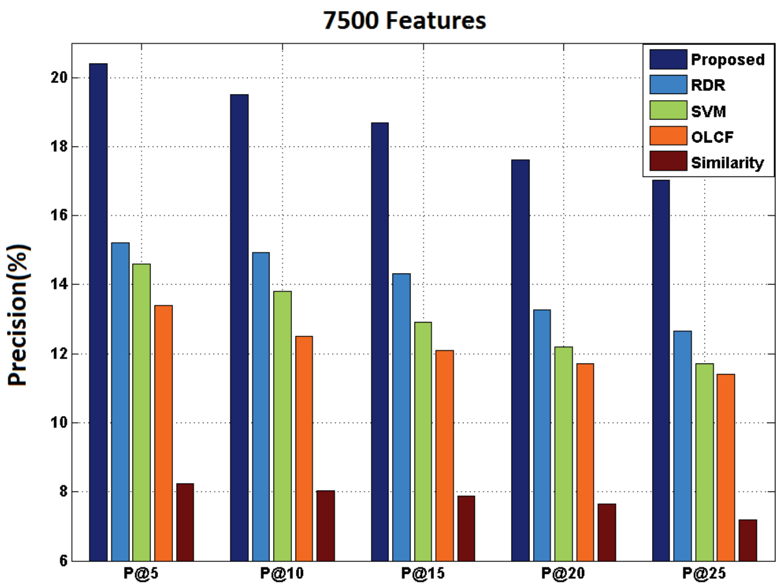


Figure 4.6: Precision with Different Algorithms for 7500 Features

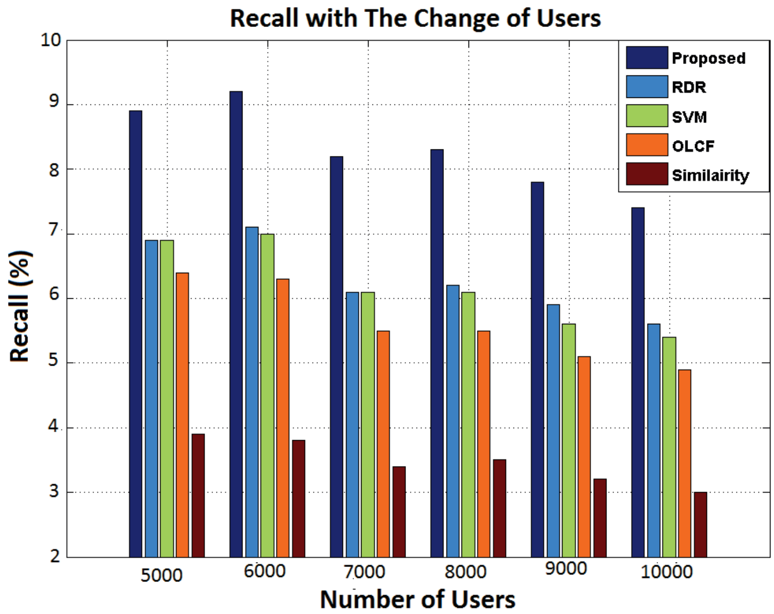


Figure 4.7: Recall with Different Algorithms

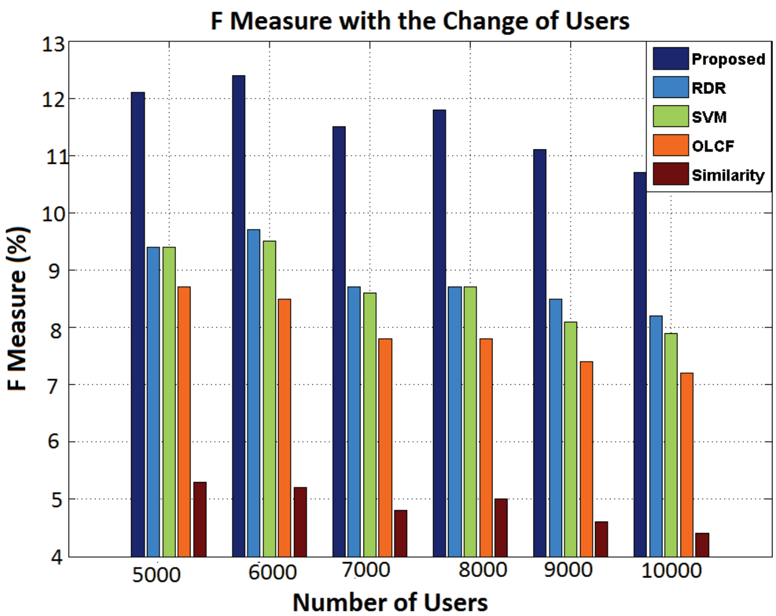


Figure 4.8: F-Measure with Different Algorithms

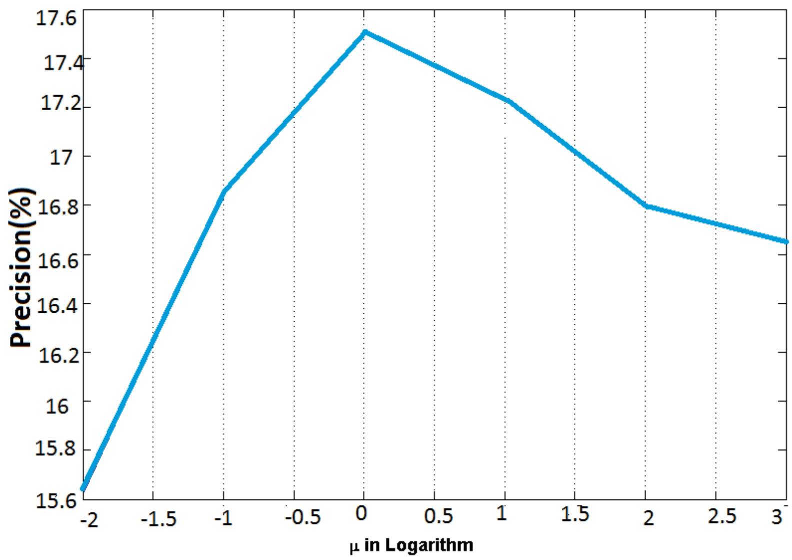


Figure 4.9: Effect of weights between network alignment and structure preservation

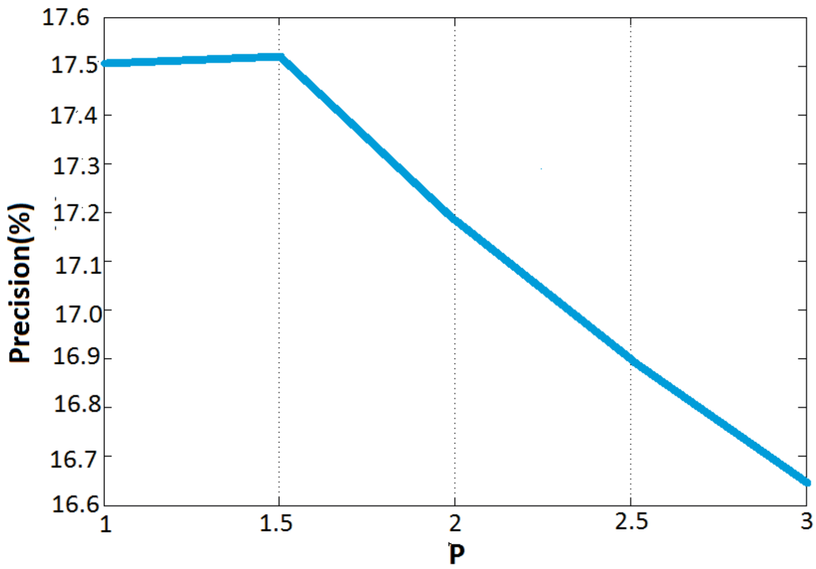


Figure 4.10: Effect of Sparsity of $\mathbf{W}$

Chapter 5

Two-Stage Friend Recommendation Based on Network Correlation and Feature Co-Clustering

The previous chapter utilises text information to recommend friend, and ignores other useful information such as images. On the other hand, image information contains much noise in the task of friend recommendation. To utilise both the text and image information efficiently , in this chapter we propose a two-stage procedure for more accurate friend recommendation: In the first stage, based on the relationship of different social networks, the Flickr tag network and contact network are aligned to generate a "possible friend list"; In the second stage, after a friend circle enlargement step, co-clustering method is applied to the tag and image information of the list to refine the recommendation result in the first stage. Experimental results show that the proposed method achieves good performance and every stage contributes to the recommendation.

5.1 Introduction

In Chapter 4 we proposes a friend recommendation method based on the alignment of different networks and the instructive feature selection. Though the method is to some extent effective, it considers only the text information into consideration. For a more precise recommendation, we treat the alignment in Chapter 4 as the baseline and make further improvement. In this chapter, we propose a staged friend recommendation scenario.

The main reason that we apply the multi-stage friend recommendation scenario lies in the complexity of multi-source information and the decision making behaviour of people. For example, an individual might make an on-line friend because they discuss a hard mathematical problem, or it is possible that he/she makes a friend because they both enjoy a film. The reason for friend making might be very diverse. It would be relatively difficult if we consider different factors together at the same time for recommendation. In our opinion, it is more convenient and clearer to analyse these factors step by step, rather than to deal with such cross-domain information as a whole. By untwisting the different factors in the recommendation procedure and analysing each factor in depth, a more precise recommendation performance is expected. As a consequence, we apply a two-stage framework to synthesise heterogeneous information from different domains. In this work, we concentrate on the widely-used image and image-related experience sharing website Flickr.

By alignment we obtain a “possible friend list”. However, an on-line social network is usually a very complex and sparse network that contains much noise, and the result of one-stage friend recommendation is usually not satisfactory. For example, both the recent work (Wan et al. 2013) and our previous experiments in 4.5 show that by applying one stage recommendation can only achieve a recommendation precision of less than 20%, which has little significance in practice. On Flickr, in addition to the tag information we considered in the previous alignment step, there is plenty of helpful information in the photos people have uploaded. The appearances of these photos are definitely very important for individual's decisions about making on-line friends.

On the other hand, the images that individuals' uploaded might contain much redundancy and noise for friend recommendation. For example, if we only know that two individuals enjoy some photos about natural scenery, it is hard to claim that they should be friends with each other. As a result, in this chapter, we design an extra stage to further raise friend recommendation accuracy utilising the image information. This stage is to refine the results from the first network alignment stage.

Another issue related to the first stage is that it only utilises tag information, which might lead to some omissions of true friends. To alleviate its negative effects, before the image-based refinement of the second stage, we design a "possible friend circle enlargement" step. The intuition of this step is based on the following assumption: the friends of user A's friends are also likely to be user A's friends. This is a common phenomenon in human society and many on-line society has applied the idea for friend recommendation. We apply it to include more "possible friends" for refinement.

Based on the possible friend list of user A, generated by the first stage and the enlargement step, we apply the image feature information in our algorithm and we co-cluster three sets: one user's possible friend list, all the tag features related to this list, and all the image features related to this list. After co-clustering, the entire possible friend list is divided into several clusters. This enables us to view each cluster as a group that interests user A. In the last step, we choose representative users in each cluster as user A's friends.

Co-clustering, or simultaneous clustering, deals with the high dimensional data. The goal is to find sub-matrices, which are subgroups of different dimensions that exhibit high correlations. (Charrad & Ahmed 2011). Two dimension, or two way co-clustering has been applied in the field of bioinformatics (Madeira & Oliveira 2004), web mining (Charrad, Lechevallier, Ahmed & Saporta 2009), text mining (Bichot 2010), and social network analysis (Hu, Pan, Long, Zhu, Jiang & Zhang 2016). higher dimensional co-clustering methods have been also developed for more complex cases. In this work we utilise a three way decomposition method to find the clusters in different dimensions.

A number of technologies other than co-clustering such as topic model (Blei, Ng & Jordan 2003*b*) may also find social groups and discover friendships. In this chapter, we apply the co-clustering method because it is conceptual more direct and the structure of clusters is similar to the communities found in on-line society. It is also a good method for simultaneously clustering heterogeneous yet correlated modalities, as in our case. The whole system is illustrated in figure 5.1:

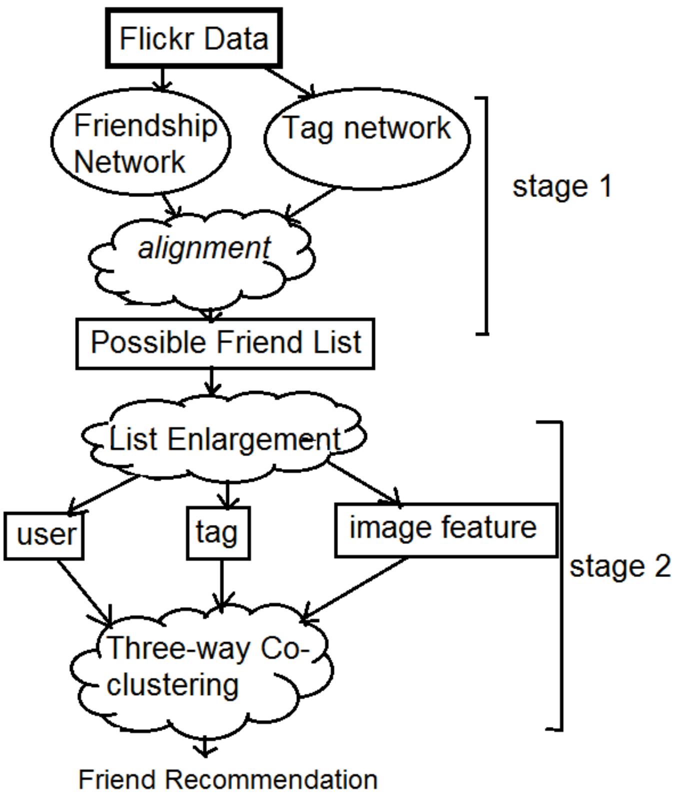


Figure 5.1: System Illustration

The two-stage framework has two main advantages: in the first place, it deals with the data from one domain in each step, thus reduces the complexity. In the second place, it can make a deep analysis about the contribution of the data from each domain, and help us to understand the problem more thoroughly.

In summary, this work makes the following contributions:

1. We propose to a two stage framework for friend recommendation, as well as a friend circle enlargement step between two stages. The second stage refines the result of the first stage.

2. A co-clustering procedure of user, tag and image feature is applied for friend recommendation.

3. We have conducted comprehensive experiments to show that the proposed method significantly improves the accuracy of friend-recommendation.

The rest of the chapter is organized as follows. Section 5.2 outlines related work. Section 5.3 gives our framework and system model, as well as the details for co-clustering. Section 5.4 shows the performance of our method and analysis is made according to the result. Lastly, Section 5.5 concludes our work.

5.2 Related Works

In this section we introduce several research fields that directly relate to our work: co-clustering and staged recommendation.

5.2.1 Co-clustering

Co-clustering is the process of clustering instances and their features at the same time. Early works such as (Dhillon, Mallela & Nodha 2003) deal with bipartite graph between documents and words which can be relaxed to a spectral clustering problem. There are several general approaches for the two way co-clustering. (Zha, He, Ding, Simon & Gu 2001) models the rows and columns of the data as a weighted bipartite graph and assigns weights to graph edges using similarity measure techniques. Then the bipartite graph is partitioned in a way that minimizes the cut of the partition. (Yang, Wang, Wang & Yu 2005) treats clusters as blocks in the matrix and minimizes the deviations of their elements. (Tang, Zhang, Ramanathan & Zhang 2001) uses one way clustering method and then combines the one dimensional clustering results to produce the final clusters. (Hochreiter, Bodenhofer, Heusel, Mayr, Mitterecker, Kasim, Khamiakova, Van Sanden, Lin, Talloen, Bijnens, Ghlmann, Shkedy & Clevert 2010) and (Dhillon et al. 2003) apply some kinds of probabilistic generative models to define the two-way clusters.

Later researches have extended the data to be high-order rather than bipartite graphs: (Bao, Min, Lu & Xu 2013) uses a star-structured K-partite graph to illustrate multi-modality data that has relationships in different domains, and co-clusters these data by reducing the mutual information. (Zhou, Xu & Zong 2012) introduces tensor decomposition for co-clustering multi-modal internet data. (Zhao & Zaki 2005) develops a method to mine the coherent clusters in three-dimensional gene expression datasets that relies on graph-based approach. None of the work cited above has taken sparsity into account, yet social media data is usually sparse. (Parpalexakis, Sidiropoulous & Bro 2009) provides a good solution for co-clustering sparse data that decomposes three- or higher-way data into sparse indicator vectors, which is based on PARAFAC decomposition (Bro 1997). In our work we apply the idea to deal with image and text data in the field of multimedia.

Multi-Stage Learning and Recommendation

Staged learning has been widely applied in different fields. To name a few, it has been applied to disease detection (Mosquera-Lopez, Agaian & Velez-Hoyos 2014), saliency detection (Han, Zhang, Wen, Guo, Liu & Li 2016), pedestrian detection (Zeng, Ouyang & Wang 2013), artificial intelligence gaming (Teng, Tan, Starzyk, Tan & Teow 2014) and behaviours (Luo, Ding, Cao & Wu 2014), and social response prediction (Chen, Ting, Shen, Hu & Liang 2015), etc. Generally, when we deal with a relatively complex task, it is natural to separate the whole task into several stages so that each stage is easier to handle. For different tasks the division of the stages are also different. For example, to check the prostate cancer, the first stage of (Mosquera-Lopez et al. 2014) applies a on-shot classification for a rough check, then several SVM are used in the second stage for result refinement. In another case of learning the humanoid robot stand-up behaviour, (Luo et al. 2014) proposes a two stage framework that in the first stage, a model is initialized from the human motion capture data, and the solution space can be pruned. In the second stage, some fast and active searching methods are developed find the solution in the previously pruned space. To predict the social response of a new message in the social network, (Chen, Ting, Shen, Hu & Liang 2015) also develops a two stage method that in the first stage, unsupervised k-means clustering is applied to partition the message space. In the second stage, a classifier is trained in each cluster for precise prediction. As is shown above, most of the researches apply the second stage as a refinement step.

Staged method can also applied for recommendation system. Existing multi-stage recommendations are usually applied to find some patterns of users or items. For example, in (Li, Lee, Chen & Cheng 2009), a two-stage mobile recommendation is proposed to help users find the correct events.

The first stage clusters people according to their profile similarity and the second stage discovers the event-participating pattern. (Zhao & Wang 2015) designs the first stage to find some related resources that one user requires, and the second stage is used to find some patterns that the user might prefer from the previous stage for further recommendation. Both (Li, Lee, Chen & Cheng 2009) and (Zhao & Wang 2015) can handle the cold-start problem well but do not consider much about the cross-domain problem. In this chapter, we apply a two stage

5.3 System Model and Proposed Two-Stage Recommendation

In this section we present our framework in two stages. The first stage is a feature selection stage based on the network alignment from tag similarity network to contact network, as mentioned in Chapter 4. Then, after a friend circle enlargement step, a co-clustering method is applied in the second stage to refine the recommendation result.

5.3.1 Problem Statement and Solution Overview

As shown in Figure 5.1, our system has two main stages: network alignment and co-clustering. These two stages ensure that we obtain a more accurate recommendation. In the first stage we align the tag similarity network to the contact network via important tag feature selection, the details of network alignment are similar with Chapter 4. In this stage we obtain a "possible friend list".

There are two problems for this "possible friend list". First, the ample image features, which may carry much important information about personal tastes, are not utilized in the system. Second, all users use the same set of important features. As a result, it may fail to take individual's personality into account, which is actually very important in a social network, and might lead to some omissions of friends. We therefore design a friend circle enlargement step following a co-clustering step to address the problems..

5.3.2 Friend Circle Enlargement

According to the assumption that “a friend’s friends also tend to be friends”, we add possible friends to the list. For example, as shown in Figure 5.2, in the first stage we get 20 “possible friends” of user A. By adding the friends of these 20 “possible friends” into the list we have about 1000 friends in total, To consider the personality of user A, we also add 500 similar users of A based on the “unimportant features”, and assume that these features show some personalities of each individual. We then crawled all the images, and tags uploaded by these 1500 users.

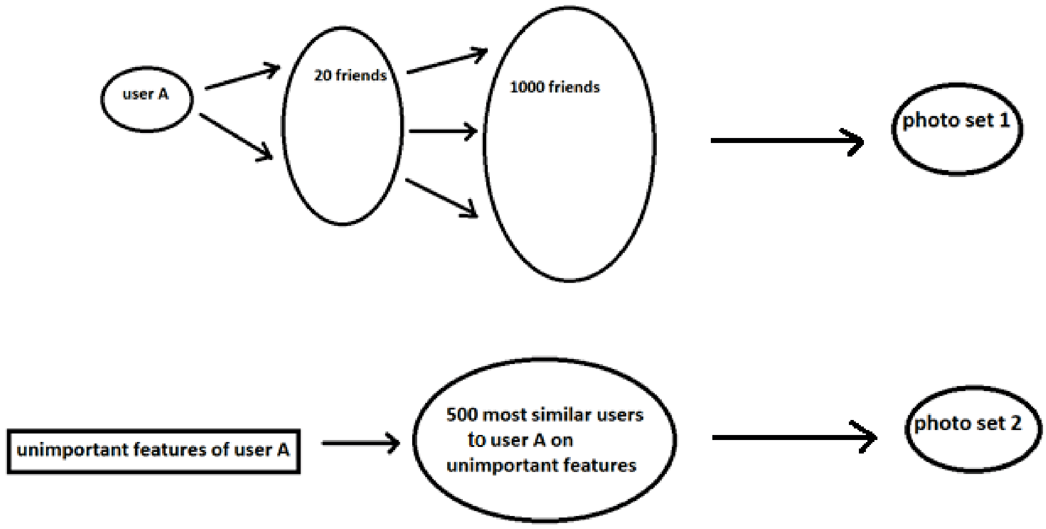


Figure 5.2: Friend List Extension

Now we have three sets: the user set, the tag set, and the image feature set. The three sets have links with each other; if a photo uploaded by user B contains one tag $\mathbf{t}_1$ and one image feature $\mathbf{i_1}$, then the node B, $\mathbf{t}_1$, and $\mathbf{i_1}$ in three different sets have links with each other. With the link information of the three domains, we can apply the co-clustering method as our second stage.

5.3.3 Three-Way Co-clustering

After obtaining all the photos and tags of the users on the list, we can conduct the three-way clustering to obtain topics that might interest user A. Because the number of image features is very large, (the number of photos for 1500 users can be hundreds of thousands, and the number of image features such as SIFT can be tens of millions), we first cluster the image features and then build a structure such as inverted index table for fast search.

