Modeling the Relationship between Links and Communities for Overlapping Community Detection

ZHANG, Hongyi

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Doctor of Philosophy

in

Computer Science and Engineering

The Chinese University of Hong Kong

September 2017

Thesis Assessment Committee

Professor LEE Ho Man Jimmy (Chair)

Professor KING Kuo Chin Irwin (Thesis Supervisor)

Professor LYU Rung Tsong Michael (Thesis Co-supervisor)

Professor SUN Hanqiu (Committee Member)

Professor Ngo Chong-wah (External Examiner)

Abstract of thesis entitled:

Modeling the Relationship between Links and Communities for Overlapping Community Detection

Submitted by ZHANG, Hongyi

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in September 2017

Community detection is an important line of research in understanding complex networks. In many real-world networks, communities naturally overlap since a node usually has multiple community memberships, which makes overlapping community detection a trend in recent years. One popular technique to cope with overlapping community detection is matrix factorization (MF). Although all existing MF-based approaches use links as input to identify overlapping communities as output, the relationship between links and communities is still under-investigated.

To view links as consequences of communities (community-to-link), our first work proposes a preference-based non-negative matrix factorization (PNMF) model to incorporate implicit link preference information. Unlike conventional matrix factorization approaches which simply approximate the original adja-

cency matrix in value, our PNMF model maximizes the likelihood of the preference order for each node by following the intuition that a node prefers its neighbors than non-neighbors. Our PNMF model overcomes the indiscriminate penalty problem in which non-linked pairs inside one community are equally penalized in objective functions as those across two communities.

Based on our first work, our second work proposes a locality-based non-negative matrix factorization (LNMF) model to refine the PNMF model by making use of "local non-neighbors" (e.g., my friend's friend but not my direct friend). We define a subgraph called "k-degree local network" to set a boundary between local non-neighbors and other non-neighbors. By discriminatively treating these two classes of non-neighbors, our LNMF model can discover more fine-grained communities.

While the LNMF model can be regarded as a generalization of the PNMF model, our third work proposes a mutual density based non-negative matrix factorization (MD-NMF) model as an alternative to the PNMF model. The MD-NMF model is based on the observation that mutual friends between two nodes can better reflect their closeness regarding community memberships compared with link existence. By introducing the concept of mutual density and using it to replace links as a new indicator to infer the similarity of community membership between two nodes, our new objective maximizes the likelihood that node pairs with larger mutual density are more similar in community memberships.

By further investigating how nodes' community memberships can be represented by their linked neighbors (link-to-community), our fourth work proposes a homophily-based non-negative matrix factorization (HNMF) method to boost community representation learning by the mutual enhancement of both-sided relationships between links and communities. In particular, from the community-to-link perspective, we adopt the PNMF model in our first work. From the link-to-community perspective, we propose a community representation learning with network embedding techniques by assuming that linked nodes have similar community representations.

For all the models we propose, we employ a learning algorithm which learns a node-community membership matrix via stochastic gradient descent with fast sampling strategies. We evaluate our models on several real-world networks including large ones with ground-truth communities. Experimental results show that by exploring and modeling the two-sided relationship between links and communities, our models outperform state-of-the-art approaches on multiple measurements and are capable of finding overlapping communities with better quality.

論文題目：可重疊社區發現中鏈接與社區的相互關係的探究和建模

作者　　：張弘毅

學校　　：香港中文大學

學系　　：計算機科學與工程學系

修讀學位：哲學博士

摘要　　：

社區發現是理解複雜網絡的一項重要研究課題。在很多實際的網絡中，由於一個節點經常存在多個身份，社區往往會相互重疊，這使得可重疊社區發現成為近些年這一課題的研究趨勢。矩陣分解是解決可重疊社區發現的一項常用的工具。雖然現存的基於矩陣分解的方法都使用鏈接作為算法輸入去識別作為算法輸出的可重疊社區，但是鏈接和社區之間的相互關係還是缺乏研究。

當鏈接被看作是社區存在后的必然結果（從社區到鏈接），我們的第一個工作提出了一個基於偏好的非負矩陣分解（PNMF）模型來利用隱藏的鏈接偏好信息。和傳統的矩陣分解方法使用目標函數在數值上去近似原本的鄰接矩陣不同，我們的PNMF模型通過遵循一個節點對它的鄰接節點有更高的偏好來最大化每個節點的偏好序列。我們的PNMF模型克服了以前工作中不區分社區內不相鄰節點對和社區之間不相鄰節點對的問題。

基於我們的第一個工作，我們的第二個工作通過利用局部非鄰接節點的概念（即朋友的朋友但非直接朋友）提出了一個基於局部性的非負矩陣分解（LNMF）模型來改善PNMF模型。我們定義了一個叫做k度局部網絡的子圖來劃分局部非鄰