Feature Clustering and Inverted Index Table

We give only a brief summary of the feature clustering and inverted index table, as this is not the main contribution of this chapter. We use the K-means method to form approximately 20,000 feature cluster centres. These cluster centres are saved in an inverted index table. When a new feature arrives, we can find which feature centre the new coming feature belongs to with the help of the inverted index table. Each feature is represented by its feature centre, and a single image is recorded by hundreds of feature centres. The illustration of the feature clustering process is given in Figure 5.3. The upper part of Fig. 5.3 shows how to build the inverted index table and the lower part shows how to quickly find the cluster to which the feature of a new photo belongs when it arrives. In Table 5.1, the structure of an inverted index table is given.

It is a interesting question that if there are some semantic meanings of the image feature centres generated by the clustering. Some discussions have been made about this topic (Rege, Dong & Fotouhi 2006). We might study further about this interesting topic in our future works.

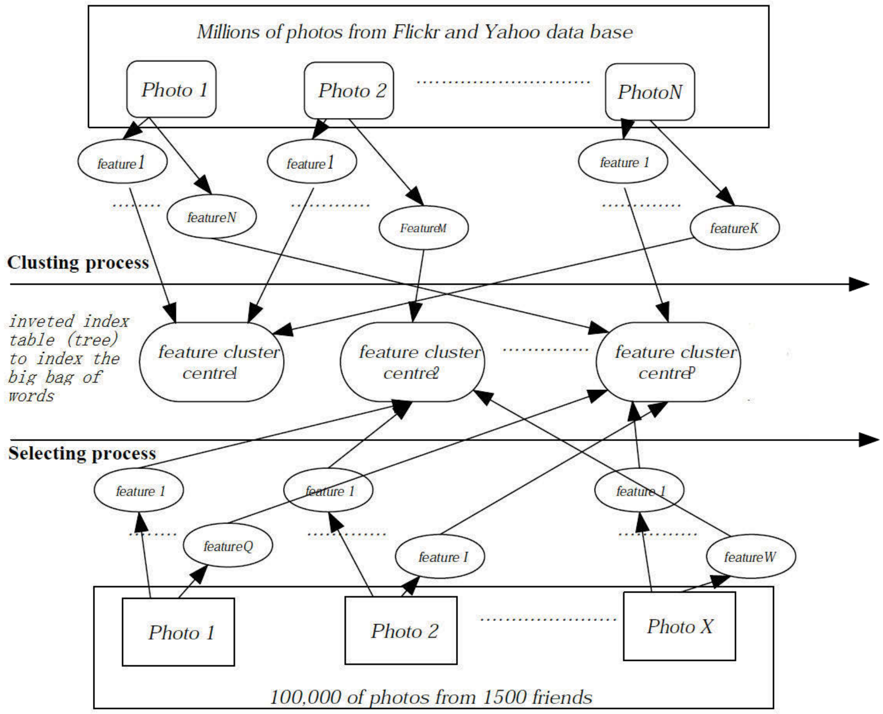


Figure 5.3: Feature Clustering

Table 5.1: Inverted Index Table

image 1	feature cluster centre 1, 3, 5,...
image 2	feature cluster centre 1, 5, 6,...
...	...
image 100,000	feature cluster centre 2, 6, 9,...

Three-way Co-Clustering

Now we have three datasets from different domains: the user set, the tag set, and the image feature set. The nodes in each dataset have links with the nodes in the other datasets. We co-cluster the data from the three datasets to find the groups that user A may be interested in.

The relationship of these three datasets can be expressed as a three-way matrix $\underline{\mathbf{X}}$. As an example, if user A uploads an image, this image has an image feature i, and this image contains a tag t, then the element $\underline{\mathbf{X}}_{Ait}$ should be 1, otherwise 0.

According to (Parpalexakis et al. 2009), a three-way array $\underline{\mathbf{X}}$ has a low rank approximation as the sum of the product of three vectors as follows:

$$\underline{\mathbf{X}} \cong \sum_{k=1}^{K} \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k \tag{5.1}$$

where $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is a rank-one three-way array with (i,j,n)-th element of $\mathbf{a}_i \mathbf{b}_j \mathbf{c}_n$. Let $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_K]$, $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_K]$, $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_K]$, and we express the three-dimensional $I \times J \times N$ array $\underline{\mathbf{X}}$ with three two-dimensional matrix as:

$$\mathbf{X}_{(1)} = \begin{bmatrix} \underline{\mathbf{X}}(:,1,:)^T \\ \vdots \\ \underline{\mathbf{X}}(:,J,:)^T \end{bmatrix}, \mathbf{X}_{(2)} = \begin{bmatrix} \underline{\mathbf{X}}(:,:,1)^T \\ \vdots \\ \underline{\mathbf{X}}(:,:,N)^T \end{bmatrix}, \mathbf{X}_{(3)} = \begin{bmatrix} \underline{\mathbf{X}}(1,:,:)^T \\ \vdots \\ \underline{\mathbf{X}}(I,:,:)^T \end{bmatrix} \tag{5.2}$$

and $\mathbf{X}_i$ can be expressed as:

$$\mathbf{X}_{(1)} = (\mathbf{B} \odot \mathbf{C})\mathbf{A}^T, \mathbf{X}_{(2)} = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T, \mathbf{X}_{(3)} = (\mathbf{A} \odot \mathbf{B})\mathbf{C}^T \tag{5.3}$$

where $\odot$ stands for the Khatri-Rao product. To co-cluster $\underline{\mathbf{X}}$ into K clusters, we can make $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ as indicator matrix as follows:

$$\min_{\{\mathbf{a}_k,\mathbf{b}_k,\mathbf{c}_k\}_{k=1}^K} \|\underline{\mathbf{X}} - \sum_{k=1}^{K} \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k\|^2 \tag{5.4a}$$

$$\mathbf{a}_k, \mathbf{b}_k, \mathbf{c}_k \in \{0,1\} \tag{5.4b}$$

$$\sum_{k=1}^{K} \mathbf{A}(l,k) = 1, \sum_{k=1}^{K} \mathbf{B}(l,k) = 1, \sum_{k=1}^{K} \mathbf{C}(l,k) = 1 \tag{5.4c}$$

(5.4b) and (5.4c) make the constraints that each node belongs to one and only one cluster. For computational efficiency (5.4b) and (5.4c) are often replaced by the sparsity constraints (Tibshirani 1994) as follows:

$$\begin{aligned} &\min_{\{\mathbf{a}_k,\mathbf{b}_k,\mathbf{c}_k\}_{k=1}^K} \|\underline{\mathbf{X}} - \sum_{k=1}^{K} \mathbf{a}_k \circ \mathbf{b}_k \circ \mathbf{c}_k\|^2 \\ &+\lambda_a \textstyle\sum_k \|\mathbf{a}_k\|_1 + \lambda_b \sum_k \|\mathbf{b}_k\|_1 + \lambda_c \sum_k \|\mathbf{c}_k\|_1 \end{aligned} \tag{5.5}$$

where $\lambda_{a,b,c}$ in (5.5) control the sparsity of each indicator vector. Following (Parpalexakis et al. 2009), this provides a computationally efficient algorithm for solving (5.5).

5.3.4 Friend Recommendation

Following three-way co-clustering, we simultaneously obtain three kinds of clusters of users, their uploaded tags and their image features. We can interpret each cluster of users as one group that might interest user A. User A's friends should come from these groups. There are different strategies for choosing friends from each of the groups. We list several below and will give the experimental results later to see which strategies work better:

1.(S1)Choose users that are near the centre of each cluster.

2.(S2)Choose users that are most active in each cluster. A user's "activeness" is measured by the following aspects: his/her communication with others, the number of photos uploaded, and the popularity of his/her photos.

3.(S3)Combine the above two criteria by averaging their ranks.

When new users come they will first upload a number of tags and photos. With these tags and the image features of photos, we can make accurate friend recommendations.

A potential problem for our algorithm is that it takes time to extract the image features for a new user and do the co-clustering, thus it is not appropriate for on-line recommendation. However, with some modifications we can make it operable in real-time. We can build up a large dataset and co-cluster its users,tags,images, and then build a large inverted index table as mentioned in 5.3.3 to quickly find those clusters to which the user belongs.

The whole algorithm is given in Algorithm 5.1.

5.4 Experiments

In this section, we conduct extensive experiments to show the effectiveness of our proposed method, and also illustrate several interesting properties. First, we briefly introduce of our social media dataset, and then we discuss our algorithm from different aspects.

Algorithm 5.1 Proposed Algorithm

Require:

tag feature matrix **S**, contact matrix **K**, image feature matrix **I**, tag feature vector **f** and image feature matrix **i** of the new user, number of recommending friends T

Ensure:

Friend recommendation list

Stage One

Calculate the tag relationship matrix **L** from **S**

Determine λ, μ via cross validation on training set

Calculate **V** by eigen-decomposition of Laplacian of **K**

Iteratively get **W** according to (Liu et al. 2014*b*).

Choose tag features by ranking the norm of rows of **W**.

Choose "possible friend list" on important features.

Stage Two

Enlarge "possible friend list" according to 5.3.2.

Extract image features of users in the list.

Co-cluster user, tag, and image feature set (Parpalexakis et al. 2009).

Recommending friends by choosing users from each cluster.

5.4.1 Experimental Settings

Dataset

We crawled a social network from the big image sharing site Flickr. As the data set is quite large, a relatively unbiased dataset was obtained. In total we crawled the data of 10000 users, and for each user, we crawled all their photos, and tags of each photo. The SIFT features for all photos were extracted. In this chapter we do not concentrate on what kind of image features are more indicative for matching images, so we choose the popular SIFT feature that has been widely in image retrieval problems. In future works, we may also study the properties of different features for more precise result. We then crawled the user contact information to form the contact network. Contact information in Flickr is acquired by when a user adds another user to his/her friend list, or vice versa. We crawled all the contacts between any two users in our dataset. A short summary of our dataset is given in Table 5.2 :

Table 5.2: Dataset Statistics

Users	10000
Photos	543,754 photos from 10000 users
Image Features	46,443,754 SIFT features
Contact	145,684 friend links among users
Tags	35,574 words after filtering

Settings and Metrics

Our task is to make precise contact information prediction,so that way when a new user enters into the social network, we recommend new friends according to key words and photos that represent the user's interests..

We use the method summarized in 5.3.4 to recommend friends to new users. We use the recommendation precision metrics to show the effectiveness of the proposed algorithm. In our experiment, precision is defined as the number of correctly recommended friends divided by all the recommended users.

We use some reference methods as mentioned in Chapter 4, they are: pure similarity, OLCF (Rendle & Thieme 2008), and RDR (Jiang et al. 2012).

The Second Stage: Co-clustering

With the "possible friend list" generated in Stage One, we can start the co-clustering process for Stage Two. This is a refinement process which selects three criteria for friend recommendation as stated in 5.3.4. Figure 5.4 illustrates one cluster from the co-clustering result. For convenience, we use a simple voting strategy in this illustration to determine which cluster an image should belongs to, instead of using SIFT features. We can see that most of the conceptually similar tags and images are clustered together.

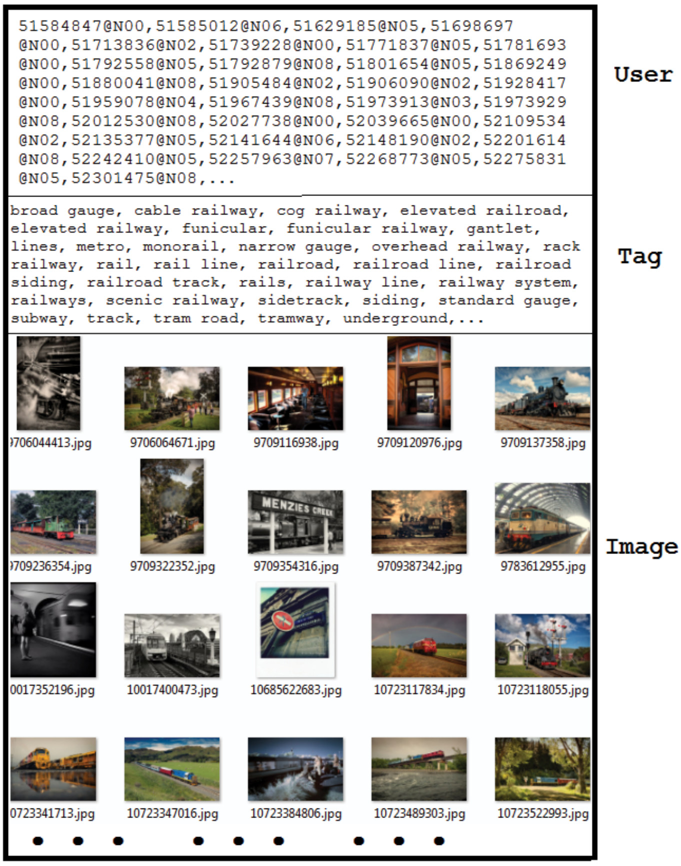


Figure 5.4: Result of Co-clustering

Most Flickr users are unlikely to upload photos that illustrate too many aspects of their interest, therefore the number of clusters in Stage Two should not be too large. We set K in 5.3.3 to be 5,10, and 20, and choose 4, 2, 1 users as the final recommended friends from each cluster. We take 1000 users in total and make recommendations for them. The tag feature number of the first stage is fixed at 4500. The results of the three criteria are shown in Figure 5.5.

From Figure 5.5, we see that the combination of near-centre users and most active users have slightly better performance. We achieve the best performance when we choose 10 as the number of clusters, but the difference is not great. Stage Two increases recommendation precision by about 8%, compared with Stage One. This illustrates the effectiveness of our assump-

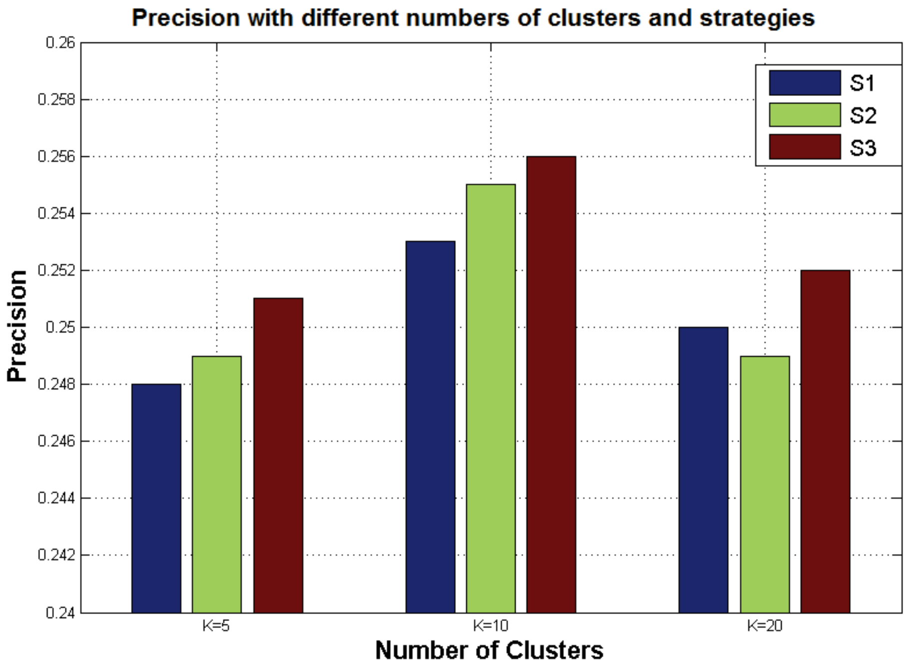


Figure 5.5: Stage Two: Precision@20 with different numbers of clusters and different strategies

tion that "a friend's friends also tend to be friends". Figure 5.6 shows the improvement of Stage Two compared with Stage One with the change of tag features in Stage One.

Comparison with Reference Methods

Next we compare our proposed methods with the reference methods. For the proposed method, we use the strategy of combining the near-centre and most active users. For $P@5$,$P@10$, $P@15$ and $P@20$, we choose the co-clustering parameter of $K = 5, 10, 15, 20$ in Stage Two, and choose one friend from each cluster. The result is shown in Figure 5.7.

Figure 5.7 shows that the proposed algorithm outperforms all the reference methods in the recommendation precision by at least 15%.

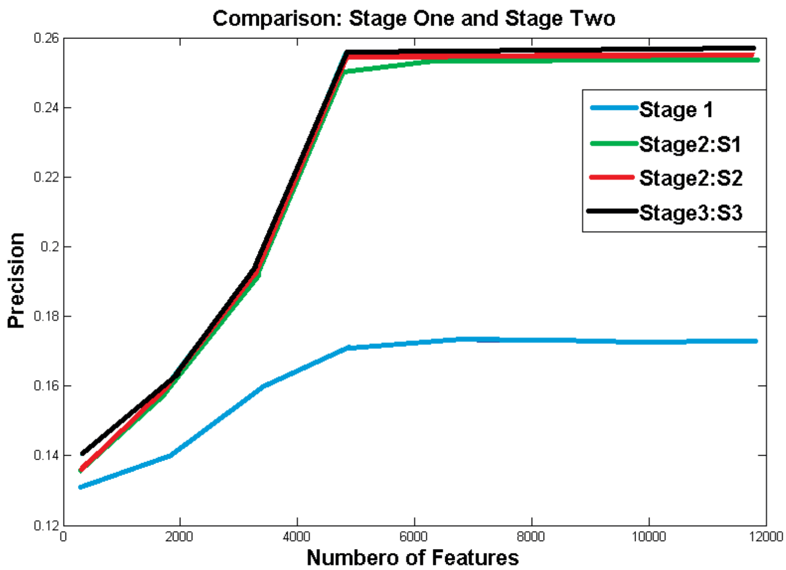


Figure 5.6: Comparison of Stage One and Two: Precision@20 with Increase in the Number of Features

5.5 Conclusion and Future Work

In this chapter, we develop a two-stage friend recommendation scenario utilizing multimedia information. In the first stage, tag information is utilized to build a tag network and is aligned to a contact network by a number of important features, to generate a "possible friend list". In the second stage, a co-clustering procedure is proposed to co-cluster user, tag, and image features to form groups and make more precise recommendations.

Experimental results show that the proposed method outperforms other methods in friend recommendation, with our method achieving the highest precision in friend prediction. The network alignment of the first stage is effective and proves that a small number of features is sufficient for friend recommendation. Co-clustering in the second stage shows that conceptually similar tags and images can be successfully clustered, improving the recommendation result has been improved by about 8%.

In the future, we will further develop our algorithm in the following as-

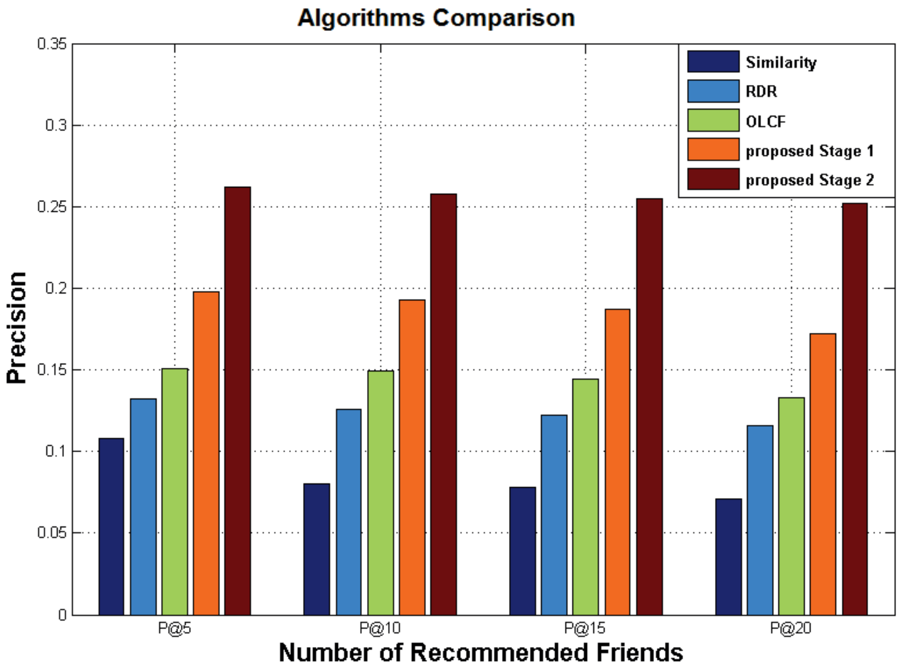


Figure 5.7: Stage Two: Precision@20 of different methods

pects. 1) We will extend our ideas to further applications such as product recommendation, media retrieval, etc., and 2) We will develop other algorithms in each of our two stages to achieve better, more acceptable recommendations.

Chapter 6

Two-Stage Friend Recommendation Based on Network Alignment and Series-Expansion of Probabilistic Topic Model

Precise friend recommendation is an important problem in social media. In Chapter 5 we apply a two stage framework to recommend friends , and in the second stage we apply a co-clustering method to refine the results of the first stage. Though the co-clustering method is conceptually easy to understand and the performance is improved, it lacks the ability to tell how close two individuals are and thus can not rank the friends according to their intimacy. In this chapter, with the relationship between image features and users we build a topic model to further refine the recommendation results. Because some traditional methods such as variational inference and Gibbs sampling have their limitations in dealing with our problem, we develop a novel method to find out the solution of the topic model based on series expansion. We conduct experiments on the Flickr dataset to show that the proposed algorithm recommends friends more precisely and faster than traditional methods.

6.1 Introduction

In the previous chapter, a staged method is applied and has shown its effectiveness. However, the co-clustering method is originally not very good method for our task —— friend recommendation. For it lacks the ability to distinguish the closeness in one cluster, which leads to some issues in the final friend recommendation step. To overcome this weakness, we propose a probabilistic topic model based method that can accurately measure the intimacy between any of the two users.

In the first stage, similar to Chapter 4, based on the correlation of different networks, we align the tag-similarity network to friend network to obtain a possible friend list. Specifically, we consider each user as one node in a graph, and we crawl the uploaded tags from each user and calculate the tag similarity between any two users as the edges to form a tag-similarity network. On the other hand, we also obtain the friendship information in Flickr, and if two users have friendship with each other, we add an edge between the two to form a contact network. In this way we build a tag-similarity network and a contact network that have the same nodes but different topologies. Because the tag-similarity network and contact network on Flickr are related to each other, we dig their correlation by choosing important tag features, to make the tag-similarity network more similar to the contact network. In this way, the chosen tag features provide a guideline for friend recommendation. This stage makes a mass election of possible friends.

In the second stage, to overcome the problem that the mass election considering only the tag information might not be precise, we build a topic model to illustrate the relationship between user's friend making behaviour and the image features they have uploaded. This stage refines the list obtained in the first stage. The main reason for applying a topic model in our second stage lies in the fact that the topic model has the ability to tell on what probability a user would prefer a photo/item/friends.

The probabilistic topic model discovers the abstract "topics" that occur in a collection of documents/datasets, and it has been widely used in recommendation systems (Min et al. 2015)(Zheng, Song & Bao 2015)(Song, Zhang & Cao 2013). By introducing some latent variables and applying Bayesian rule, it is conceptually easy to combine information from different domains and make specific recommendations (Min et al. 2015)(Song et al. 2013). Generally it assumes that people's various behaviours such as shopping, posting and friend making are controlled by some latent topics. Certain people have particular bias on different latent topics. For an individual that acts differently in different domains, his/her latent interest topic might be similar. For example, a user who posts many different photos about food on Flickr might have higher probability to be interested in the topic of cooking, and thus it is reasonable to recommend some kitchenware to him/her on Amazon. Furthermore, the topic model provides a relatively precise probability to show to what extent an individual is interested in a topic, and thus makes it easy for further recommendation.

However, it is often not easy to find the solution of a topic model when different domains are concerned, for it involves the integrals of several coupled random variables, which is a complicated mathematical problem in general (Blei, Ng & Jordan 2003*c*). Two methods are widely used to deal with this problem: Gibbs sampling (Griffiths & Steyvers 2002) and variational inference (Blei et al. 2003*c*), or the combination of the two (Welling & Teh 2008). Although applied successfully in many cases, both of them have some disadvantages: for Gibbs sampling, it is inefficient for large count values since it requires averaging over many samples to reduce variance; for variational inference, though it is efficient to deal with large scale data, the variational step makes it hard to control the precision when approximating the integrals when making the Bayesian inference. In this chapter, with the help of Mellin and inverse Mellin transform, we propose a novel way based on series expansion to calculate the coupled integrals that are required in the Bayesian inference.

Matrix factorization (MF) method can be also applied to deal with the cross domain recommendation problems (Ma, Yang, Lyu & King 2008*b*)(Guo, Zhang & Yorke-Smith 2015). It decomposes different social networks into latent vectors to find the important factors that influence individuals' social behaviours, and make recommendations based on these latent factors. However, it lacks a mechanism to draw the complete distributions of the whole social network, and thus might lead to some local optimum. Our proposed method provides a way to describe the whole distribution of the social network, to perform a better recommendation.

To sum up, we build a two-stage friend recommendation system based on text and image data: in the first stage, we apply tag-user information to get a possible friend list, and in the second stage we refine the list by utilizing the image-user information. Our main contributions are as follows: Firstly, we build a topic model to analyse the relationship of the data from different domains. Secondly, we propose a novel method based on the study of the distribution of algebra of random variables to find a solution of the model. The solution is given in a series expansion form, and can lead to more precise solutions of the model. We also make comprehensive experiments to show the effectiveness of our method.

The rest of Chapter 6 is organised as follows: Section 6.2 outlines related work. Section 6.3 introduces our system framework. Section 6.4 gives the detailed explanation of our series expansion method. Section 6.5 evaluates the performance of our method and some analysis is made according to the results. Lastly, Section 6.6 concludes our work.

6.2 Related Work

Our work in this chapter is mainly related to the following research fields: friend and cross domain recommendation, topic model, and algebra of random variables.

Cross-Domain Recommendation

Individuals' decision of making friends are often multi-dimensional. As a result, recently many researchers consider friend recommendation based on cross-domain information. (Chin, Xu & Wang 2013) considers the friend recommendation problem at working places and conferences, by utilising both users' temporal location as well as their common friend information. (Zeng & Chen 2013) combines three aspects of each user's information: the items one likes, the friends one has, and the groups one belongs to. Such information of different aspects is synthesised and integrated into one cost function. By optimizing the cost function the heterogeneous data are fused for item, group and friend recommendations. In (Yan, Sang, Mei & Xu 2013), individuals that have both accounts in Flickr and Twitter are collected to build the relationship of the two social websites. The common behaviours of each user in Flickr and Twitter are analysed and the friend recommendation of the two domain is made based on these common behaviours. (Guo, Tian & Mei 2014) divides the different data in Flickr into two classes: interaction data(comments, making favorite photos) and similarity data(common friends, groups, tags, geo, visual), and applies these two classes of data comprehensively to estimate the strength of the ties between users.

For the works listed above, the data from different domains are processed simultaneously or fused together to get the final recommendation result. On the one hand, the above methods take the advantage that data from different domains might be related to each other; On the other hand, these methods combine the cross-domain information in one step ((Yan et al. 2013)) or synthesise it in one cost function ((Zeng & Chen 2013)), thus usually can not give a good explanation of how the data from a specific domain contribute to the final recommendation result, and the twisted data from different domains often makes the problem more complex. To have a better understanding of the effectiveness of the data from each domain, in this chapter, we design a two-step recommendation that in each step we utilise the data from one domain.

6.2.1 Probabilistic Topic Model

In the second stage of our method, a topic model is applied to get a more precise recommendation.

Topic Model in Recommendation

The probabilistic topic model is a successful approach solving the problem for information retrieval (Blei et al. 2003*c*) and recommendation (Min et al. 2015)(Zheng et al. 2015)(Song et al. 2013). For example, (Zheng et al. 2015) recommends temporary friends to users by building models that contain latent variables that illustrate users' interests change with time.

By assuming some latent factors it is conceptually easy to build the relationships among different domains. (Min et al. 2015) designs a model that connects the Flickr and Foursquare data for image, topic and item recommendation. It assumes that both domains have some common latent factors and each domain also has its own latent factors, and the users' activities on these two platforms are the synergism of all these factors. Gibbs sampling is applied to find the value of the latent factors. (Song et al. 2013) considers the friendships and the votings on the large Film rating website. To predict individual's flavour about films his/her social relationships and scores of films are combined with some latent factors. Variational methods are applied to solve the model.

To make the model to illustrate the situation of the real world more accurately and reasonably, both (Min et al. 2015) and (Song et al. 2013) make many assumptions of the latent topic and thus contain many unknown parameters to infer: (Song et al. 2013) contains more than 10 unknown parameters and (Min et al. 2015) has more than 30. The presence of so many unknown variables not only greatly increases the complexity of the algorithm, but also leads to other problems such as over-fitting or redundancies. Different from the above works, we build the model in a more compact manner.

Gibbs Sampling, Variational Inference and Matrix Factorization

Due to the coupling of latent variables, the direct inference is usually impossible for a specific topic model. Generally there are two methods to find a solution for topic model: Gibbs sampling (Geman & Geman 1984) and variational inference (Blei et al. 2003*c*). For some complex multivariate probability distributions, to determine the parameters of the distribution, direct sampling is difficult. Gibbs sampling samples the marginal distribution of one variable each time, and iteratively samples all the marginal distributions. The variational method, on the other hand, approaches the solution by approximating the original complex distribution with a factorised one, which is easier to handle.

As stated in section 6.1, both of the two methods have some weaknesses: Gibbs sampling has difficulties in handling big data problems, and the variational method can not determine if the approximation is close to the original one. Some researchers consider combining the two in one problem: In (Welling & Teh 2008), small counts of data are sampled and the variational method is applied to update large counts, which improves the performance on the large dataset. However, how accurate the approximation of variational method is not yet discussed in (Welling & Teh 2008). In this chapter, we propose a new solution to a topic model by directly calculating the distribution of the latent variables.

Compared with the above two methods, MF-based method also assumes some latent variables but instead of determining the marginal distribution of the observed data, it factorizes the observed data into different latent factors, which leads to some computational convenience and efficiency. Both of (Ma et al. 2008*b*) and (Guo et al. 2015) utilize user friendship network and user-item network and obtain some latent factors that show the preference of individuals. The recommendations based on these latent factors are relatively effective. On the other hand, they do not try to find the probabilistic distribution of the network and all of these methods apply some gradient descent methods, that are relatively easy to be trapped into a local optimum.

Our method avoids this drawback by deducing the distribution of the whole probabilistic model.

6.2.2 Algebra of Random variables

The essential problem of our approach in this chapter is to get the exact mathematical expression of the coupling of different random variables, mainly the sum and product of random variables. These problems were extensively discussed in the 1950s to 1970s year, last century, during which time the random process was a hot research topic but the computer simulation technology was not well developed. In (Carter & Springer 1977)(Springer & Thompson 1977)(Pruett 1972), the products of typical distributions such as Beta, Gamma and Rayleigh are discussed. Most of these works utilise the Mellin transform (Michiel 2001) as the essential tool for deducing. (Springer 1979) gives a good summary of these works and also discusses the distribution of the sum of random variables. The algebra of random variables has also been studied recently in certain fields such as wireless communication in (Ahmed, Yang & Hanzo 2011) and (Mallik & Sagias 2011). These works show that the product and quotient of random variables with certain distributions can be expressed analytically. We will mainly apply some of the results in (Pruett 1972)(Springer 1979) later in our work. As Gaussian distribution has some good properties(its domain of definition is all the real values, and has a central point, etc.) we assume that our latent variables to be Gaussian distributed.

6.3 System Model

The main framework of our model is shown in Figure 6.1, which contains two stages: In the first stage, network alignment is applied to generate a possible friend list, by correlating the tag and contact data in Flickr; In the second stage, the user-uploaded image features generate some topics by utilising a probabilistic topic model, and a new method is developed to solve the model for precise friend recommendation.

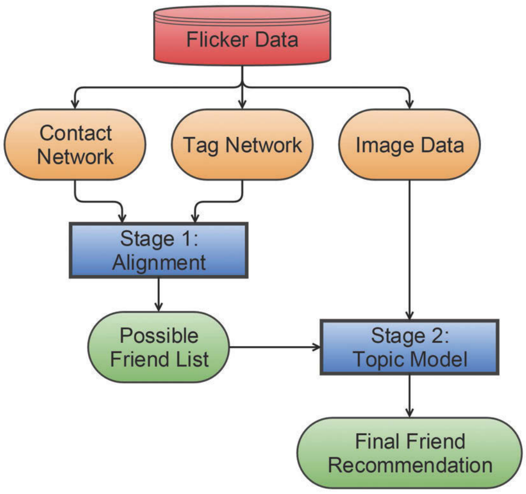


Figure 6.1: Two-Stage System Illustration

6.3.1 First Stage: Network Alignment

The detailed alignment method has been discussed in (Huang, Zhang, Wang & Hua 2016). The following is an introduction of its basic idea. An individual may join different social networks for different purposes. For example, one may at the same time join a football fan network for physical practice and a restaurant information sharing network to look for the best food. He/she plays different social roles in these different networks, and might make different friends. However, these different social roles for one individual are not independent, but related to each other.(The man might look for some food that helps quickly recuperate after hard physical practice). The motivation for social network alignment lies on the fact that these different networks, though having different edges (relationships), are usually related to each other. Taking Flickr as an example, according to the uploaded-tag-similarity of each user and their contact list, a tag similarity network and a contact network are formed. Although the topologies of the two networks are not the same, they are related to each other, for users uploading similar tags on

Flickr have higher probability to make friends with each other. By digging the correlation of the topologies of different networks we may make inference for the knowledge from one domain to another.

Specifically, we align the Flickr tag-similarity network with the contact network, so that after the alignment, one tight edge between two users in the tag similarity network would imply that these two users have higher probability to have contact with each other. We align the tag-similarity network with the contact network by selecting important tag features. The reason we apply feature selection here lies in the phenomenon that when we look for online friends, it is common that we do not take care of all the factors of a person but concentrate on certain points that would interest ourselves. As an example, a traveller might post his photos with the following tags: "Sydney", "Blue Mountain", "great view", and "street". Among these tags some people might contact him/her for some more details about the experiences in ''Sydney" and ''Blue Mountain", but seldom would have interests about "great view" or "street" because they are too common. We can treat these two tags as redundancy for friend making. Based on this observation, we believe that some Flickr tags can be more indicative in the task of friend recommendation, because they are more important to reflect the connections on the contact network. We can treat these tags as important features for friend recommendation. Inspired by this phenomenon, we design a method to choose some important features that are more helpful for friend making decision.

Mathematically, assume that the feature selection matrix to be $\mathbf{W}$, the known tag-user matrix to be $\mathbf{X}$, the tag distance matrix to be $\mathbf{L}$, and the first d eigenvector-matrix of the contact network to be $\mathbf{V}$, the important feature can be obtained by solving the following problem:

$$\min_{\mathbf{W}} \|\mathbf{XW}-\mathbf{V}\| + \mu tr(\mathbf{W}^T\mathbf{X}^T\mathbf{LXW}) + \lambda\|\mathbf{W}\|_{2,1} \tag{6.1}$$

The first term of Eq. (6.1) aligns the tag-similarity network to the contact network so that they become more similar to each other, and the second term preserves the local structure of the original tag-similarity network. The third term is for regularization. In this way the tag feature selection matrix $\mathbf{W}$ makes the topology of the tag-similarity network more similar to the contact network, while preserving the topology of the tag-similarity network as much as possible. In other words, we align the tag-similarity network to the contact network. By comparing the similarity of two users on the those important tags we can generate a possible friend list for each user. The solution of $\mathbf{W}$ in Eq. (6.1) is discussed more thoroughly in (Huang et al. 2016).

6.3.2 Second Stage: Topic Model

In the previous stage we get a possible friend list by considering the correlation between the tag and contact networks on Flickr. However, as the real world friend relationship is affected by many factors (Alex 2012), one stage is usually not enough for a precise friend recommendation. In the following stage, we introduce the image data as auxiliary information to refine the recommendation list.

We apply the topic model to combine the image data and the friendships in Flickr. It is common sense that a person uploads a photo on Flickr because he/she likes the photo. Why does he/she like the photo? We assume that in one's mind, some latent interest factors control his/her taste of image. For example, some people like colourful, vivid photos, while others prefer spectacular or imposing ones; children enjoy comic-style pictures while adults have more interests in realistic-style paintings; young women pay much attention to photos of beautiful clothes while young men to electrical devices. These latent factors are determined by various aspects such as age, gender, living experiences, etc. and can not be observed or simply summarised. We assume individuals' interest latent factor to be $\mathbf{v}$. Each image contains the factors that attract people, such as colour, line, or history, which we assume to be $\mathbf{a}$. The correlation of $\mathbf{v}$ and $\mathbf{a}$ determines whether a user would upload an image.

Similarly, we assume that each user exhibits some attractive factors during his/her activities in Flickr such as uploading photos, writing descriptions of photos and making comments, etc. We also summarise these attractive factors with the third latent variable $\mathbf{b}$. Notice that the same user's interest latent factor $\mathbf{v}$ and attractive factor $\mathbf{b}$ are not the same. The combination of $\mathbf{b}$ and $\mathbf{v}$ determines whether two users should make friends with each other. For simplicity we view them as independent from each other. The topic model is shown in Figure 6.2.

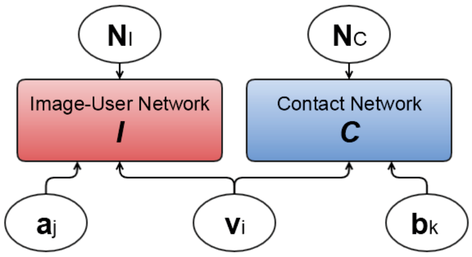


Figure 6.2: The Probabilistic Topic Model Combining Image-User Network and Contact Network

In Figure 6.2, $\mathbf{C}$ and $\mathbf{I}$ stand for the $0-1$ contact network and image-user network, respectively. $\mathbf{C}$ is an $n \times n$ matrix where n is the number of users. $\mathbf{I}$ is an $n \times f$ matrix where f stands for the number of total features. For $\mathbf{C}$, if user k and user j are friends with other, then $\mathbf{C}_{kj}$ equals one, and zero otherwise. For $\mathbf{I}$, if the uploaded photos of user i contain an image feature j, then $\mathbf{I}_{ij}$ equals one, and zero otherwise. a stands for image factor, and b stands for individuals' social interest factor, respectively. v stands for individuals' common interest factor that has effect on both his choice of images and friends. $\mathbf{N_I}$ and $\mathbf{N_C}$ stand for zero-mean additive noises. The relationship can be mathematically expressed as follows:

$$\mathbf{I}_{ij} = a_i \times v_j + \mathbf{N}_{Iij}, \mathbf{C}_{kj} = b_k \times v_j + \mathbf{N}_{Ckj} \tag{6.2}$$

We assume that all the latent random variables a_i, b_k and v_j are Gaussian distributed with the parameters of means and variances of μ_a, σ_a, μ_b, σ_b, μ_v, and σ_v, respectively. The reason we choose Gaussian distribution is as follows: Although some other distributions that are in the form of an H-function (such as Beta, Gamma or Rayleigh distributions) would lead to some calculation convenience (Springer 1979), we assume Gaussian distribution here because it is defined on the whole real domain and contains negative values and has a central point, while other distributions such as Beta are only defined on the positive real domain.

The coupling between random variables a, b and v makes the integral of Eq. (6.2) often intractable. Traditional methods dealing with Eq. (6.2) contain Gibbs sampling (Griffiths & Steyvers 2002) and variational inference (Blei et al. 2003*c*). Gibbs sampling meets with difficulties when the data scale is large, and the variational method applies some approximation that the precision is hard to control. In the following we develop a new approach to solve Eq. (6.2) that is based on Mellin transform and series expansion.

6.4 Series Expansion

6.4.1 Product of Gaussian Random Variables

When dealing with the distribution of product of random variables, the Mellin transform is an essential tool (Springer 1979). We take the first equation in Eq.(6.2) to explain its basic idea. For simplicity we first neglect the noise term $\mathbf{N}_{ij}$ (its effectiveness is to be discussed later) and we have $\mathbf{I}_{ij} = a_i v_j$ for two random variables a_i and v_j with different probability distribution functions. One useful property for Mellin transofrm is: the Mellin transform of the product of two probability density functions (PDF) is equal to the product of the Mellin transforms of their PDFs.

Mathematically, we recall the following rule (Springer 1979): If a_i and v_j are two non-negative random variables with the PDFs $f_a(a_i)$ and $f_v(v_j)$, their product $\mathbf{I}_{ij} = a_i v_j$ has a distribution $h(I_{ij})$, and then the Mellin transform of $h(I_{ij})$ is precisely the product of Mellin transform of $f_a(a_i)$ and $f_v(v_j)$, respectively. The expression is given as:

$$\mathcal{M}(h(\mathbf{I}_{ij})) = \mathcal{M}(f_a(a_i))\mathcal{M}(f_v(v_j)) \tag{6.3}$$

where the Mellin transform and its inverse of an analytical function $f(x)$ are defined as follows:

$$\mathcal{M}(s) = \int_{0}^{+\infty} x^{s-1} f(x) dx \tag{6.4}$$

$$\mathcal{M}^{-1}(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} x^{-s} \mathcal{M}(s) ds \tag{6.5}$$

where c in Eq. (6.5) stands for an arbitrary real number. With the help of Eq. (6.3)-(6.5) and the known distribution of a_i and v_j, we can give an exact mathematical expression for distribution of the coupling of the two random variables a_i and v_i.

In this way we can first deduce the Mellin transform of each of the distributions, then make product of the two, and finally inverse the Mellin transform to get the final product distribution. In this way, we first calculate the distribution of $\mathbf{I}$ in Eq. (6.2).

From the previous assumption we know that a_i, b_k and v_j follow the Gaussian distribution with mean μ_{ai}, μ_{bk}, μ_{vj} and the variance σ_{ai}, σ_{bk}, σ_{vj}. We further take the symbol of f_{ai}, f_{bk} and f_{vj} as their PDFs. We first do the Mellin transform on a_i and v_j separately to get $\mathcal{M}(f_a(a_i))$ and $\mathcal{M}(f_v(v_j))$, and then we product them and do the inverse Mellin transform to finally get the distribution of product of two random variables, which is the distribution of the variables in image-user matrix $\mathbf{I}$. The details are given in (Springer 1979) and (Pruett 1972), which provide two equivalent expressions for the distribution of two Gaussian random variables. We apply the expression from (Pruett 1972) and the details are briefly outlined in the following.

To calculate the distribution of $\mathbf{I}_{ij} = a_i v_j$ with Gaussian random variables a_i and v_j, we take the Mellin tranform of $f_a(a_i)$ and $f_v(v_j)$. Notice that according to Eq. (6.4), the positive and negative parts of the distribution of a_i and v_j should be considered separately. We apply the property that the Mellin transform of the standard Gaussian distribution is Gamma function (Bateman 1954): $\mathcal{M}\{e^{-x^2/2}\} = 2^{s/2-1}\Gamma(s/2)$, and a non-central Gaussian distribution can be expressed as a standard Gaussian distribution multiplied by a series in the form: $\mathrm{e}^{-\frac{1}{2}(x-\mu)^2} = \mathrm{e}^{-\mu^2/2}\sum_{j=0}^{\infty}\frac{1}{j!}\mu^j x^j \mathrm{e}^{-x^2/2}$. If we define the following:

$$a_{i1} = max(a_i, 0), v_{i1} = max(v_j, 0),$$
$$a_{i2} = min(a_i, 0), v_{i2} = min(v_j, 0),$$
$$\mathbf{I}_{ij-1} = a_{i1}v_{i1}, \mathbf{I}_{ij-2} = a_{i1}v_{v2},$$
$$\mathbf{I}_{ij-3} = a_{i2}v_{v1}, \mathbf{I}_{ij-4} = a_{i2}v_{j2}$$

And we also define the probability distribution function of $\mathbf{I}_{ij-1}$, $\mathbf{I}_{ij-2}$, $\mathbf{I}_{ij-3}$, and $\mathbf{I}_{ij-4}$ to be $h_1(\mathbf{I}_{ij})$, $h_2(\mathbf{I}_{ij})$, $h_3(\mathbf{I}_{ij})$ and $h_4(\mathbf{I}_{ij})$, respectively. Following the methods of (Pruett 1972), and taking $\mathbf{I}_{ij-1}$ as an example, we have:

$$\mathcal{M}_{\mathbf{I}_{ij-1}}(s) = \sum_{o=0}^{\infty}\frac{\mu_{a_i}^{2o}}{(2o)!}\frac{\mu_{v_j}^{2o}}{(2o)!}\Gamma^2(s) \tag{6.6}$$

To get the distribution of $\mathbf{I}_{ij-1}$, we do the inverse Mellin transform of Eq. (6.6) as:

$$h_1(\mathbf{I}_{ij}) = \sum_{o=0}^{\infty}(\frac{1}{2\pi i})\int_{c-i\infty}^{c+i\infty}(y^2)^{-s}\frac{\mu_{a_i}^{2o}}{2o!}\frac{\mu_{v_j}^{2o}}{2o!}\Gamma^2(s+o)ds \tag{6.7}$$

Eq. (6.7) is an integral on half of the complex plane. According to Residue Theorem (Ahlfors 1979), the solution is expressed with the infinite residues that are related to the poles on the real plane. By calculating the residues we get:

$$h_1(\mathbf{I}_{ij}) = C1\Bigg[\sum_{o=0}^{\infty}C2\sum_{s=o}^{\infty}[\frac{(\mathbf{I}_{ij})^{2s}}{\prod\limits_{t=0}^{s-o-1}(-s+o+t)^2}(2\psi(1) \\ -2\sum_{w=0}^{s-o-1}\frac{1}{-s+o+w}) - \frac{(\mathbf{I}_{ij})^{2s}\ln((\mathbf{I}_{ij})^2)}{\prod\limits_{w=0}^{s-o-1}(-s+o+w)^2}]\Bigg] \tag{6.8}$$

where $C1 = \frac{1}{\pi}\mathrm{e}^{-\frac{1}{2}(\frac{\mu_{ai}^2}{\sigma_{ai}}+\frac{\mu_{vj}^2}{\sigma_{vj}})}$, $C2 = \big((\frac{1}{(2o)!})^2(2\frac{\mu_{ai}^2}{\sigma_{ai}}\frac{\mu_{vj}^2}{\sigma_{vj}})^o\big)$, and $\psi(1)$ is the Euler-Mascheroni constant.