接節點和其他非鄰接節點。通過區別對待這兩種非鄰接節點，我們的LNMF模型可以發現更加細粒度的社區。

如果說LNMF模型可以被看做是PNMF模型的一種一般化的話，我們的第三個工作提出了一種基於共同好友密度的非負矩陣分解（MD-NMF）模型來替換PNMF模型。MD-NMF模型的提出是因為我們發現兩個節點的共同好友比兩個節點之間是否存在鏈接更好的反應了兩個節點社區從屬的相似性。通過引入共同好友密度這個概念并將其取代鏈接作為推斷兩個節點社區從屬關係相似性的指示符，我們的新目標函數是去最大化共同好友密度越大的節點對之間社區從屬關係更相似的概率。

通過進一步探索節點的社區從屬關係如何被它們的鄰接節點所表達（從鏈接到社區），我們的第四個工作提出了一種基於趨同性的非負矩陣分解（HNMF）模型。這種模型通過鏈接與社區之間的相互作用來加快對社區表達的學習。具體來說，從社區到鏈接的角度，我們使用了我們的第一個工作，即PNMF模型；從鏈接到社區的角度，我們通過假設相鄰節點具有更相似的社區從屬關係提出了一種通過網絡嵌入技術的社區表達學習算法。

對於我們提出的上述模型，我們都採用隨機梯度下降算法和快速的抽樣方法來學習節點和社區之間的從屬關係矩陣。我們的實驗數據都是實際的網絡，其中包括有真實社區信息的大型網絡。實驗結果顯示，通過對鏈接和社區相互關係的探究和建模，我們所提出的四個模型比現有的模型在多個指標上擁有更好的效果，可以發現更高質量的社區。

Acknowledgement

I would like to thank my parents, without whom none of these would be possible.

I would like to thank my supervisors, Prof. Irwin King and Prof. Michael R. Lyu, for their guidance, encouragements and patience during my postgraduate study in the Chinese University of Hong Kong. I am deeply grateful for all the efforts they have put on my research as well as writing and presentation skills.

I would like to thank my thesis committee members, Prof. Jimmy Lee, Prof. Hanqiu Sun, Prof. Chong-Wah Ngo, and former thesis committee chairman, Prof. Yufei Tao, for their valuable advices.

I would like to thank Wei Xiang and Qian Xu, my mentors during my internship at Baidu.

I would like to thank my life-long friends, Chun Chen, Ge Fang, Tiansheng Yao, for their trust and support.

I would like to thank my talented colleagues, Haiqin Yang, Chao Zhou, Baichuan Li, Shouyuan Chen, Qirun Zhang, Guang Ling, Chen Cheng, Tong Zhao, Shenglin Zhao, Xixian Chen,

Xiaotian Yu, Yuxin Su, Jichuan Zeng, Ken Chan, Jiani Zhang, Wang Chen, Han Shao, Yue Wang and Xingyu Niu. I am grateful to share my life with them.

To my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the emergence and prevalence of online social networks, it becomes much easier for geographically distant people to get acquainted and keep in touch with each other. People, especially teenagers, spend more and more time on online social networks in their daily lives [15]. For example, Facebook, the currently largest online social network, has over 1.86 billion monthly active users with a 17% yearly increase[1]. Thus, the crucialness of online social networks can no longer be ignored, and researchers from various disciplines have looked into it to get a better understanding of social behaviors [36, 102].

In computer science, an online social network is usually abstracted as a graph with a set of nodes and edges. Unlike a random graph where each node pair has the same probability to be linked [25], an online social network has certain structures [48]. A typical structure is that there are groups of nodes closely connected inside the group but rarely making connections

[1] https://zephoria.com/top-15-valuable-facebook-statistics/

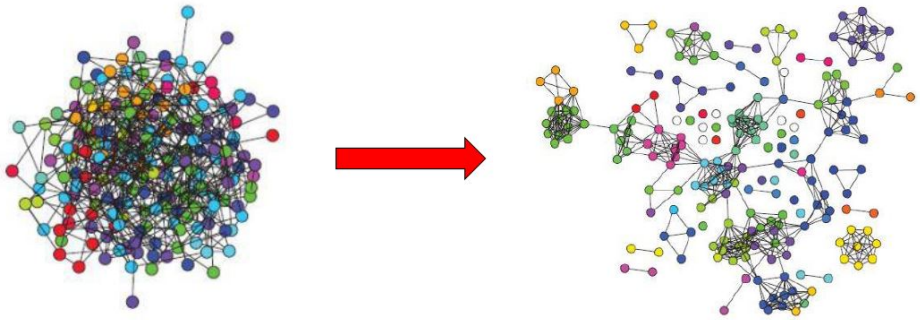


Figure 1.1: The process of community detection.

with nodes outside the group, e.g., people in the same organization or people sharing the same interest. Such structure is called community structure [35]. Community detection is the task of uncovering community structure in complex networks (see Figure 1.1). Apart from finding groups in an online social network, community detection has many other concrete applications. An online retailer can build a better recommender system by clustering customers according to their interests [85]. A Web service can cluster their clients according to their interests or patterns and allocate a dedicated mirror server for each cluster to improve the performance of the service [46].

A community detection problem is naturally viewed as a graph partition/clustering problem and thus can be solved by unsupervised learning algorithms. An implicit assumption behind it is that each node can be assigned to one and only one community. However, it has been shown that overlap is a significant feature of many real-world social networks [86]. For example, in Figure 1.2, the man in the middle has multiple identities

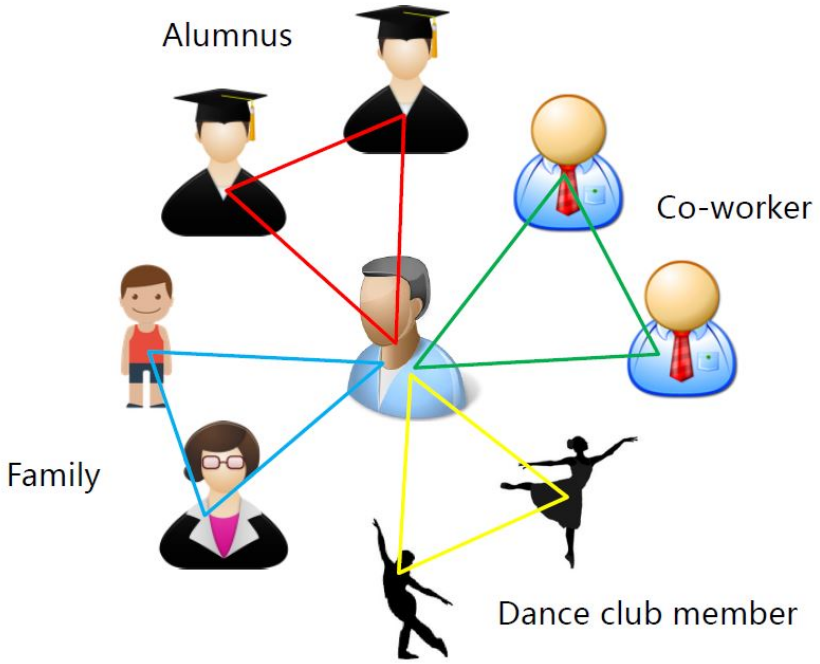


Figure 1.2: Communities are overlapped.

in his social network, e.g., father, an alumnus, a company employer, a dance club member, etc. For each identity, we can find a corresponding community to define it, e.g., his family, all the alumni of his college, all the employees in his company, all the members in his dance club, etc. This particular branch of community detection is called overlapping community detection. Since it is more realistic compared with disjoint community detection, overlapping community detection has drawn more attention recently.

Among various approaches dealing with overlapping community detection, matrix factorization (MF) is one of the standard frameworks. Figure 1.3 shows a typical MF framework where a

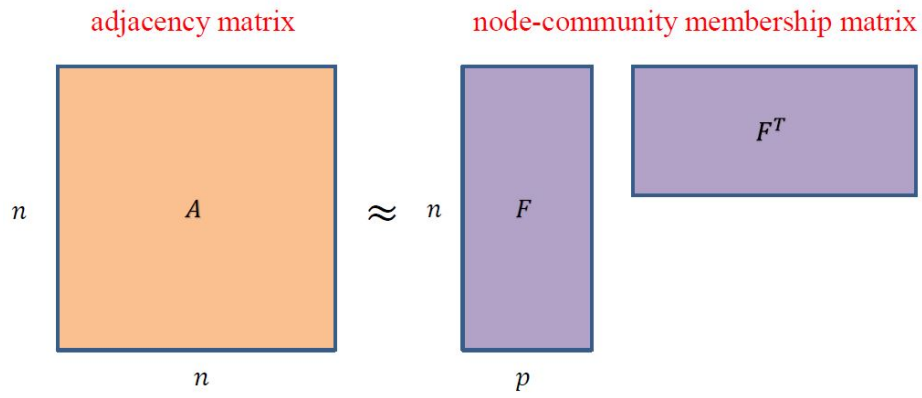


Figure 1.3: An illustration of a typical MF framework for overlapping community detection.

node-community membership matrix is learned to approximate the original adjacency matrix with an optimization function. Each entry in the node-community membership matrix represents the weight of the corresponding node in the corresponding community. By learning the node-community membership matrix, we can determine all the communities a node belongs to according to the weights on its corresponding row.