Similarly we should also consider the case of $h_2(\mathbf{I}_{ij})$ for $a > 0 \cap v < 0$, $h_3(\mathbf{I}_{ij})$ for $a < 0 \cap v > 0$, and $h_4(\mathbf{I}_{ij})$ for $a < 0 \cap v < 0$. To sum up, we have:

$$\begin{aligned} h(\mathbf{I}_{ij}) &= h_1(\mathbf{I}_{ij}) + h_2(\mathbf{I}_{ij}) \qquad (y > 0) \\ &= h_3(\mathbf{I}_{ij}) + h_4(\mathbf{I}_{ij}) \qquad (y < 0) \end{aligned} \tag{6.9}$$

In a similar manner we can give the expression for $h(\mathbf{C}_{kj})$

Here we give a short discussion about this series. In the first place, this is basically an alternating and power series (E.L.Lady 1998) with infinite terms, with some of the terms multiplied with a logarithm factor. This is a series that when the sequence number of the term increases, the absolute value of the term increases. Some of the terms are positive and some are negative, and the sum of the terms eventually becomes convergent, as discussed in (Balser 1994). However, similar to some of the convergent Taylor series, when the absolute value of the series terms is large, these series converge only when the term number of the series is also large. In order to make the series to converge rapidly with relatively a small number of terms, in practice, we may normalise the value of $\mathbf{I}_{ij}$ to be relatively small (In the experiments, the ground truth of $\mathbf{I}_{ij}$ and $\mathbf{C}_{ij}$ are 0 or 1, which is small enough).

6.4.2 Additive Noise

From Fig. 6.2 we see that after the products of a, v and b, v, the results should also add a bias value or noise to get the value of $\mathbf{I}_{ij}$ and $\mathbf{C}_{kj}$. In practice it can be interpreted as all the outer environmental influences other than the users and the items. For example, the change of seasons for the favour of clothing, or the change of temperature for the preference of food, etc. Mathematically the PDF of two independent random variables are the convolution of their PDFs of the two (Springer 1979). In our case, we can simply consider the environmental influences $\mathbf{N}_I$ and $\mathbf{N}_C$ to be independent from the image factor a, social attractive factor b and individual's latent factor v. For simplicity we assume the additive noise of $\mathbf{N}_I$ and $\mathbf{N}_C$ to be Gaussian distributed with zero mean and variance of σ_{Ni} and σ_{Nc}, respectively. Taking $\mathbf{I}_{ij}$ for example, from

$$
\begin{aligned}
h(\mathbf{I}_{ij}) = {} & \frac{1}{\pi} e^{-\frac{1}{2}(\frac{\mu_{ai}^2}{\sigma_{ai}}+\frac{\mu_{vj}^2}{\sigma_{vj}})} \Big[\big[\sum_{t=0}^{\infty}(\frac{1}{(2t)!}\frac{1}{(2t)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^t(2\frac{\mu_{vj}^2}{\sigma_{vj}})^t)\sum_{s=t}^{\infty}[\frac{\mathbf{I}_{ij}^{2s}\mathrm{e}^{\frac{(-\mathbf{I_{ij}}^2)}{\sigma_N{}^i}}}{\prod_{m=0}^{s-t-1}(-s+t+m)^2}(2\psi(1)-2\sum_{m=0}^{s-j-1}\frac{1}{-s+t+m}-\frac{\mathbf{I}_{ij}^{2s}\ln(\mathbf{I}_{ij}^2)}{\prod_{i=0}^{s-t-1}(-s+t+m)^2}] \\
& +\sum_{r=0}^{\infty}\sum_{t=0}^{\infty}(\frac{1}{(2t)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^t\frac{1}{(2r)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^r+\frac{1}{(2t)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^t\frac{1}{(2r)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^r)\sum_{s=t}^{r-i}[\mathbf{I}_{ij}^{2s}\frac{\prod_{m=1}^{r-s-1}m}{\prod_{q=0}^{s-j-1}-q-1}\ln(\mathbf{I}_{ij}^2)]+\sum_{t=0}^{\infty}\sum_{r=t+1}^{\infty}(\frac{1}{(2t)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^t \\
& \frac{1}{(2r)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^r+\frac{1}{(2t)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^j\frac{1}{(2r)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^{2r})\sum_{s=r}^{\infty}[\frac{\mathbf{I}_{ij}^{2s}(2\psi(1)-\sum_{m=0}^{r-t-1}\frac{1}{-s+t+m}-\sum_{q=r-t}^{s-t-1}\frac{2}{-s+t+q})}{\prod_{m=0}^{r-t-1}(-s+t+m)\prod_{q=r-t}^{s-t-1}\prod_{q=r-t}^{s-t-1}(-s+t+q)^2}-\frac{\mathbf{I}_{ij}^{2s}\ln(\mathbf{I}_{ij}^2)\mathrm{e}^{\frac{(-\mathbf{I_{ij}}^2)}{\sigma_N{}^i}}}{\prod_{i=0}^{r-j-1}(-s+j+i)\prod_{k=r-1}^{s-j-1}(-s+j+k)^2}]\big] \\
& \pm\big[\sum_{k=0}^{\infty}(\frac{1}{(2k+1)!}\frac{1}{(2k+1)!})\sum_{s=k}^{\infty}[\frac{(\mathbf{I}_{ij}^2)^{s+1/2}}{\prod_{m=0}^{s-q-1}(-s+q+m)^2}(2\psi(1)-s\sum_{m=0}^{s-q-1}\frac{1}{-s+q+m})-\frac{(\mathbf{I}_{ij}^2)^{s+1/2}\ln(\mathbf{I}_{ij}^2)\mathrm{e}^{\frac{(-\mathbf{I_{ij}}^2)}{\sigma_N{}^i}}}{\prod_{i=0}^{s-q-1}(-s+q+i)^2}]+\sum_{p=1}^{\infty}\sum_{q=0}^{p-1}(\frac{1}{(2q+1)!} \\
& (2\frac{\mu_{ai}^2}{\sigma_{ai}})^{(p+0.5)}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^{p+0.5}+\frac{1}{(2q+1)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^{q+0.5}\frac{1}{(2p+1)}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^{p+0.5})\sum_{s=q}^{p-1}[\frac{(\mathbf{I}_{ij}^2)^(s+1/2)\prod_{m=1}^{p-s-m}(m)}{\prod_{n=0}^{s-q-1}(-n-1)}\ln(\mathbf{I}_{ij}^2)]+\frac{1}{(2q+1)!}(2\frac{\mu_{vj}^2}{\sigma_{vj}})^{q+0.5} \\
& \frac{1}{(2p+1)!}(2\frac{\mu_{ai}^2}{\sigma_{ai}})^{p+0.5})\sum_{s=p}^{\infty}[\frac{(\mathbf{I}_{ij}^2)^{s+1/2}(2\psi(1)-\sum_{m=0}^{p-q-1}\frac{1}{-s+q+1}-\sum_{l=p-q}^{s-q-1}\frac{2}{-s+q+l})}{\prod_{i=0}^{p-q-1}(-s+q+m)\prod_{l=p-q}^{s-q-1}(-s+q+l)^2}-\frac{(\mathbf{I}_{ij}^2)^{s+1/2}\ln(\mathbf{I}_{ij}^2)\mathrm{e}^{\frac{(-\mathbf{I_{ij}}^2)}{\sigma_N{}^i}}}{\prod_{i=0}^{p-q-1}(-s+q+m)\prod_{l=p-q}^{s-q-1}(-s+q+l)^2}]\big]\Big]
\end{aligned}
\tag{6.12}
$$

Eq. (6.8) we see that the most important calculation is the convolution of the Gaussian function from additive noise $\mathrm{e}^{-\mathbf{I}_{ij}^2/\sigma_{Ni}^2}$ and the term $\mathbf{I_{ij}}^{2s}\log(\mathbf{I_{ij}}^2)$ from Eq. (6.8), which is formally written as follows:

$$d_2(\mathbf{I_{ij}}) = \mathrm{e}^{-\mathbf{I}_{ij}^2/\sigma_{Ni}} * \mathbf{I}_{ij}^{2s}\ln(\mathbf{I}_{ij}^2) \tag{6.10}$$

By calculating the convolution we see Eq. (6.10) can be expressed as follows:

$$d_2(\mathbf{I_{ij}}) = \mathbf{I_{ij}}^{2s+2}(\frac{\ln\mathbf{I_{ij}}^2}{2s+2}-\frac{1}{(2s+2)^2})\mathrm{e}^{\frac{(-\mathbf{I_{ij}}^2)}{\sigma_{Ni}}} \tag{6.11}$$

In this way we can get a series expression of Eq. (6.9).

So the expression for the distribution of $\mathbf{I}_{ij}$ considering the additive noise is given in Eq.(6.12).

In a similar way we can also obtain the distribution of $\mathbf{C}_{jk}$.

6.4.3 EM for Parameter Estimation

Applying the above, we obtain the exact infinite expansion expression of the PDF. of $\mathbf{I}$ in a series form given in Eq.(6.12). The expression of $\mathbf{C_{kj}}$ can be obtained in a similar way. From Eq. (6.12) we can see that the exact value of μ_{ai} and σ_{ai} does not matter much, but the value of $\frac{\mu_{ai}^2}{\sigma_{ai}}$ matters. So we can assume that a_i has standard derivation of 1, and we only need to calculate the average value of a_i. Similarly, we also do not need to calculate σ_{bk} and σ_{vj} but only assume that v_j and b_k have standard derivation.

All the parameters $\mathcal{P}$ are summarised in Table 6.1. As mentioned in Section 6.4.1, in the experiments, when we choose the starting point of the parameters not too large, we can make the series converge in a relatively small number of terms. Then we can apply the standard EM method to refine the parameters iteratively. Experimental result shows that the number of series terms can be no longer than 10 and after several EM iterations, the precision becomes stable.

Table 6.1: Summary of Parameters

μ_{ai}	mean of image factor a_i
μ_{bk}	mean of individuals' social attractive factor b_k
μ_{vj}	mean of individual's interest factor v_j
σ_{NI}	variance of image noise N_I
σ_{NC}	variance of social noise N_C

The EM training process is introduced as follows. For E step, Consider Eq.(6.12), which is the Equation we want to maximize by knowing the value of $\mathbf{I}_{ij}$, with respect to the parameters $\mathcal{P}$ as follow:

$$\max_{\mathcal{P}} h(\mathbf{I} \mid \mathcal{P})) \tag{6.13}$$

In the M step, we find the derivative of each parameter in $\mathcal{P}$ by fixing other parameters. Then we set the derivative to be zero to get the value for each parameter. The whole process goes until convergence.

One problem to solve Eq. (6.12) is that Eq.(6.12) contains not only polynomial terms but also exponential terms for the parameters. For simplicity we can make an assumption that the parameters are relatively small, and then we can use the first several terms, or following (Richards & Acharya 2010) to get a polynomial expression of the parameters, to make Eq (6.12) solvable.

Another problem is that for some parameters such as μ_{ai}, it contains infinity high order terms that makes the solution intractable. Again we can make the assumption that these parameters to be smaller than one, and discard the high order terms. In practise we keep the terms whose orders are equal or lower than 4, and follow the method discussed in (*Ferrari's Solution of a Quartic Equation* n.d.) to calculate the values of the parameters.

From Eq. (6.12) we can obtain the parameters that related to the image-user matrix **I**, such as μ_{ai}, μ_{vj}, and σ_{NI}. In a similar manner we can also get the parameters related to the contact matrix **C**, such as σ_{NC}, μ_{bk}, and also μ_{vj}. By iteratively updating these parameters relating to the two matrix we can finally determine the value of all the parameters.

After the EM iterations we fix all the parameters in Table 6.1 and according these parameters we can make the final friend recommendation.

6.4.4 Recommendation Method

When a new user i comes into the network, he/she may upload some favourite photos as well as some tags. The recommendation procedure is divided in two stages. In the first stage, a list of possible friends is generated according to the similarity of the selected important tags. In the experiments, we put the top 200 users into the list.

In the second stage, according to the features of the images uploaded by use i, we get the individuals' interest factor v_i of this user. For a user k in the possible friend list obtained from the first stage, we can also calculate his/her attractive and interest factors b_k. The similarity score of user i and k is obtained by $S_{ik} = v_i b_k$. The higher the similarity score, the more likely that they are to be friends. So we can rank the 200 users in the list according to the similarity score with user i, and recommend the top ones as user i's friends

The whole procedure is given in Algorithm 6.1.

Algorithm 6.1 Two Stage Friend Recommendation

Require:

tag feature matrix $\mathbf{T}$, contact matrix $\mathbf{C}$, image-user matrix $\mathbf{I}$, tag and image feature of the new user $\mathbf{t}$ and $\mathbf{i}$, the numbers of possible friends in Stage 1 and final friends $k1$ and k, respectively

Ensure:

Friend recommendation list of the new friend

Training:

Stage I

1: Determine λ and μ in Eq. (6.1) via cross validation.

2: Solve Eq. (6.1) with the method in (Huang et al. 2016)

Stage II

3: Generate the expression of distribution of h(Eq. (6.12)) in the form of series.

4: Apply EM method determining the parameters in Table 6.1

Testing:

5: Stage I: Use $\mathbf{W}$ calculated in Step 2 to obtain $k1$ possible friend list.

6: Stage II: Use the parameters in Step 4 to refine the final recommendation friend list, recommend top k users

---

6.4.5 Complexity Analysis

The complexity analysis of our algorithm is also divided by the two stages as follows:

Considering the first stage, the complexity of the network alignment is mainly decided by two steps: the eigenvalue calculation and the inverse of the similarity matrix, which is given by $\max(\min\{n, e\}^3, dn^2)$ as discussed in (Huang et al. 2016). e stands for the number of total tags. As previous defined, n stands for the number of users, and d stands for the first d eigenvectors.

To solve the topic model of the second stage, Assume together we need to make L time iterations. in each iteration of the EM step, assume that we calculate the first g terms of the series of Eq. (6.12)(In practise we make $g = 4$). And it takes e steps to solve a 4th order polynomial equation, as mentioned in Section 6.4.3. Then the complexity would be of $\mathcal{O}(L*e*g*(n*f+n*n))$, where f is the number of image features, as previously defined.

6.5 Experiments

In this section, we make experiments to show the advantage of our proposed method. First, we introduce our social media dataset, and then we discuss the results of our algorithm by comparing it with reference methods. We utilise a cluster containing 16 cores and 128G memories to run our experiments.

6.5.1 Dataset and Feature Extraction

We crawled a social network from the big image sharing site Flickr. As the data set is quite large, a relatively unbiased dataset was obtained. In total we crawled the data of 30000 users, and for each user, we crawled all their photos, and tags of each photo. We tried the SIFT feature and the deep network extracted features through an CNN autocoder realized by Caffe (Jia, Shelhamer, Donahue, Karayev, Long, Girshick, Guadarrama & Darrell 2014*a*). For the CNN features we follow the steps of the widely used AlexConvNet (Krizhevsky, Sutskever & Hinton 2012*b*) and use the 4096 dimensional features vectors from the last full-connected layer. In most cases the CNN features performs better than the SIFT features, so we chose the CNN extracted features for the rest of our experiments. In the future we can also refine feature extraction method for better performance. We then crawled the user contact information to form the contact network. The contact information in Flickr was acquired by checking if a user added another user to his/her friend list, or vice versa. We crawled all the contacts between any two users in our dataset. A short summary of our dataset is given in Table 6.2 :

Table 6.2: Dataset Statistics

Users	30000
Photos	1,356,293 photos from 30000 users
CNN features	4096
Contact	628,153 friend links among users
Tags	42,739 words after filtering

6.5.2 Settings and Metrics

Our task is to make precise contact information prediction. When a new user enters into the social network, we recommend new friends according to key words and photos that represent the user's interests.

In friend recommendation, assume we recommend T friends to each user. We use the existing contact information as the ground truth for training and testing. In the first stage, the parameters of Eq. (6.1) λ, μ are determined on the training set by a fourfold cross validation to find the best. The ranges for these parameters are: $\lambda \in 10^{[-2:1:3]}$, $\mu \in 10^{[-2:1:3]}$.

We use the method summarised in Algorithm 6.1 to recommend friends to new users. We use the recommendation precision metrics to show the effectiveness of the proposed algorithm. In our experiment, precision is defined as the number of correctly recommended friends divided by all the recommended users. We also introduce the precision-recall curve to further show the advantage of our algorithm, where recall is defined as the number of the correctly recommended friends divided by the number of all friends, and F-measure is the combination of the two.

6.5.3 Reference Methods

The performance analysis of our first stage: network alignment methods can be seen in some previous related papers such as (Huang, Zhang, Lu & Hua 2015)(Huang et al. 2016). For the performance analysis of the second stage in which the topic model method is applied, we choose several widely-used methods for comparison.

The first is the variational method, which has been widely applied in this decade for solving the Bayesian network problem (Blei et al. 2003*c*). Basically we apply the methods in (Song et al. 2013) with some slight modifications to our problem.

The second is the widely-used Gibbs sampling method, which is also very popular in dealing with topic model. Compared with the variational method, the idea of Gibbs sampling is simpler but usually it has difficulty in dealing with large scale problems. We apply the method based on (Min et al. 2015) for comparison.

The third method is a co-clustering based method (Huang et al. 2015). It is not a topic model-based method, but has a relatively simpler concept: In the second stage, we do co-clustering of image features, users and tags to get a . We apply a simple ranking method, similar to (Huang et al. 2015) for the final friend recommendation.

To further check the advantage of our method, we also compare our whole two-stage recommendation algorithm with several state-of-the-art recommendation systems. The first one is based on matrix factorization(MF). MF method decomposes the item-user or user-user matrix to infer the latent factors that catch individuals' interests and has been widely discussed for different kinds of recommendation problems (Ma et al. 2008*b*)(Guo et al. 2015). In this chapter we apply a recent method proposed in (Guo et al. 2015) for comparison, for it jointly considers the information from two different domains.

Another recent method is based on Bayesian collaborative filtering that takes the social connections into account, called SBPR (Zhao, McAuley &

King 2014). As a widely-used recommendation method, collaborative filtering assumes that two users that choose the same items behave similar on other items. Traditional collaborative filtering methods do not consider much about the social connections between users. SBPR removes this drawback by taking the social connections into account by assigning a social coefficient to each user. [1].

At last we consider a multi-network based algorithm for comparison. When considering social multiple network problems, transition probability propagation is a method that is frequently used (Jiang et al. 2012)(Liu et al. 2012). We choose (Jiang et al. 2012) as a reference method for the following reasons: 1) It considers the relationships of different networks, which is similar to our idea; 2) It uses the information of other networks for recommendation, which again has some similarities with ours. (Jiang et al. 2012) enhances the links in one network and between different networks using a random walk propagation method. After a sufficient number of walks, it obtains the modified link weights between each user pair. We use the weights for friend recommendation.

6.5.4 Experimental Results

Here we report the results of our method for friend recommendation as follows.

Performance of Series Expansion

In this experiment we compare the proposed series expansion method with the variational, Gibbs sampling, and co-clustering methods in the second stage. We treat the performance of the first stage as the baseline.

From Fig. 6.3 we can see that our method has the best performance for accurate recommendation. $P@X$ stands for that each time we recommend

[1]The realization of (Guo et al. 2015) and (Zhao et al. 2014) is based on the existing open-source Java package LibRec at http://www.librec.net/

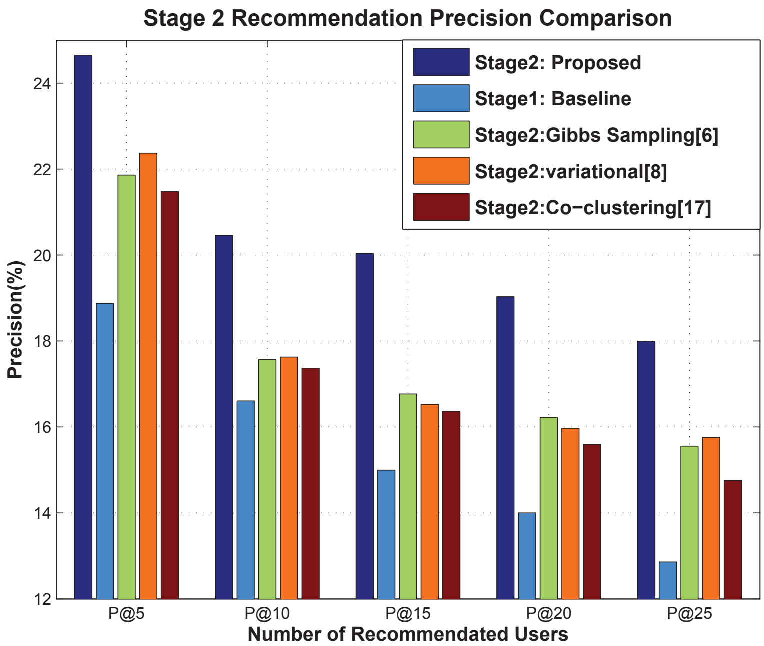


Figure 6.3: Stage 2 Recommendation Precision Comparison

the top X friends to users. Generally, the second stage improves the recommendation precision from only applying the first stage, illustrating the effectiveness and necessity of applying the two staged methods. Our proposed method improves about 5-7% compared with the performance of the first stage, and also makes about 2-3% improvement compared with the Gibss sampling method and the variational method. The reason for the improvement mainly lies in that we apply an exact expression to approach the PDF of the data, rather than an approximation or sampling method. The co-clustering method lacks the ranking ability and thus the performance is not good.

Fig. 6.4 illustrates the precision-recall curve of the proposed and reference methods. Based on the result of the first stage, the series expansion method achieves the highest performance(The upper right line on the figure). We can see from Figure. 6.4 that when precision or recall is fixed, we can achieve a 3-4% improvement over the best reference methods. This

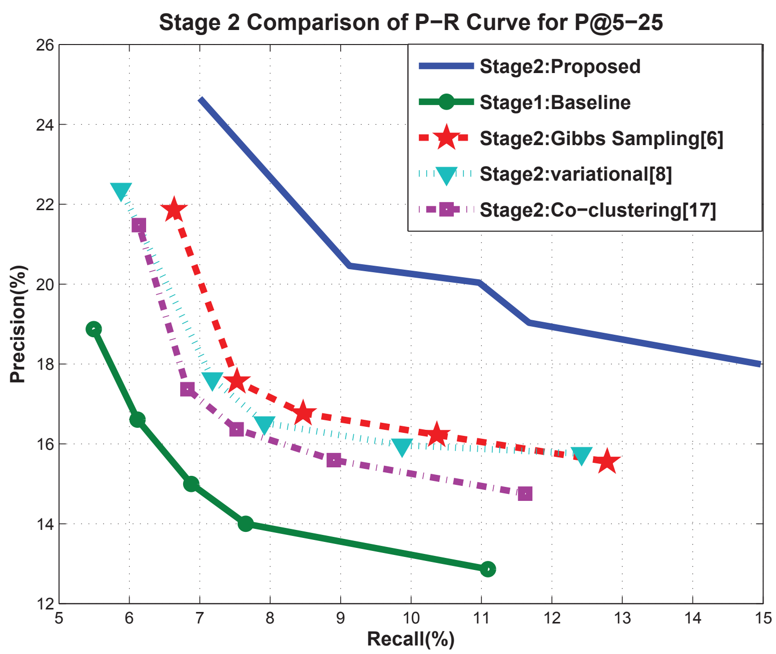


Figure 6.4: Recommendation Precision and Recall for Stage 2

means that the proposed method can achieve both the highest precision and recall. This experimental results shows that the series expansion method can best approximate the real distribution of the data, and thus makes the most precise recommendation.

On the other hand, the proposed method have also imposed Gaussian distribution assumption to the latent variables a, b, and v. This may also cause some negative effect although it can give an analytic expression. It is worthy to make a depth observation of the distribution of the latent variables in our future studies.

Performance of The Proposed Two-stage Method

Now we compare our two-stage method with some recently-proposed recommendation systems as mentioned in 6.5.3. The main results for precision and precision-recall curve are shown in Fig. 6.5 and 6.6.

From Fig. 6.5 and Fig. 6.6 we can see that our system achieves the best

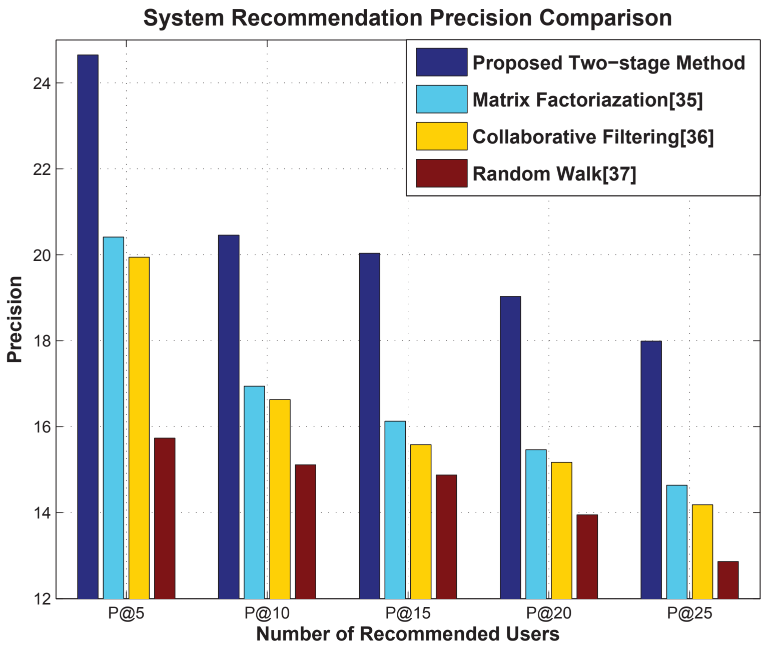


Figure 6.5: Two-stage Recommendation Precision Compared with State-of-The-Art Systems

performance, compared with other state-of-the-art recommendation systems. In average, our system improves the recommendation accuracy by about 3-4%, compared with the second best one. MF based method (Guo et al. 2015) has the best performance among all the reference methods, for it decomposes the item-user and user-user matrix into different social factors in a proper way. The reason that the proposed method performs better than MF might lies in that the MF method does not consider the whole distribution of the network and is trapped into some local optimum. Collaborative filtering based method (Zhao et al. 2014) has slight lower performance than (Guo et al. 2015), the reason might be that its assumptions about the users' positive and negative feedback are not very proper for the Flickr dataset. Finally, the random-walk based method (Jiang et al. 2012) has the lowest performance, since the random walk algorithm is not accurate enough for precise friend recommendation.

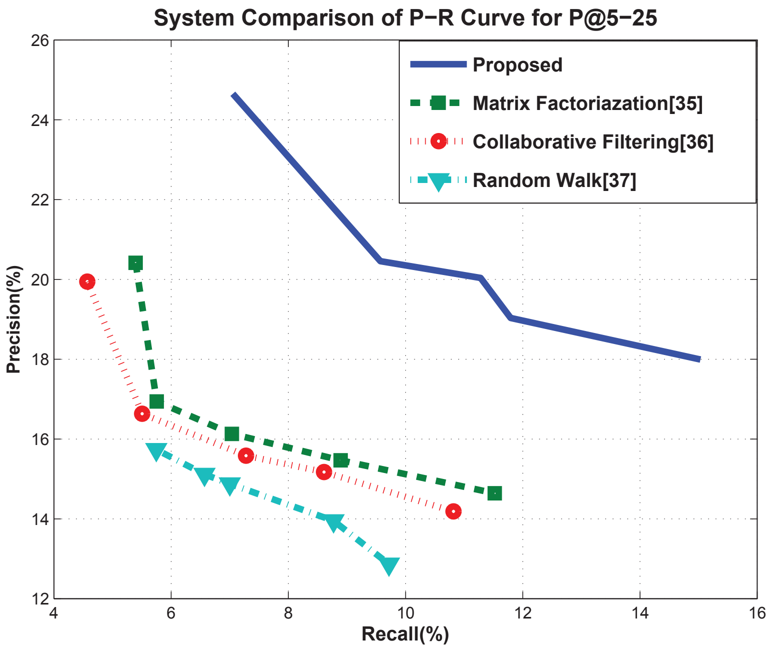


Figure 6.6: Recommendation Precision and Recall Compared with State-of-The-Art Systems

6.5.5 The Influence of Several Settings

The Influence of Additional Noise The introduction of the additive noise, as shown in Section 6.4.2, makes the model more precise. However, it also leads to complicated inferences and calculations. In the following experiment we study the influence of the additive noise. In Table 6.3, we compare the recommendation accuracy of the model that contains the additive noise and the model that does not contain the noise.

Table 6.3: The Influence of Additive Noise

Precision(%)	P@5	P@10	P@15	P@20	P@25
Model With Noise	24.6	21.0	19.8	18.1	17.5
Model Without Noise	22.7	19.3	18.2	16.8	15.9

From Table 6.3 we see that by considering the additive noise we get a precision gain of about $1 - 2\%$, which is useful in the case where a more precise result is required.

The Influence of the Value of $\mathbf{C}_{kj}$ and $\mathbf{I}_{ij}$ As shortly discussed in Section 6.4, the convergence speed of the series is largely determined by the level of values of **C** and **I**. If it is too large, then the convergence speed will decrease, leading to either the inaccuracy of the model, or larger number of terms. On the other hand, if the level is too small, the logarithmic terms in Eq. (6.12) will drop quickly and make the system unstable. In our experiments, contact network **C** stands for the intimacy of two individuals and in the image-user network **I**, it stands for to what extent an individual favours an image feature. The values of each entry of **C** and **I** can be set according to our requirements. For example, we can set $\mathbf{C}_{jk}$ to be 1 if two individuals are friends with each other and 0 otherwise; for image-user network we can also set $\mathbf{I}_{ij} = 1$ if an individual has a certain image feature in his/photos, and 0 otherwise. On the other hand, we can also raise the level of the elements in **C** and **I** to be 5 or 10, or reduce it to be smaller than 1. The relationship between any two nodes would not change in the networks by varying the element value of **C** and **I**, but the value does have an influence on the accuracy in our algorithm. We set the value of **C** and **I** on four levels to be 0.3, 1, 5 and 10 to check its influence on the performance.

In the following we compare the recommendation precision of these four levels.

Table 6.4: The Influence of Values of **C** and **I**

y	0.3	1	5	10
Precision(%)	19.6	24.6	13.7	11.0

From Table 6.4 we see that the recommendation precision decreases rapidly as we increase the value of **C** and **I**. On the other hand, if it is too small, the performance also goes down as the system becomes unstable around the poles of the logarithmic terms in Eq. (6.12). This indicates that we should choose the value of **I** and **C** around 1 for precise calculation.

6.6 Conclusion

In this chapter, we develop a two-stage friend recommendation scenario utilizing multimedia information. In the first stage, tag information is utilised to build a tag-similarity network and is aligned to a contact network by a number of important features to generate a "possible friend list". In the second stage, a topic model is proposed and a new method based on series expansion is developed to combine image features and contact information to make more precise recommendations.

The experimental results show that the proposed method outperforms other methods in friend recommendation in that our method achieves the highest precision and recall in friend prediction. The network alignment of Stage One is effective. The topic model in Stage Two refines the result of stage one and the new series expansion method has better performance than the traditional variational and Gibbs sampling methods.

In the future, we will further develop our algorithm in the following aspects. 1) We will extend our ideas to further applications such as product recommendation, media retrieval, etc. 2) the current series expansion method, though effective, is still a little complicated mathematically, and we will refine the idea to make it more compact and extendible. 3) we will develop other algorithms in each of our two stages to achieve better, more acceptable recommendations. Specifically, we will introduce the concept of deep learning in our scenario for more efficient feature learning.

Chapter 7

Staged Social Friend Recommendation Based on Deep Neural Network

Deep learning framework has developed rapidly in this decade and made great achievements in different fields such as image classification, object detection, natural language processing, etc. As a tool that is very suitable for big data processing, it has been also applied in the field of social media. In this chapter we utilise the deep learning framework in the task of online social relationship prediction in a staged manner. In the first stage, a classical CNN network is applied to extract some representative image features. In the second stage, the features obtained from the first stage are fed into another Deep Neural Network (DNN), whose output labels are obtained from the result of a deep learning based clustering algorithm. The second stage can correlate the image features and the online friendships. We make a summary of the comparison of friend recommendation methods proposed in this thesis.

7.1 Introduction

In this chapter, we will further discuss the online friendship between two individuals. In the previous chapters, feature selection is an important tool for the studies in the field of social media, for it reduces the high dimensional social data and simplify the original problem. Feature extraction, as another widely used dimension reduction method, transforms the original feature space into a lower dimensional, newly-created space. Compared with feature selection method, it usually has higher discriminating power as a trade off with the interpretability of features (Hira & Gillies 2015).

Deep learning framework, as a special case of feature extraction, has attracted the interests of a large number of researchers and made great achievements in this decade. It provides a very successful tool in dealing with a large amount of data and has been applied in the fields of image recognition (Wu & Chen 2015), natural language processing (Sarikaya et al. 2014), information retrieval (Lam, Nguyen, Nguyen & Nguyen 2015), and bioinformatics (Ibrahim, Yousri, Ismail & El-Makky 2014), etc. In general, it refers to a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for pattern classification, feature extraction or representation learning (Deng 2014). A deep neural network usually contains many layers and a great amount of parameters, which makes it quite applicable in large scales and complex problems that traditional methods can hardly deal with. Online recommendation systems, which may contain tens of thousands of users as well as more attributes and relationships, have benefited a lot from the development of the deep learning framework. The applications include content-based video (Covington, Adams & Sargin 2016) and music (van den Oord et al. 2013) recommendation, matrix-factorization based file recommendation (Deng et al. 2016), and social link prediction (Liu, Liu, Sun, Liu & Wang 2013). In (Wang, Wang & Yeung 2015), a deep-learning-based collaborative filtering (Wang, Wang & Yeung 2015) is also developed for general recommendation tasks.

Though the deep learning framework has been successfully applied in recommendation system, seldom has considered the friend recommendation task. Some recent methods that have been proposed for link prediction task such as (Liu, Liu, Sun, Liu & Wang 2015) and (Xiaoyi Li 2014) only take the link information of the network into consideration, for example, the degrees or the neighbours of the nodes in the network. On the other hand, as illustrated in the previous chapters, the images and texts contain ample information for friendships between online friends. The problem is, how to extract the helpful information with the deep neural network? In the following, we will propose a content-based friend recommendation algorithm with the help of deep learning framework.

As the friend recommendation is a complex task and different factors are considered, we develop a two-staged framework that in each stage a deep neural network is presented. In the first stage, a widely-used Convolutional Neural Network (CNN) is applied to extract the high level features of images from each user. In the second stage, the features extracted from the last layer of the first stage is feed into another Deep Neural Network, whose output are the cluster label of the user who uploads the original images. The cluster label is obtained from a deep-learning-based graph clustering method, which to some extent contains the information about the relationships between users.

To be specific, in the first stage, we utilise a widely-used CNN network to extract the features that correlate the tags and images. As there is usually many tags for one photo on Flickr, a multi-label CNN is required. The CNN is first proposed for image classification (Krizhevsky et al. 2012*a*), soon it is applied in the multi-label problems (Wei, Xia, Huang, Ni, Dong, Zhao & Yan 2014). In this chapter, following (Frogner, Zhang, Mobahi, Araya & Poggio 2015) we implement a multi-label framework based on Caffe (Jia, Shelhamer, Donahue, Karayev, Long, Girshick, Guadarrama & Darrell 2014*b*). As the tag set on Flickr contains much noise, some basic tag-cleaning work is applied as a pre-processing step.

The features obtained from the first stage correlated the information from the image and text domain, but do not directly relate to the friend recommendation task. In the second stage, to obtain the friendship information, we first cluster the existing friendship network to obtain some cluster labels. For the friendship network, the nodes are the users and the edges are the friendships between users. The clustering task is purely based on the network structure, and utilises a DNN that are specifically designed for graph structure (Cao, Lu & Xu 2016*a*).

With the clustering labels as the output, we design another DNN, whose input is the features extracted from the last layer of the CNN in the first stage. As the clustering labels are highly related to the relationships between individuals, the DNN may extract some high level features that are instructive for friend recommendation.

As the last chapter of this thesis, in the experimental part, we summarize all the methods proposed in this thesis and make comprehensive experiments for all the proposed methods in this thesis, as well as some analysis based on the experimental results.

The rest of the chapter is organized as follows: Section 7.2 reviews some state-of-the-art methods that related to the proposed method. Section 7.3 provides the framework of the whole algorithm. Section 7.4 introduces the details of the CNN in the first each stage. Section 7.5 introduces the DNN-based graph clustering method, as well as the DNN friendship-related feature learning method. Section 7.6 makes comprehensive experiments to show the advantages of the proposed methods, as well as summarises all the experiments proposed in this thesis. The last section concludes the method of this chapter.

7.2 Related Work

Some general methods about deep learning have been reviewed in 2.2.1 and 2.2.3. In this section we will review some more specific methods in the deep learning framework.

Since Hinton developed a layer-wise training method for DNN (Hinton & Salakhutdinov 2006)(Hinton et al. 2006), deep learning method has been applied successfully in different fields. At first it is designed for binary classification problem, recently many multi class/label problems has also adopted some DNN frameworks. The first stage of our algorithm, as mentioned in Chapter **??**, is a multi-lcieabel DNN problem. In the following we will review the important works related to this task.

7.2.1 Multi-Label Deep Learning

The multi-label problem has many applications in the social network (Wang & Sukthankar 2013), because most of the real social media websites provide many labels or descriptions for one object: Flickr, Youtube, Instagram, etc. DNN has also been designed for multi-label problem. In (Wei, Xia, Huang, Ni, Dong, Zhao & Yan 2014), the authors propose a flexible deep CNN infrastructure, where some object segment hypotheses are taken as the inputs of a shared CNN, then the CNN outputs from different hypotheses are aggregated with max pooling to make the multi-label predictions. (Huang, Wang, Wang & Tan 2013) deals with the problem with a different way by applying a DBN structure and define the learning of each label as a binary classification task, and the multi-label learning is accordingly transformed to multiple binary classification learning. It defines the output layer with multiple different aims. In (Read & Hollmén 2014), a classifier chain is applied where the prediction of binary classifiers are cascaded along a chain as additional features, and a DBN is utilised to implement the classifier chain. (Read & Perez-Cruz 2014) goes further by applying a DBN to study the dependencies between labels, as well as training a multi-label classifier. (Wang, Yang, Mao, Huang, Huang & Xu 2016) also considers the dependency among labels by a utilising a Recurrent Neural Network (RNN) to learn a joint image-label embedding to characterize the semantic label dependency.

One important problem for multi-label deep learning is how to define the loss function of the network. In (Gong, Jia, Leung, Toshev & Ioffe 2013), the authors implement several multi-label loss functions including softmax (Guillaumin, Mensink, Verbeek & Schmid 2009), pairwise ranking (Joachims 2002), and weighted approximated ranking (WARP) (Weston, Bengio & Usunier 2011). According to this work, WARP has the best performance, which specifically optimizes the top-k accuracy for annotation by using a stochastic sampling approach. In (Frogner et al. 2015), a new loss function called Wasserstein distance loss is proposed, which provides a natural notion of dissimilarity for probability measures. In the first stage of our work, we adopt the Wasserstein distance for multi-label prediction.

7.2.2 Link Prediction based on DNN

In the second stage of the proposed method, a DNN based graph clustering method is required. In the following we will review the existing algorithms related to this problem

The social link prediction problem (Li, Gao, Guo, Du, Li & Zhang 2014), as well as the community detection problem (Abdelbary, ElKorany & Bahgat 2014) have attracted some attentions recently, with the rapid development of the online social network. Many of them are content based (Abdelbary et al. 2014), which is not fit for our requirement. Some graph based link prediction methods, which means that the known information is only the links between users, are also presented recently, and the deep learning framework has been applied to solve this problem. (Shalforoushan & Jalali 2015) investigates the hidden principles that drive the behaviours of social members via the DBN. (Li, Tarlow, Brockschmidt & Zemel 2015) proposes a complex graph structure prediction method based on a gated graph neural network. On AAAI'16, (Cao, Lu & Xu 2016*b*) provides a DNN graph presentation that learns the important feature representations from the network matrix. Furthermore, (von Winckel 2014) applies the DNN into a dynamic network to make time-sensitive link prediction.

In summary, the structure and the links of a network can provide ample information for link prediction and a deep structure can be utilised to solve such problem. In this chapter, we adopt the method proposed in (Cao et al. 2016*b*) to cluster the users as a preliminary step for friend related feature learning in the second stage of the proposed algorithm. In this method, the links with other individuals are treated as original features and a deep audoencoder is applied to extract the high-level features from the original ones.

7.3 Proppsed System Framework

As mentioned in 7.1, the proposed method is a two staged method that in the first stage, the features that correlate the text and image domains are learned from a CNN. In the second stage, the friend related features are learned from a DNN where the input is the features obtained from the first stage, and the output is the clustering result from a network. The framework is shown in Figure 7.1.

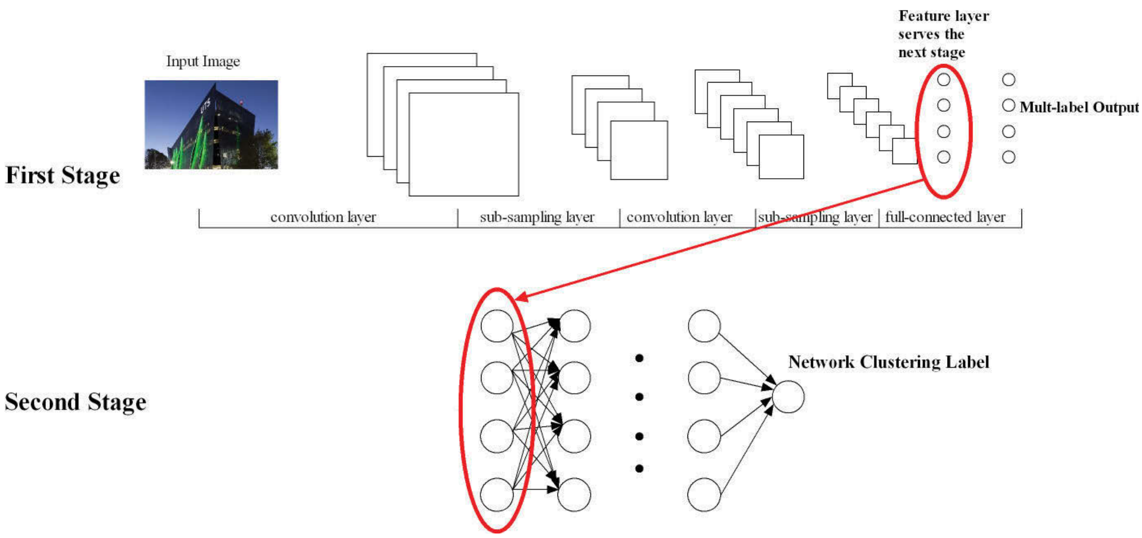


Figure 7.1: Staged Deep Friend Recommendation Framework

In the first stage, as the dataset we apply is from Flickr, where each image usually contains many tags, we apply a mutli-label CNN to learn the representative features. Mainly we utilise the Wasserstein loss, which has been proposed (Frogner et al. 2015). Compared with some other widely used multi-label loss functions such as KL divergence, the Wasserstein loss takes the correlation between labels into consideration, and thus achieves better performance. A Caffe implementation of the deep Wasserstein loss is publised by Frogner on Github[1]. In this step the mutli-lable based features are extracted.

In the second stage, we first obtain the clustering information of the users via a graph based DNN, which is based on the realization of (Cao et al. 2016*b*). The motivation for this step is that we assume that the clustering of the network contains some important information for the friend making behaviours of the online users. So the clustering results can be viewed as labels. Then we utilise a DNN whose input is the features extracted from the first stage, and the output is the clustering labels. The essential idea of (Cao et al. 2016*b*) is that the links with other users can be viewed as original features, and a DNN can extract high level features with an autoendoder. Through these two stages the features for friend recommendation is obtained. In the following we will go to the details about the each stage.

7.4 First Stage: Deep Wasserstein Loss

Learning to predict multi-label outputs is challenging, but in many problems there is a natural metric on the outputs that can be used to improve predictions. In this chapter we utilise a loss function for multi-label learning, based on the Wasserstein distance, which considers the relationships between labels.

Some widely used loss function such as Kullback-Leibler (KL) divergence can also be applied in multi-label problems. However, most of these functions

[1]https://github.com/frogner/caffe/tree/wasserstein

assumes that the labels are independent from each other, which is not true in reality. Different labels might lead to nearly-equivalent semantic meanings and similar appearance. For example, two images tagged with "aerobus" and "Boeing 747" might looks similar, but if we assume that the labels are independent from each other, we can not acquire this similarity semantically. Wasserstein distance loss can be applied to alleviate this problem.

Wasserstein distance can be defined as the cost of optimal transport plan for moving the mass in the predicted measure to match that in the target (Frogner et al. 2015). Mathematically, consider two families $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_m)$ in an arbitrary space Ω. When $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\upsilon = \sum_{i=1}^{m} b_i \delta_{y_i}$, the Wasserstein distance between μ and υ is the optimum of a network flow problem known as the *transportation problem* (Frogner et al. 2015). Also a distance matrix $\mathbf{M}_{xy}$ is defined as a cost parameter as $\mathbf{M}_{XY} = [D(x_i, y_j)]_{ij} \in \mathbb{R}^{n \times m}$.

A transportation polytobe $U(\mathbf{a}, \mathbf{b})$ is defined as the set of $n \times m$ nonnegative matrix such that their rows and column marginals are equal to $\mathbf{a}$ and $\mathbf{b}$ respectively. Define 1_n as the n-dimensional vector of ones, we can include all the transportations from $\mathbf{a}$ to $\mathbf{b}$ in the polytobe U as: $U(\mathbf{a}, \mathbf{b}) = \{T \in \mathbb{R}_{+}^{n \times m} \| T 1_m = \mathbf{a}, T^T 1_n = \mathbf{b}\}$.

Define $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}^T \mathbf{B})$ as the Frobenius dot-product of matrices, Wasserstein distance is defined mathematically as follows:

$$W(\mu, \delta) = \min_{T \in U(\mathbf{a}, \mathbf{b})} \langle T, M_{XY} \rangle \tag{7.1}$$

The Wasserstein distance W is the loss function we should optimize in the DNN.

Unfortunately, computing a subgradient of the exact Wasserstein loss is quite costly, as follows. The exact Wasserstein loss is a linear program and a subgradient of its solution can be computed using Lagrange duality. (Frogner et al. 2015) proposes a good approximation to make an efficient calculation and they provide a DNN Caffe based implementation. We adopt it and make some necessary modifications.

In this way, in the first stage the representative features that correlate the image and text domains are obtained.

7.5 Second Stage: DNN based Friend Recommendation

7.5.1 DNN based Network Clustering

In the second stage, the purpose is to obtain some friendship related features for the final recommendation task. To utilise the DNN framework, on important question is how to express the friendship in a proper way for further feature learning. In the real and online society, it is natural that many friends are likely to communicate with each other or share experiences in some groups ore communities. This triggers the idea that we should first find the community or clustering structure of the network. As the social network is a complex network, it has some central structures that are relatively easy to be clustered.

We adopt the method proposed in (Cao et al. 2016*a*) to find the clusters in the social network, where only the link information of the network is available. The essential problem of (Cao et al. 2016*a*) is to learn representative features of nodes from a complex graph structure. In general, it contains three major steps as follows.

In the first step, a random walk procedure starts from each node. The path from each node is recorded.

In the second step, a pointwise mutual information matrix is formed according the record obtained in the first step. This matrix carries the relationships between any two nodes in the graph.

In the final step, a deep autoencode is applied to learning the clustering features. The input and output of the autoencode is the matrix obtained in the second step.

With the three steps, the instructive features that are highly related to the graph structure is learned, and the features can be used for clustering tasks.

A implementation is available on Gitub [2], and we make some modifications to fit our problem. In this way the clustering label is obtained.

7.5.2 Friendship related Feature Learning with DNN

With the clustering label information, we go further to train a DNN network that the input is the features extracted from the first stage, and the output is the clustering labels. This is in general a single-label, multi-class problem, In this situation, the widely used Softmax function can be utilised as the loss function. we implement it with the existing DNN tools Tensorflow[3].

The final recommendation can be made simply by comparing the similarity of the features learned from the last layer of the DNN.

7.6 Overall Experiments

In this section, we will make comprehensive experiments of the method proposed in this chapter. Also we will make a summary of all the methods proposed in this thesis, and discuss some details of the algorithm according to the experimental results. We run most of the experiments on the Titan cluster 16 core I7, 128G memory and Nvidix 1G graphical memory. For the deep learning part, the Caffe framework in C++ is applied in the first stage and the DNN friendship-related feature learning in the second stage. Also the Keras framework [4] in Python is applied for the clustering task in the second stage.

[2]https://github.com/ShelsonCao/DNGR
[3]www.tensorflow.org
[4]https://keras.io/

7.6.1 Experimental Settings

The general experimental settings are similar as in the previous chapters. One thing worthy of mentioning is that for friend recommendation, when a new user comes into the network with only some tags and without photos, these tags are severed as the output side of the well-trained CNN in the first stage. Then we apply the back propagation method to obtain the features to serve as the input of the second stage.

Proposed algorithms in this theses

The following is a summary of the methods proposed in this thesis.

- text-based network alignment method (abbr. NC-based SFR, Chapter 4)
- text-and-image-based two staged method: alignment and co-clustering (abbr. SRCC, Chapter 5)
- text-and-image-based method, utilising probabilistic topic modelling (abbr. PTM-SE, Chapter 6)
- text-and-image-based method, utilising different DNN networks (abbr. DR-SFR, Chapter 7)

Reference Friend Recommendation Methods

The following is a summary of the state-of-the-art reference methods in the experiments:

To check the advantage of our method, we compare our recommendation algorithms with several state-of-the-art recommendation systems. The first one is based on matrix factorization(MF). MF method decomposes the item-user or user-user matrix to infer the latent factors that catch individuals' interests and has been widely discussed for different kinds of recommendation prolems (Ma et al. 2008*b*)(Guo et al. 2015). In this thesis we apply a recent method proposed in (Guo et al. 2015) for comparison, for it jointly considers the information from two different domains.

Another recent method is based on Bayesian collaborative filtering that takes the social connections into account, called SBPR (Zhao et al. 2014). As a widely-used recommendation method, collaborative filtering assumes that two users that choose the same items behave similar on other items. Traditional collaborative filtering methods do not consider much about the social connections between users. SBPR removes this drawback by taking the social connections into account by assigning a social coefficient to each user. [5].

At last we consider a multi-network based algorithm for comparison. When considering social multiple network problems, transition probability propagation is a method that is frequently used (Jiang et al. 2012)(Liu et al. 2012). We choose (Jiang et al. 2012) as a reference method for the following reasons: 1) It considers the relationships of different networks, which is similar to our idea; 2) It uses the information of other networks for recommendation, which again has some similarities with ours. (Jiang et al. 2012) enhances the links in one network and between different networks using a random walk propagation method. After a sufficient number of walks, it obtains the modified link weights between each user pair. We use the weights for friend recommendation.

7.6.2 Experimental Results and Discussions

We compare the proposed DNN based friend recommendation method with all the previously proposed methods, as well as some reference methods.

The overall friend recommendation precision performance is given in Figure 7.2.

From Fig. 7.2 we can conclude that the DNN based staged recommendation method achieves the best performance. This is mainly due to that

[5] The realization of (Guo et al. 2015) and (Zhao et al. 2014) is based on the existing open-source Java package LibRec at http://www.librec.net/

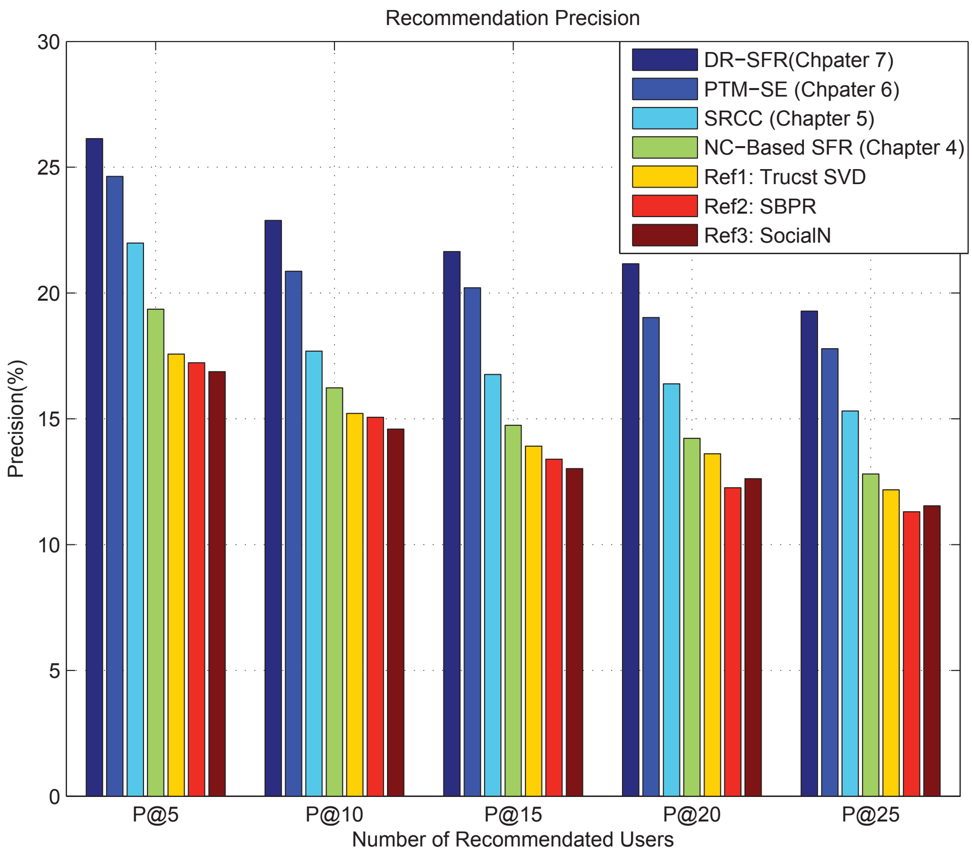


Figure 7.2: Friend Recommendation Precision Comparison in This Thesis

the DNN can learning the feature representation quite efficient. The clustering step in the second stage can catch the friendship-related information correctly.

The PTM based method achieves the second best performance, for it learns the model with a precise series expansion. However, it still has some limitation for its Gaussian assumption and mathematical complexity.

In the following we also draw the precision-recall curve to compare the performance of different algorithms. The results are shown in Figure 7.3:

Again we see the advantage of the proposed DNN based method. The

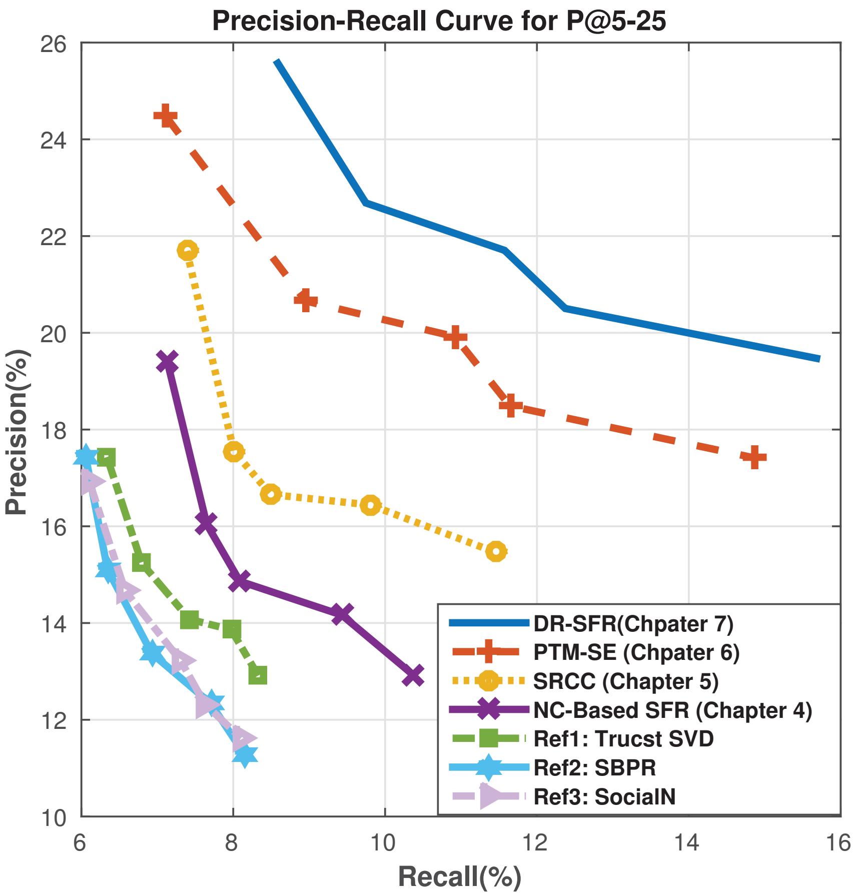


Figure 7.3: Overall Friend Recommendation Precision-Recall Curve

performance of F-measure is shown in Table 7.1

Table 7.1: F-Measure of Friend Recommendation

	P@5	P@10	P@15	P@20	p@25
DR-SFR	0.127	0.138	0.152	0.156	0.173
PTM-SE	0.111	0.125	0.141	0.142	0.159
SRCC	0.109	0.111	0.114	0.123	0.131
NC-Based SFR	0.104	0.105	0.106	0.115	0.118
Ref1:Turst SVD	0.0.0953	0.0941	0.0964	0.0997	0.101
Ref2: SBPR	0.0902	0.0907	0.0938	0.0952	0.0968
Ref3: SocialN	0.0909	0.0922	0.0935	0.0940	0.0957

In summary, all the proposed methods have better friend recommendation performance, compared with the state-of-the-art reference methods. The DNN based method has the best performance in recommendation precision and recall, the reason might be that the deep structure has the ability to catch the internal information of the social multimedia network. Staged methods(PTM-SE and SRCC) has better performance than single stage method(NC-based SFR), and the probabilistic-topic-model based method has better performance than co-clustering based method.

7.7 Conclusion

In this chapter, we propose a staged DNN framework for social friend recommendation. To utilise the text and image information simultaneously, in the first stage, a CNN is applied to obtain the features that correlate the information from both domain. In the second stage, the social network is first clustered via a DNN to obtain the labels that are related to social friendship. Then a DBN is utilised for important friendship-related feature learning. We make comprehensive experiments to show the advantages of the proposed method. We also summarise the experimental results of all the methods proposed in this thesis.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis mainly presents several online friend recommendation methods in the multimedia environment. These methods can be developed further for social websites to improve the friend recommendation accuracy and we hope these methods can help the sociologists to have a deeper understanding of the online social networks, as well as to help the economist to better interpret the online economy.

As one important method for social friend recommendation is to find the instructive features for friend recommendation, in Chapter 3 a general feature selection algorithm is develop for different kinds of datasets. The feature selection method can both increase the system performance as well as reduce the computational complexity. We introduce the concept of global and local structure preservation into the feature selection problem. Consequently, the selected features can both keeps the similarity relationships between different items(local structure preservation), and help to make the get the precise classification/clusterings (global structure preservation). In comparison of some state-of-the-art feature selection algorithms, the proposed method achieves the highest performance.

In Chapter 4 we extend the basic idea of feature selection method mentioned in Chapter 3 to the social friend recommendation problem. For text-based recommendation, we propose a projection method that project the data with high dimensional feature space into the low dimensional feature space, during which the similarity relationship between nodes and the eigen presentation of the data are kept. In this way, different social network are aligned to be as similar as possible via feature selection. The selected features can be viewed as the instructive features that help individuals to find online friends more easily and convincing. We also make a discussion in Chapter 4 about the extension of the network alignment method in the problem of more than two networks.

Then we start to take the image information into consideration for the recommendation task in Chapter 5. We adopt a two-stage recommendation scenario that to consider the text information in the first stage and the image information in the second stage. The main reasons for the two stage framework are two folds: first, the image information is usually much noisy for friend recommendation task, so we utilize it as an additional material to refine the result of the text-based recommendation; second, the two stage method can reduce the complexity of the system, as well as help us to understand the contribution of each data domain. For image information in the second stage, we apply a three-way co-clustering method that utilise all the information from text, image, and friendship domain for a comprehensive recommendation result. The proposed method further improves the recommendation accuracy compared with one stage method discussed in Chapter 3.

Chapter 6 provides a probabilistic topic model, instead of the co-clustering method proposed in Chapter 5, to refine the recommendation performance in the second stage. One drawback of co-clustering based method in the recommendation task is that it can not provide quantitative measurements to illustrate the similarities between individuals, and probabilistic topic model can calculate the similarities and rank the intimacies between friends easily, through its latent variable expressions.

The solution of a probabilistic topic model is usually not easy to obtain because it involves the integral of correlation of different random variables. Traditional methods such as Gibbs sampling and variational methods have their own advantages and disadvantages. In Chapter 6, we propose a novel method to solve the integral, based on the series expansion expression of the integral. The series expansion approach can give a precise analytical solution of the integral and to some extent balance the requirement of the computational speed and the accuracy, and thus more flexible and adaptive for different tasks and requirements. The experimental results show the advantage of the proposed series expansion method.

Each of the chapters from Chapter 3 to Chapter 6 of this thesis is supported by one conference or journal papers[1] listed in **List of Publications**. To conclude, the works presented in this thesis are helpful for the research and design of recommendation systems, as well as for the social websites to improve their recommendation methods.

8.2 Future Work

Recommendation systems have a wide application in the age of Internet. In future, we will continue to explore the possible directions, which will be both theory-driven and application-driven. I think the following directions are worthy to be studied further.

For application-driven problems, we are interested in the following topics:

1. **Item/Product Recommendation Based on Relationships**: In this thesis we concentrate on friend recommendation, which is an important topic for the study of online society. However, item/product recommendation may bring more direct commercial profits and thus has attracted more attentions. (Alex 2012) It is an interesting problem that how the online friendship influences the market, and have been

[1]The papers related to Chapter 3, 4 and 5 are published, and the paper of Chapter 6 is under the second round review

studied by some previous researches(Song et al. 2013). From our point of view, these pilot researches do not make a deep analysis of the relationship between friendships and shopping behaviours. Based on our previous study, we have a plan to go further in this topic, and seek for some commercial profits upon online friendships.

2. **Real time Friend Recommendation**: Existing friend recommendation methods such as (Min et al. 2015)(Huang et al. 2015) seldom support real time recommendation. Also many of them do not have a mechanism for online update. In this thesis we also do not take the time as an important constraint. This is partly because that friend recommendation usually does not consider real time reactions as an urgent demand in reality. However, it is a interesting problem to build some time-sensitive models for friend recommendation, both in short-term and long term. It helps us to understand the change of the online friendship as the time goes, as well as to give some hints about how to make online time-sensitive product recommendations.

For theory-driven problems, we think the following problems are important and may lead to greater contributions.

1. **Series Expansion method**: In Chapter 6, a series expansion method is proposed to find the solution of a probabilistic topic model, and shows its advantages over widely-used Gibbs sampling or variational methods. However, there are still many topics that are not fully studied about this method: how to generalize it to more complex models, and how to apply it to other distributions. Another topic in this direction is that how to simplify the mathematical deducing procedure so it might become a more general, easy-to-handle method that can be applied to solve most of the problems in probabilistic topic models.

2. **Deep Learning Framework in Friend Recommendation**: Deep learning framework has been widely applied in the fields of image

processing (Song et al. 2013) and natural language processing (Min et al. 2015) and have made great achievements (Min et al. 2015). Recommendation system has also utilised it as a general feature learning tool (Min et al. 2015). However, to what extent the deep learning framework can help to dig the social media and the friendship between individuals, is still an open topic. In our opinion, this is a important topic that might lead to a great success in social media. The problems are, how to express the social friendship in a proper way that the deep learning framework can make the calculations properly, and how the features extracted from the deep network can be properly applied for further works. we will study all these problems in our future study.

Bibliography

Abdelbary, H. A., ElKorany, A. M. & Bahgat, R. (2014), Utilizing deep learning for content-based community detection, *in* '2014 Science and Information Conference', pp. 777–784.

Ahlfors, L. (1979), *Complex Analysis*, McGraw Hill.

Ahmed, K. W., Rashid, M. M., Hasan, M. K. & Mahmud, H. (2015), Cohesion based personalized community recommendation system, *in* '2015 18th International Conference on Computer and Information Technology (ICCIT)', pp. 32–37.

Ahmed, S., Yang, L.-L. & Hanzo, L. (2011), Probability distributions of products of rayleigh and nakagami-m variables using mellin transform, *in* 'Communications (ICC), 2011 IEEE International Conference on', Kyoto, Japan, pp. 1–5.

Alex, W. (2012), 'Why is it hard to make friends over 30?', *The New York Times* .

Alsaedi, N. & Burnap, P. (2015), Feature extraction and analysis for identifying disruptive events from social media, *in* '2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)', pp. 1495–1502.

Azadifar, S. & Monadjemi, S. A. (2015), Feature selection using social network techniques, *in* 'Information and Knowledge Technology (IKT), 2015 7th Conference on', pp. 1–6.

Balser, W. (1994), *From Divergent Power Series to Analytic Functions*, Springer-Verlag.

Bao, B.-K., Min, W., Lu, K. & Xu, C. (2013), Social event detection with robust high-order co-clustering, *in* 'ICMR'13', New York, NY, pp. 135–142.

Barnett, E. & Casper, M. (2001), 'A defination of 'social environment'', *American Journal of Public Health* **91**(3).

Bateman, H. (1954), *Tables of Integral Transforms*, Vol. 1, New York: McGraw-Hill.

Bayati, M., Gerritsen, M., Gleich, D., Saberi, A. & Wang, Y. (2009), Algorithms for large, sparse network alignment problems, *in* 'ICDM'09', Miami, FL, pp. 705–710.

Bellman, R. E. (2003), *Dynamic Programming*, Courier Dover Publications.

Berkovsky, S., Kuflik, T. & Ricci, F. (2007), Cross-domain mediation in collaborative filtering, *in* 'Proceedings of the 11th International Conference on User Modeling', UM '07, Springer-Verlag, Berlin, Heidelberg, pp. 355–359.

Bichot, C.-E. (2010), 'Co-clustering documents and words by minimizing the normalized cut objective function', *Journal of Mathematical Modelling and Algorithms* **9**(2), 131–147.

Bishop, C. M. (2006), *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Blei, D. M. (2012*a*), 'Probabilistic topic models', *Commun. ACM* **55**(4), 77–84.

Blei, D. M. (2012*b*), 'Probabilistic topic models', *Commun. ACM* **55**(4), 77–84.

Blei, D. M. & Lafferty, J. D. (2006), Correlated topic models, *in* 'In Proceedings of the 23rd International Conference on Machine Learning', MIT Press, pp. 113–120.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003*a*), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003*b*), Latent dirichlet allocation, *in* 'Journal of Machine Learning Research'.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003*c*), 'Latent dirichlet allocation', *J. Mach. Learn. Res.* **3**, 993–1022.

Bobadilla, J., Ortega, F., Hernando, A. & GutiéRrez, A. (2013*a*), 'Recommender systems survey', *Know.-Based Syst.* **46**, 109–132.

Bobadilla, J., Ortega, F., Hernando, A. & GutiéRrez, A. (2013*b*), 'Recommender systems survey', *Know.-Based Syst.* **46**, 109–132.

Borrís, J., Moreno, A. & Valls, A. (2014), 'Review: Intelligent tourism recommender systems: A survey', *Expert Syst. Appl.* **41**(16), 7370–7389.

Breese, J. S., Heckerman, D. & Kadie, C. (1998), Empirical analysis of predictive algorithms for collaborative filtering, *in* 'Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence', UAI'98, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43–52.

Bro, R. (1997), 'Parafac. tutorial and applications', *Chemometrics and Intelligent Laboratory Systems* **38**(2), 149 – 171.

Brusilovsky, P., Kobsa, A. & Nejdl, W., eds (2007), *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer-Verlag, Berlin, Heidelberg.

Brym, R. (2009), *Sociology: Your Compass for a New World, Brief Edition*, Cengage Learning.

Cai, D., Xhang, C. & He, X. (2010), Unsupervised feature selection for multi-cluster data, *in* 'ACM SIGKDD'10', Washington DC, USA, pp. 333–342.

Cantador, I., Fernández-Tobías, I., Berkovsky, S. & Cremonesi, P. (2015), Cross-domain recommender systems, *in* 'Recommender Systems Handbook', Springer, pp. 919–959.

Cao, S., Lu, W. & Xu, Q. (2016*a*), 'Deep neural networks for learning graph representations'.

Cao, S., Lu, W. & Xu, Q. (2016*b*), 'Deep neural networks for learning graph representations'.

Carrer-Neto, W., Hernández-Alcaraz, M. L., Valencia-García, R. & García-Sánchez, F. (2012), 'Social knowledge-based recommender system. application to the movies domain', *Expert Syst. Appl.* **39**(12), 10990–11000.

Carter, B. D. & Springer, M. D. (1977), 'The distribution of products, quotients and powers of independent h-fucntion variates', *SIAM Journal on Applied Mathematics* **33**(4), 542–558.

Carullo, G., Castiglione, A. & Santis, A. D. (2014), Friendship recommendations in online social networks, *in* 'Intelligent Networking and Collaborative Systems (INCoS), 2014 International Conference on', pp. 42–48.

Charrad, M. & Ahmed, M. B. (2011), Simultaneous clustering: A survey, *in* 'Proceedings of the 4th International Conference on Pattern Recognition and Machine Intelligence', PReMI'11, Springer-Verlag, Berlin, Heidelberg, pp. 370–375.

Charrad, M., Lechevallier, Y., Ahmed, M. b. & Saporta, G. (2009), *Block Clustering for Web Pages Categorization*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 260–267.

Chen, H., II, A. G. O. & Giles, C. L. (2015), 'Expertseer: a keyphrase based expert recommender for digital libraries', *CoRR* **abs/1511.02058**.

Chen, J., Geyer, W., Dugan, C., Muller, M. & Guy, I. (2009), Make new friends, but keep the old: Recommending people on social networking sites, *in* 'ACM CHI'09', New York, NY.

Chen, K. H., Ting, Z. Y., Shen, J. Y., Hu, Y. J. & Liang, T. (2015), A two-stage learning method for response prediction, *in* '2015 IEEE International Conference on Data Mining Workshop (ICDMW)', pp. 1336–1341.

Chen, R. & Li, T. H. (1995), 'Blind restoration of linearly degraded discrete signals by gibbs sampling', *IEEE Transactions on Signal Processing* **43**(10), 2410–2413.

Chen, X., Hu, X., Lim, T. Y., Shen, X., Park, E. K. & Rosen, G. L. (2012), 'Exploiting the functional and taxonomic structure of genomic data by probabilistic topic modeling', *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **9**(4), 980–991.

Chen, Y.-Y., Cheng, A.-J. & Hsu, W. H. (2013), 'Travel recommendation by mining people attributes and travel group types from community-contributed photos', *IEEE Trans. on Multimedia* **15**, 1283–1295.

Chicco, D., Sadowski, P. & Baldi, P. (2014), Deep autoencoder neural networks for gene ontology annotation predictions, *in* 'Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics', BCB '14, pp. 533–540.

Chin, A., Xu, B. & Wang, H. (2013), Who should i add as a "friend"?: A study of friend recommendations using proximity and homophily, *in* 'Proceedings of the 4th International Workshop on Modeling Social Media', MSM '13, ACM, Paris, France, pp. 7:1–7:7.

Chung, R., Sundaram, D. & Srinivasan, A. (2007), Integrated personal recommender systems, *in* 'Proceedings of the Ninth International Conference on Electronic Commerce', ICEC '07, ACM, New York, NY, USA, pp. 65–74.

Cilibrasi, R. & Vitanyi, P. M. B. (2007), 'The Google similarity distance', *IEEE Trans. Knowledge and Data Engineering* **19**(3), 370–383.

Ciresan, D. C., Meier, U. & Schmidhuber, J. (2012), 'Multi-column deep neural networks for image classification', *CoRR* **abs/1202.2745**.

Conte, D., Foggia, P. P., Sansone, C. & Vento, M. (2004), 'Thirty years of graph matching in pattern recognition', *Intl. J. of Pattern Recognition and Artificial Intelligence* **18**(3), 265–298.

Covington, P., Adams, J. & Sargin, E. (2016), Deep neural networks for youtube recommendations, *in* 'Proceedings of the 10th ACM Conference on Recommender Systems', New York, NY, USA.

Dao, M. D., Rangamani, A., Chin, S. P., Nguyen, N. P. & Tran, T. D. (2015), Targeted dot product representation for friend recommendation in online social networks, *in* '2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)', pp. 349–356.

Daud, A., Li, J., Zhou, L. & Muhammad, F. (2010), 'Knowledge discovery through directed probabilistic topic models: a survey', *Frontiers of computer science in China* **4**(2), 280–301.

Deng, L. (2014), 'A tutorial survey of architectures, algorithms, and applications for deep learning', *APSIPA Transactions on Signal and Information Processing* **3**, e2.

Deng, S., Huang, L., Xu, G., Wu, X. & Wu, Z. (2016), 'On deep learning for trust-aware recommendations in social networks', *IEEE Transactions on Neural Networks and Learning Systems* **PP**(99), 1–14.

Dhillon, I., Mallela, S. & Nodha, D. (2003), Information-theoretic co-clustering, *in* 'SIGKDD'01', New York, NY.

Ding, R., Zhu, J., Tang, Y., Lin, X., Xiao, D. & Dong, H. (2016*a*), A novel feature selection strategy for friends recommendation, *in* '2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)', pp. 123–128.

Ding, R., Zhu, J., Tang, Y., Lin, X., Xiao, D. & Dong, H. (2016*b*), A novel feature selection strategy for friends recommendation, *in* '2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD)', pp. 123–128.

Dolatkia, I. & Azimzadeh, F. (2016), Music recommendation system based on the continuous combination of contextual information, *in* '2016 Second International Conference on Web Research (ICWR)', pp. 108–114.

Dy, J. G. & Brodley, C. E. (2004), 'Feature selection for unsupervised learning', *J. Mach. Learn. Res.* **5**, 845–889.

Eleftheriadis, S., Rudovic, O. & Pantic, M. (2016), 'Joint facial action unit detection and feature fusion: A multi-conditional learning approach', *IEEE Transactions on Image Processing* **25**(12), 5727–5742.

E.L.Lady (1998), 'Convergence of infinite series in general and taylor series in particular'.

Erosheva, E., Fienberg, S. & Lafferty, J. (2004), 'Mixed-membership models of scientific publications', *Proceedings of the National Academy of Sciences* **101**(suppl 1), 5220–5227.

Fernández-Tobías, I., Cantador, I., Kaminskas, M. & Ricci, F. (2011), A generic semantic-based framework for cross-domain recommendation, *in* 'Proceedings of the 2Nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems', HetRec '11, ACM, New York, NY, USA, pp. 25–32.

Fernández-Tobías, I., Cantador, I., Kaminskas, M. & Ricci, F. (2012), Cross-domain recommender systems: A survey of the state of the art, *in* 'Spanish Conference on Information Retrieval'.

Ferrari's Solution of a Quartic Equation (n.d.), *in* 'Maths notes'.

Frogner, C., Zhang, C., Mobahi, H., Araya, M. & Poggio, T. A. (2015), Learning with a wasserstein loss, *in* 'Advances in Neural Information Processing Systems', pp. 2053–2061.

Gelman, A., Carlin, J. B., Stern, H. S. & Rubin, D. B. (2003), *Bayesian Data Analysis*, Chapman and Hall/CRC.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, gibbs distributions, and the bayesian restoration of images', *IEEE Trans. Pattern Anal. Mach. Intell.* **6**(6), 721–741.

Geng, X., Zhang, H., Bian, J. & Chua, T. S. (2015), Learning image and user features for recommendation in social networks, *in* '2015 IEEE International Conference on Computer Vision (ICCV)', pp. 4274–4282.

Ghorbani, S. & Ganjali, Y. (2012), Will you be my friend? privacy implications of accepting friendships in online social networks, *in* 'Information Society (i-Society), 2012 International Conference on', pp. 340–345.

Givon, S. & Lavrenko, V. (2009), Predicting social-tags for cold start book recommendations, *in* 'Proceedings of the Third ACM Conference on Recommender Systems', RecSys '09, ACM, New York, NY, USA, pp. 333–336.

Gong, Y., Jia, Y., Leung, T., Toshev, A. & Ioffe, S. (2013), 'Deep convolutional ranking for multilabel image annotation', *CoRR* **abs/1312.4894**.

Goodfellow, M. H., Wilson, J. & Hunt, E. (2010), Biochemical network matching and composition, *in* 'Proceedings of the 2010 EDBT/ICDT Workshops', EDBT '10, ACM, New York, NY, USA, pp. 40:1–40:7.

Gori, M., Maggini, M. & Sarti, L. (2005), 'Exact and approximate graph matching using random walks', *IEEE Trans. on Pattern Analysis and Machine Intelligence* **27**, 1100–1111.

Griffiths, T. L. & Steyvers, M. (2002), 'A probabilistic approach to semantic representation', *Proceedings of the 24th Annual Conference of the Cognitive Science Society* pp. 381–386.

Gu, Q., Li, Z. & Han, J. (2011), Joint feature selection and subspace learning, *in* 'Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two', IJCAI'11, AAAI Press, pp. 1294–1299.

Guillaumin, M., Mensink, T., Verbeek, J. & Schmid, C. (2009), Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, *in* '2009 IEEE 12th International Conference on Computer Vision', pp. 309–316.

Guo, C., Tian, X. & Mei, T. (2014), User specific friend recommendation in social media community, *in* 'Multimedia and Expo (ICME), 2014 IEEE International Conference on', Chengdu, China, pp. 1–6.

Guo, G., Zhang, J. & Yorke-Smith, N. (2015), 'Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings'.

Guo, J., Man-Wai, M. & Kung, S.-Y. (2006), Eukaryotic protein subcellular localization based on local pairwise profile alignment SVM, *in* 'IEEE 16th MLSP Workshop', pp. 391–396.

Han, J., Zhang, D., Wen, S., Guo, L., Liu, T. & Li, X. (2016), 'Two-stage learning to predict human eye fixations via sdaes', *IEEE Transactions on Cybernetics* **46**(2), 487–498.

Hannon, J., Bennett, M. & Smyth, B. (2010), Recommending twitter users to follow using content and collaborative filtering approaches, *in* 'ACM RecSys'10', Barcelona, Spain, pp. 199–206.

Hasan, M. M., Shaon, N. H., Marouf, A. A., Hasan, M. K., Mahmud, H. & Khan, M. M. (2015*a*), Friend recommendation framework for social networking sites using user's online behavior, *in* '2015 18th International Conference on Computer and Information Technology (ICCIT)', pp. 539–543.

Hasan, M. M., Shaon, N. H., Marouf, A. A., Hasan, M. K., Mahmud, H. & Khan, M. M. (2015*b*), Friend recommendation framework for social networking sites using user's online behavior, *in* '2015 18th International Conference on Computer and Information Technology (ICCIT)', pp. 539–543.

Hatammimi, J. & Sharif, O. O. (2014), Selection the way to start business based on social media features, *in* 'Technology Management and Emerging Technologies (ISTMET), 2014 International Symposium on', pp. 371–376.

He, C., Li, H., Fei, X., Tang, Y. & Zhu, J. (2015), A topic community-based method for friend recommendation in online social networks via joint nonnegative matrix factorization, *in* '2015 Third International Conference on Advanced Cloud and Big Data', pp. 28–35.

He, X., Cai, D. & Niyogi, P. (2005), Laplacian score for feature selection, *in* 'Proceedings of the 18th International Conference on Neural Information Processing Systems', NIPS'05, MIT Press, Cambridge, MA, USA, pp. 507–514.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., r. Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. & Kingsbury, B. (2012), 'Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups', *IEEE Signal Processing Magazine* **29**(6), 82–97.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006), 'A fast learning algorithm for deep belief nets', *Neural Comput.* **18**(7), 1527–1554.

Hinton, G. E. & Salakhutdinov, R. R. (2006), 'Reducing the dimensionality of data with neural networks', *Science* **313**(5786), 504–507.

Hira, Z. M. & Gillies, D. F. (2015), 'A review of feature selection and feature extraction methods applied on microarray data', *Advances in bioinformatics* **2015**.

Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., Bijnens, L., Ghlmann, H. W. H., Shkedy, Z. & Clevert, D.-A. (2010), 'Fabia: factor analysis for bicluster acquisition', *Bioinformatics* **26**(12), 1520–1527.

Hofmann, T. (1999), Probabilistic latent semantic analysis, *in* 'Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence', UAI'99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 289–296.

Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z. & Zhu, C. (2013), Personalized recommendation via cross-domain triadic factorization, *in* 'Proceedings of the 22Nd International Conference on World Wide Web', WWW '13, ACM, New York, NY, USA, pp. 595–606.

Hu, R., Pan, S., Long, G., Zhu, X., Jiang, J. & Zhang, C. (2016), Co-clustering enterprise social networks, *in* '2016 International Joint Conference on Neural Networks (IJCNN)', pp. 107–114.

Hu, Y., Wang, Z., Wu, W., Guo, J. & Zhang, M. (2010), Recommendation for movies and stars using yago and imdb, *in* 'Web Conference (APWEB), 2010 12th International Asia-Pacific', pp. 123–129.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A. & Heck, L. (2013), Learning deep structured semantic models for web search using clickthrough data, *in* 'Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management', CIKM '13, ACM, New York, NY, USA, pp. 2333–2338.

Huang, S., Zhang, J., Lu, S. & Hua, X.-S. (2015), Social friend recommendation based on network correlation and feature co-clustering, *in* 'Proceedings of the 5th ACM on International Conference on Multimedia Retrieval', ICMR '15, ACM, Shanghai, China, pp. 315–322.

Huang, S., Zhang, J., Wang, L. & Hua, X. S. (2016), 'Social friend recommendation based on multiple network correlation', *IEEE Transactions on Multimedia* **18**(2), 287–299.

Huang, Y., Wang, W., Wang, L. & Tan, T. (2013), Multi-task deep neural network for multi-label learning, *in* '2013 IEEE International Conference on Image Processing', pp. 2897–2900.

Ibrahim, R., Yousri, N. A., Ismail, M. A. & El-Makky, N. M. (2014), Multi-level gene/mirna feature selection using deep belief nets and active learning, *in* '2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society', pp. 3957–3960.

Isinkaye, F., Folajimi, Y. & Ojokoh, B. (2015), 'Recommendation systems: Principles, methods and evaluation', *Egyptian Informatics Journal* **16**(3), 261 – 273.

Jain, A. K., Duin, R. P. W. & Mao, J. (2000), 'Statistical pattern recognition: A review', *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 4–37.

Jebabli, M., Cherifi, H., Cherifi, C. & Hamouda, A. (2015), User and group networks on youtube: A comparative analysis, *in* '2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)', pp. 1–8.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. & Darrell, T. (2014*a*), 'Caffe: Convolutional architecture for fast feature embedding', *arXiv preprint arXiv:1408.5093* .

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. & Darrell, T. (2014*b*), 'Caffe: Convolutional architecture for fast feature embedding', *arXiv preprint arXiv:1408.5093* .

Jiang, M., Cui, P., Wang, F., Yang, Q., Zhu, W. & Yang, S. (2012), Social recommendation across multiple relational domains, *in* 'ACM CIKM'12', Maui, HI.

Jin, Y., Khan, L., Wang, L. & Awad, M. (2005), Image annotations by combining multiple evidence & Wordnet, *in* 'ACM MM'05', Vol. 15, Singapore, pp. 706–715.

Joachims, T. (2002), Optimizing search engines using clickthrough data, *in* 'Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '02, ACM, New York, NY, USA, pp. 133–142.

Johnson, R. A. & Wichern, D. W., eds (1988), *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jolliffe, I. T. (1986), *Principal Component Analysis*, Springer, New York.

Kacchi, T. R. & Deorankar, A. V. (2016), Friend recommendation system based on lifestyles of users, *in* '2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)', pp. 682–685.

Khalid, S., Khalil, T. & Nasreen, S. (2014), A survey of feature selection and feature extraction techniques in machine learning, *in* 'Science and Information Conference (SAI), 2014', pp. 372–378.

Klau, G. (2009), 'A new graph-based method for pairwise global network alignment', *BMC Bioinformatics* **10**(S1), S59.

Knerr, S., Personnaz, L. & Dreyfus, G. (1990), 'Single-layer learning revisited: A stepwise procedure for building and training a neural network', *Neurocomputing: Algorithms, Architectures and Application* **68**, 41–50.

Kohavi, R. & John, G. H. (1997), 'Wrappers for feature subset selection', *Artif. Intell.* **97**(1-2), 273–324.

Kosir, P., DeWall, R. & Mitchell, R. A. (1996), A multiple measurement approach for feature alignment, *in* 'IEEE NAECON'96', Vol. 1, pp. 94–101.

Kreibirch, C. & Crowcroft, J. (2006), Efficient sequence alignment of network traffic, *in* 'ACM IMC'06', Rio de Janeiro, Brazil, pp. 307–312.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012*a*), Imagenet classification with deep convolutional neural networks, *in* 'Advances in neural information processing systems', pp. 1097–1105.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012*b*), Imagenet classification with deep convolutional neural networks, *in* F. Pereira, C. J. C. Burges, L. Bottou & K. Q. Weinberger, eds, 'Advances in Neural Information Processing Systems 25', Curran Associates, Inc., pp. 1097–1105.

Lam, A. N., Nguyen, A. T., Nguyen, H. A. & Nguyen, T. N. (2015), Combining deep learning with information retrieval to localize buggy files for bug reports (n), *in* 'Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on', pp. 476–481.

Li, B., Yang, Q. & Xue, X. (2009), Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction, *in* 'Proceedings of the 21st International Jont Conference on Artifical Intelligence', IJCAI'09, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 2052–2057.

Li, H., Nie, Z., Lee, W.-C., Giles, L. & Wen, J.-R. (2008), Scalable community discovery on textual data with relations, *in* 'Proceedings of the 17th ACM Conference on Information and Knowledge Management', CIKM '08, ACM, New York, NY, USA, pp. 1203–1212.

Li, J., Chen, Z., Wei, L., Xu, W. & Kou, G. (2007), 'Feature selection via least squares support feature machine', *International Journal of Information Technology amp; Decision Making* **06**(04), 671–686.

Li, J., Hu, X., Tang, J. & Liu, H. (2015), Unsupervised streaming feature selection in social media, *in* 'Proceedings of the 24th ACM International on Conference on Information and Knowledge Management', CIKM '15, ACM, New York, NY, USA, pp. 1041–1050.

Li, K., Gao, J., Guo, S., Du, N., Li, X. & Zhang, A. (2014), Lrbm: A restricted boltzmann machine based approach for representation learning on linked data, *in* '2014 IEEE International Conference on Data Mining', pp. 300–309.

Li, L.-H., Lee, F.-M., Chen, Y.-C. & Cheng, C.-Y. (2009), A multi-stage collaborative filtering approach for mobile recommendation, *in* 'Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication', ICUIMC '09, ACM, Suwon, Korea, pp. 88–97.

Li, N. & Chen, G. (2009), Multi-layered friendship modeling for location-based mobile social networks, *in* 'IEEE MobiQuitous'09', Toronto, Canada, pp. 1–10.

Li, Q., Zheng, Y., Xie, X., Chen, Y., Liu, W. & Ma, W.-Y. (2008), Mining user similarity based on location history, *in* 'Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems', GIS '08, ACM, New York, NY, USA, pp. 34:1–34:10.

Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. (2015), 'Gated graph sequence neural networks', *arXiv preprint arXiv:1511.05493* .

Liang, T.-P., Ho, Y.-T., Li, Y.-W. & Turban, E. (2011), 'What drives social commerce: The role of social support and relationship quality', *International Journal of Electronic Commerce* **16**(2), 69–90.

Liu, D., Ye, G., Chen, C.-T., Yan, S. & Chang, S.-F. (2012), Hybrid social media network, *in* 'ACM MM'12', Nara,Japan, pp. 659–668.

Liu, F., Liu, B., Sun, C., Liu, M. & Wang, X. (2013), Deep learning approaches for link prediction in social network services, *in* 'International Conference on Neural Information Processing', Springer, pp. 425–432.

Liu, F., Liu, B., Sun, C., Liu, M. & Wang, X. (2015), 'Deep belief network-based approaches for link prediction in signed social networks', *Entropy* **17**(4), 2140–2169.

Liu, H. & Motoda, H. (2007), *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*, Chapman & Hall/CRC.

Liu, H. & Mototda, H. (2008), *Computational Methods for Feature Selection*, Chapman & Hall.

Liu, J., Rui, W., Zhang, L. & Jia, Y. (2016), Social relation extraction with improved distant supervised and word embedding features, *in* '2016 IEEE International Conference on Big Data Analysis (ICBDA)', pp. 1–5.

Liu, P., Cao, J., Liang, X. & Li, W. (2015), A two-stage cross-domain recommendation for cold start problem in cyber-physical systems, *in* 'Machine Learning and Cybernetics (ICMLC), 2015 International Conference on', Vol. 2, pp. 876–882.

Liu, X., Wang, L., Zhang, J., Yin, J. & Liu, H. (2014*a*), 'Global and local structure preservation for feature selection', *IEEE Transactions on Neural Networks and Learning Systems* **25**(6), 1083–1095.

Liu, X., Wang, L., Zhang, J., Yin, J. & Liu, H. (2014*b*), 'Global and local structure preservation for feature selection', *IEEE Transactions on Neural networks and learning systems* pp. 1083–1095.

Low, Y., Agarwal, D. & Smola, A. J. (2011), Multiple domain user personalization, *in* 'Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '11, New York, NY, USA, pp. 123–131.

Luo, D., Ding, Y., Cao, Z. & Wu, X. (2014), A multi-stage approach for efficiently learning humanoid robot stand-up behavior, *in* '2014 IEEE International Conference on Mechatronics and Automation', pp. 884–889.

Ma, H., Yang, H., Lyu, M. R. & King, I. (2008*a*), Sorec: social recommendation using probabilistic matrix factorization, *in* 'ACM CIKM'08', Napa Valley, CA, pp. 931–940.

Ma, H., Yang, H., Lyu, M. R. & King, I. (2008*b*), Sorec: Social recommendation using probabilistic matrix factorization, *in* 'Proceedings of the 17th ACM Conference on Information and Knowledge Management', CIKM '08, ACM, New York, NY, USA, pp. 931–940.

Madeira, S. C. & Oliveira, A. L. (2004), 'Biclustering algorithms for biological data analysis: A survey', *IEEE/ACM Trans. Comput. Biol. Bioinformatics* **1**(1), 24–45.

Mallik, R. & Sagias, N. (2011), 'Distribution of inner product of complex gaussian random vectors and its applications', *Communications, IEEE Transactions on* **59**(12), 3353–3362.

McCaffery, J. P. & Maida, A. S. (2013), Toward a causal topic model for video scene analysis, *in* 'Neural Networks (IJCNN), The 2013 International Joint Conference on', pp. 1–8.

McCulloch, W. & Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics* **5**, 115–133.

Michiel, H. (2001), *Mellin Transform*, Encyclopedia of Mathematics, Springer.

Min, W., Bao, B.-K., Xu, C. & Hossain, M. (2015), 'Cross-platform multi-modal topic modeling for personalized inter-platform recommendation', *Multimedia, IEEE Transactions on* **17**(10), 1787–1801.

Mosquera-Lopez, C., Agaian, S. & Velez-Hoyos, A. (2014), The development of a multi-stage learning scheme using new tissue descriptors for automatic grading of prostatic carcinoma, *in* '2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 3586–3590.

Nallapati, R. & Cohen, W. W. (2008), Link-plsa-lda: A new unsupervised model for topics and influence of blogs., *in* 'ICWSM'.

Nie, F., Huang, H., Cai, X. & Ding, C. (2010), Efficient and robust feature selection via joint ℓ2,1-norms minimization, *in* 'Proceedings of the 23rd International Conference on Neural Information Processing Systems', NIPS'10, Curran Associates Inc., USA, pp. 1813–1821.

Nie, F., Xiang, S., Jia, Y., Zhang, C. & Yan, S. (2008), Trace ratio criterion for feature selection., *in* 'AAAI', Vol. 2, pp. 671–676.

Nithiya, P., Vidhya, V. & Ganesan, L. (2010), Development of semantic based information retrieval using word-net approach, *in* 'ICCCNT'10', Karur, India, pp. 320–323.

Oord, A. v. d., Dieleman, S. & Schrauwen, B. (2013), Deep content-based music recommendation, *in* 'Proceedings of the 26th International Conference on Neural Information Processing Systems', NIPS'13, Curran Associates Inc., USA, pp. 2643–2651.

Pan, J. G. & Lin, Y. F. (2011), Reconsidering the marketing strategies over social network - on the perspective of network topology, *in* 'Service Sciences (IJCSS), 2011 International Joint Conference on', pp. 155–157.

Pan, W., Aharony, N. & Pentland, A. S. (2011), Composite social network for predicting mobile apps installation, *in* 'AAAI'11', San Francisco, CA.

Parpalexakis, E. E., Sidiropoulous, N. D. & Bro, R. (2009), From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors, *in* 'WI-IAT'09', Milan, Italy, pp. 311–314.

Pazzani, M. J. (1999), 'A framework for collaborative, content-based and demographic filtering', *Artif. Intell. Rev.* **13**(5-6), 393–408.

Peng, H., Long, F. & Ding, C. (2005), 'Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy', *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(8), 1226–1238.

Peng, X., Zhang, L., Tian, F. & Zhang, D. (2015), 'A novel sensor feature extraction based on kernel entropy component analysis for discrimination of indoor air contaminants', *Sensors and Actuators A: Physical* **234**, 143 – 149.

Pierre, C. (2007), 'Semi-supervised feature selection via spectral analysis'.

Pipanmekaporn, L. & kamolsantiroj, S. (2016), Mining semantic location history for collaborative poi recommendation in online social networks, *in* '2016 2nd International Conference on Open and Big Data (OBD)', pp. 31–38.

Pruett, J. M. (1972), The distribution of products of some independent, non-standardized random variables, *in* 'Ph. D. Dissertation', University of Arkansas.

Pudil, P., Novovičová, J. & Kittler, J. (1994), 'Floating search methods in feature selection', *Pattern Recogn. Lett.* **15**(11), 1119–1125.

Qinglin, W., Huifeng, X., Bo, L. & Minghu, W. (2010), A two-stage personalized recommendation in cts using graph-based clustering, *in* 'E-Business and E-Government (ICEE), 2010 International Conference on', pp. 3629–3632.

Read, J. & Hollmén, J. (2014), A deep interpretation of classifier chains, *in* 'International Symposium on Intelligent Data Analysis', Springer, pp. 251–262.

Read, J. & Perez-Cruz, F. (2014), 'Deep learning for multi-label classification', *CoRR abs/1502.05988* .

Rege, M., Dong, M. & Fotouhi, F. (2006), Co-clustering image features and semantic concepts, *in* 'Image Processing, 2006 IEEE International Conference on', IEEE, pp. 137–140.

Rendle, S. & Thieme, L. S. (2008), Online-updating regularized kernel matrix factorization models for large-scale recommender systems, *in* 'ACM RecSys'08', Lausanne, Swizerland, pp. 251–258.

Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B. (2010), *Recommender Systems Handbook*, 1st edn, Springer-Verlag New York, Inc., New York, NY, USA.

Richards, W. A. & Acharya, M. R. (2010), An efficient plynomial approximation to the normal distribution function and its inverse function, *in* 'Journal of Mathematics Research'.

Roweis, S. T. & Saul, L. K. (2000), 'Nonlinear dimensionality reduction by locally linear embedding', *Science* **290**(5500), 2323–2326.

Roy, S. D., Mei, T., Zeng, W. & Li, S. (2012), Socialtransfer: Cross-domain transfer learning from social streams for media applications, *in* 'ACM MM'12', Nara, Japan, pp. 649–658.

Rummel, R. J. (1991), *The Conflict Helix:Principles and Practices of Interpersonal, Social, and International Conflict and Cooperation*, New Brunswick, N.J.

Sarikaya, R., Hinton, G. E. & Deoras, A. (2014), 'Application of deep belief networks for natural language understanding', *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **22**(4), 778–784.

Schein, A. I., Popescul, A., Ungar, L. H. & Pennock, D. M. (2002), Methods and metrics for cold-start recommendations, *in* 'Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', SIGIR '02, ACM, New York, NY, USA, pp. 253–260.

Shah, F. P. & Patel, V. (2016), A review on feature selection and feature extraction for text classification, *in* '2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)', pp. 2264–2268.

Shalforoushan, S. H. & Jalali, M. (2015), Link prediction in social networks using bayesian networks, *in* 'Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on', pp. 246–250.

Shi, J. & Malik, J. (2000), 'Normalized cuts and image segmentation', *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905.

Siting, Z., Wenxing, H., Ning, Z. & Fan, Y. (2012), Job recommender systems: A survey, *in* 'Computer Science Education (ICCSE), 2012 7th International Conference on', pp. 920–924.

Sivakumar, N., Balaganesh, N. & Muneeswaran, K. (2015), Feature selection for recommendation of movies, *in* 'Communication Technologies (GCCT), 2015 Global Conference on', pp. 250–255.

Song, Y., Zhang, J. & Cao, L. (2013), 'A joint interest-social latent factor model for social recommendation', *Technical report* .

Springer, M. D. (1979), *The Algebra of Random Variables*, John Wilye & Sons, Inc.

Springer, M. D. & Thompson, W. (1977), 'The distribution of products of beta, gamma and gaussian random variables', *SIAM Journal on Applied Mathematics* **18**(4), 721–737.

Stoppiglia, H., Dreyfus, G., Dubois, R. & Oussar, Y. (2003), 'Ranking a random feature for variable and feature selection', *J. Mach. Learn. Res.* **3**, 1399–1414.

Su, X. & Khoshgoftaar, T. M. (2009), 'A survey of collaborative filtering techniques', *Adv. in Artif. Intell.* **2009**, 4:2–4:2.

Sun, G., Yang, G., Liao, D., Xu, Z. & Luo, K. (2015), Efficient mapping of hybrid virtual networks across multiple domains, *in* '2015 IEEE Global Communications Conference (GLOBECOM)', pp. 1–6.

Sun, R. & Luo, Z. Q. (2016), 'Guaranteed matrix completion via non-convex factorization', *IEEE Transactions on Information Theory* **62**(11), 6535–6579.

Swets, D. & Weng, J. (1996), 'Using Discriminant Eigenfeatures for Image Retrieval', *IEEE Trans. Pattern Anal. Machine Intell.* **18**(8), 831–836.

Tang, C., Zhang, L., Ramanathan, M. & Zhang, A. (2001), Interrelated two-way clustering: An unsupervised approach for gene expression data analysis, *in* 'Proceedings of the 2Nd IEEE International Symposium on

Bioinformatics and Bioengineering', BIBE '01, IEEE Computer Society, Washington, DC, USA, pp. 41–.

Tang, J. & Liu, H. (2014), 'An unsupervised feature selection framework for social media data', *IEEE Transactions on Knowledge and Data Engineering* **26**(12), 2914–2927.

Tao, W. & Rui, G. (2006), Empirical study on the influence of commercial friendship on relationship quality and customer loyalty, *in* '2006 International Conference on Management Science and Engineering', pp. 729–735.

Teng, T. H., Tan, A. H., Starzyk, J. A., Tan, Y. S. & Teow, L. N. (2014), Integrating self-organizing neural network and motivated learning for coordinated multi-agent reinforcement learning in multi-stage stochastic game, *in* '2014 International Joint Conference on Neural Networks (IJCNN)', pp. 4229–4236.

Thi, D. B. & Hoang, T. A. N. (2013), Features extraction for link prediction in social networks, *in* 'Computational Science and Its Applications (ICCSA), 2013 13th International Conference on', pp. 192–195.

Tian, J., Huang, Y., Guo, Z., Qi, X., Chen, Z. & Huang, T. (2015), 'A multi-modal topic model for image annotation using text analysis', *IEEE Signal Processing Letters* **22**(7), 886–890.

Tibshirani, R. (1994), Regression shrinkage and selection via the lasso, *in* 'Journal of the Royal Statistical Society, Series B', pp. 267–288.

Tiroshi, A. & Kuflik, T. (2012), Domain ranking for cross domain collaborative filtering, *in* 'Proceedings of the 20th International Conference on User Modeling, Adaptation, and Personalization', UMAP'12, Springer-Verlag, Berlin, Heidelberg, pp. 328–333.

van den Oord, A., Dieleman, S. & Schrauwen, B. (2013), Deep content-based music recommendation, *in* 'Advances in Neural Information Processing Systems 26', Curran Associates, Inc., pp. 2643–2651.

Verma, A. K. & Pal, M. (2015), Evolving social networks via friend recommendations, *in* '2015 11th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)', pp. 379–383.

von Winckel, G. (2014), 'A deep learning approach to link prediction in dynamic networks'.

Wagner, S. & Wagner, D. (2007), 'Comparing clusterings- an overview'.

Wan, S., Lan, Y., Guo, J., Fan, C. & Cheng, X. (2013), Informational friendship recommendation in social media, *in* 'ACM SIGIR'13', Dublin, Ireland, pp. 1045–1048.

Wang, F., Lin, S. & Yu, P. S. (2016), Collaborative co-clustering across multiple social media, *in* '2016 17th IEEE International Conference on Mobile Data Management (MDM)', Vol. 1, pp. 142–151.

Wang, H., Wang, N. & Yeung, D.-Y. (2015), Collaborative deep learning for recommender systems, *in* 'Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '15, ACM, New York, NY, USA, pp. 1235–1244.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C. & Xu, W. (2016), 'Cnn-rnn: A unified framework for multi-label image classification', *CVPR'16* .

Wang, S., Tang, J. & Liu, H. (2015), 'Embedded unsupervised feature selection'.

Wang, X., Crammer, K. & McCallum, A. (2007), 'Undirected and interpretable continuous topic models of documents'.

Wang, X. & Sukthankar, G. (2013), Multi-label relational neighbor classification using social context features, *in* 'Proceedings of The 19th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)', Chicago, USA, pp. 464–472.

Wang, Z., Liao, J., Cao, Q., Qi, H. & Wang, Z. (2015), 'Friendbook: A semantic-based friend recommendation system for social networks', *IEEE Transactions on Mobile Computing* **14**(3), 538–551.

Weber, M. (1991), *The Nature of Social Action*, Cambridge University Press.

Wei, J., Zeng, Q.-f., Wang, X., Wang, J.-b. & Wen, G.-h. (2014), 'Integrating local and global topological structures for semi-supervised dimensionality reduction', *Soft Computing* **18**(6), 1189–1198.

Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y. & Yan, S. (2014), 'Cnn: Single-label to multi-label', *arXiv preprint arXiv:1406.5726* .

Welling, M. & Teh, Y. W. (2008), Hybrid variational/gibbs collapsed inference in topic models, *in* 'In Proceedings of the Conference on Uncertainty in Artifical Intelligence (UAI)', pp. 587–594.

Weston, J., Bengio, S. & Usunier, N. (2011), Wsabie: Scaling up to large vocabulary image annotation, *in* 'Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three', IJCAI'11, AAAI Press, pp. 2764–2770.

won You, G., won Hwang, S., Nie, Z. & Wen, J.-R. (2011), Socialsearch: Enhancing entity search with social network matching, *in* 'EDBT'11', Upplsala, Sweden, pp. 515–519.

Wu, H. C., Luk, R. W. P., Wong, L. K. F. & Kwok, K. L. (2008), 'Interpreting TF-IDF term weights as making relevance decisions', *ACM Trans. Information Systems* **26**, 13:1–13:37.

Wu, M. & Chen, L. (2015), Image recognition based on deep learning, *in* 'Chinese Automation Congress (CAC), 2015', pp. 542–546.

Wu, Z., Jiang, S. & Huang, Q. (2009), Friend recommendation according to appearances on photos, *in* 'Proceedings of the 17th ACM International Conference on Multimedia', MM '09, ACM, New York, NY, USA, pp. 987–988.

x. Lv, H., Yu, G., y. Tian, X. & Wu, G. (2014), Deep learning-based target customer position extraction on social network, *in* '2014 International Conference on Management Science Engineering 21th Annual Conference Proceedings', pp. 590–595.

Xiaoyi Li, Nan Du, H. L. K. L. J. G. A. Z. (2014), 'A deep learning approach to link prediction in dynamic networks.', *SIAM International Conference on Data Mining* pp. 289–297.

Xu, Z., King, I., Lyu, M. R.-T. & Jin, R. (2010), 'Discriminative semi-supervised feature selection via manifold regularization', *Trans. Neur. Netw.* **21**(7), 1033–1047.

Yan, M., Sang, J., Mei, T. & Xu, C. (2013), Friend transfer: Cold-start friend recommendation with cross-platform transfer learning of social knowledge, *in* 'Multimedia and Expo (ICME), 2013 IEEE International Conference on', pp. 1–6.

Yang, J., Frangi, A. F., Yang, J.-Y., Zhang, D. & Jin, Z. (2005), 'Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(2), 230–244.

Yang, J., Wang, H., Wang, W. & Yu, P. (2005), 'An improved biclustering method for analyzing gene expression profiles', *International Journal on Artificial Intelligence Tools* **14**(5), 771–789.

Yang, Y., Ma, Z., Hauptmann, A. G. & Sebe, N. (2013), 'Feature selection for multimedia analysis by sharing information among multiple tasks', *IEEE Trans. on Multimedia* **15**, 661–669.

Yang, Y., Shen, H. T., Ma, Z., Huang, Z. & Zhou, X. (2011), L2,1-norm regularized discriminative feature selection for unsupervised learning, *in* 'Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two', IJCAI'11, AAAI Press, pp. 1589–1594.

Yang, Y., Xu, Z., Liu, G., Ma, Z. & Sebe, N. (2013), Global structual feature selection with sparsity for multimedia data understanding, *in* 'ACM MM'13', Barcelona, Spain, pp. 537–540.

Yao, T., Ngo, C.-W. & Mei, T. (2011*a*), Context-based friend suggestion in online photo-sharing community, *in* 'Proceedings of the 19th ACM International Conference on Multimedia', MM '11, ACM, New York, NY, USA, pp. 945–948.

Yao, T., Ngo, C.-W. & Mei, T. (2011*b*), Context-based friend suggestion in online photo-sharing community, *in* 'Proceedings of the 19th ACM International Conference on Multimedia', MM '11, ACM, New York, NY, USA, pp. 945–948.

Yao, T., Ngo, C.-W. & Mei, T. (2011*c*), Context-based friend suggestion in online photo-sharing community, *in* 'Proceedings of the 19th ACM International Conference on Multimedia', MM '11, ACM, New York, NY, USA, pp. 945–948.

Yin, H., Zhou, X., Cui, B., Wang, H., Zheng, K. & Nguyen, Q. V. H. (2016), 'Adapting to user interest drift for poi recommendation', *IEEE Transactions on Knowledge and Data Engineering* **28**(10).

Zeng, W. & Chen, L. (2013), Heterogeneous data fusion via matrix factorization for augmenting item, group and friend recommendations, *in* 'Proceedings of the 28th Annual ACM Symposium on Applied Computing', SAC '13, ACM, Coimbra, Portugal, pp. 237–244.

Zeng, X., Ouyang, W. & Wang, X. (2013), Multi-stage contextual deep learning for pedestrian detection, *in* '2013 IEEE International Conference on Computer Vision', pp. 121–128.

Zha, H., He, X., Ding, C., Simon, H. & Gu, M. (2001), Bipartite graph partitioning and data clustering, *in* 'Proceedings of the Tenth International Conference on Information and Knowledge Management', CIKM '01, ACM, New York, NY, USA, pp. 25–32.

Zhang, L., Fang, H., Ng, W. K. & Zhang, J. (2011), Intrank: Interaction ranking-based trustworthy friend recommendation, *in* '2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications', pp. 266–273.

Zhao, J., Lu, K. & He, X. (2008), 'Locality sensitive semi-supervised feature selection', *Neurocomput.* **71**(10-12), 1842–1849.

Zhao, L. & Zaki, M. J. (2005), tricluster: An effective algorithm for mining coherent clusters in 3d microarray data, *in* 'In Proc. of the 2005 ACM SIGMOD international conference on Management of data', ACM Press, pp. 694–705.

Zhao, T., McAuley, J. & King, I. (2014), Leveraging social connections to improve personalized ranking for collaborative filtering, *in* 'Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management', CIKM '14, ACM, New York, NY, USA, pp. 261–270.

Zhao, X. & Wang, J. (2015), A theoretical analysis of two-stage recommendation for cold-start collaborative filtering, *in* 'Proceedings of the 2015 International Conference on The Theory of Information Retrieval', ICTIR '15, ACM, Northampton, Massachusetts, USA, pp. 71–80.

Zhao, Y., Wang, G., Yu, P. S., Liu, S. & Zhang, S. (2013), Inferring social roles and statuses in social networks, *in* 'ACM KDD'13', pp. 695–703.

Zhao, Z. & Liu, H. (2007), Spectral feature selection for supervised and unsupervised learning, *in* 'Proceedings of the 24th International Conference on Machine Learning', ICML '07, ACM, New York, NY, USA, pp. 1151–1157.

Zhao, Z., Wang, L., Liu, H. & Ye, J. (2013), 'On similarity preserving feature selection', *IEEE Trans. on Knowl. and Data Eng.* **25**(3), 619–632.

Zheng, N., Song, S. & Bao, H. (2015), 'A temporal-topic model for friend recommendations in chinese microblogging systems', *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* **45**(9), 1245–1253.

Zhong, E., Fan, W. & Yang, Q. (2014), 'User behavior learning and transfer in composite social networks', *ACM Transactions on Knowledge Discovery from Data'14* **8**.

Zhou, Q., Xu, G. & Zong, Y. (2012), Web co-clustering of usage network using tensor decomposition, *in* 'IEEE Transations on Signal Processing', pp. 493–506.

Zhou, S., Liu, X., Zhu, C., Liu, Q. & Yin, J. (2014), Spectral clustering-based local and global structure preservation for feature selection, *in* '2014 International Joint Conference on Neural Networks (IJCNN)', pp. 550–557.

Zhuang, Y.-T., Yang, Y. & Wu, F. (2008), 'Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval', *IEEE Trans. on Multimedia* **10**, 221–229.