In the rest of this chapter, we first describe the data type of the input and output of overlapping community detection in Section 1.1. Then we introduce the motivation of this thesis in Section 1.2. We conclude the main contributions of this thesis in Section 1.3 and present the overall roadmap of this thesis in Section 1.4.

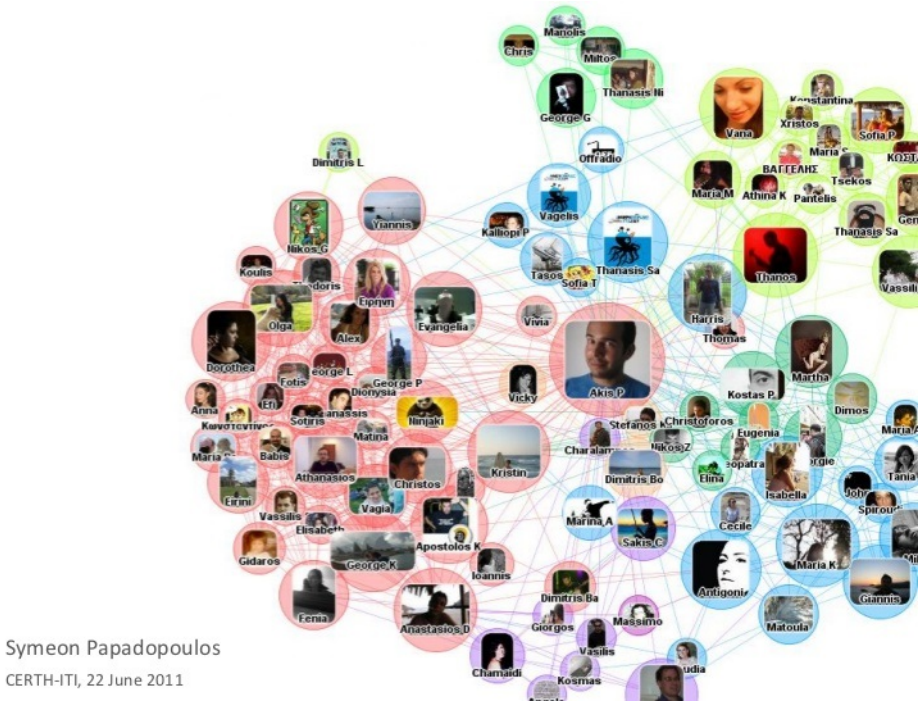


Figure 1.4: An online social network.

1.1 Data Type

A graph (or a network) is one of the basic data structures in computer science. The basic components of a graph are nodes and edges (or links). A node is an entity, and an edge reveals the relation between two nodes. As we mentioned, a real-world graph has community structure. A community can be regarded as a set of nodes sharing a feature. For example, in a social network (see Figure 1.4 [77]), a node represents a person, an edge represents the friend relationship, and a community can be a college, a company, etc.; in a protein-protein interaction network (see Figure 1.5 [76]), a node represents a type of protein, an

edge represents that there is an interaction between two types of proteins, and a community is a set of proteins with a particular functionality.

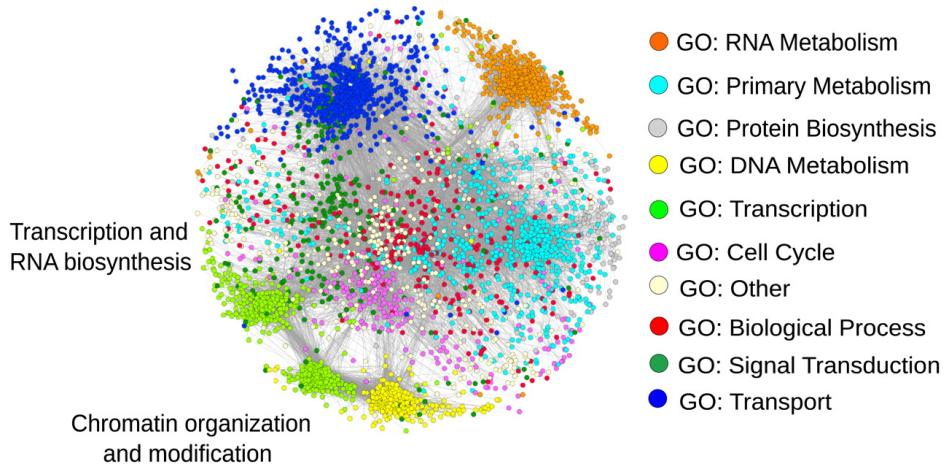


Figure 1.5: A protein-protein interaction network.

This thesis focuses on overlapping community detection of an undirected network. The input is the adjacency matrix, which includes and only includes node and edge information. We can not deny that other information is available in network data as well, such as node attribute, edge weight, etc. But they are beyond the content of this thesis. The output is a set of communities where each of them contains a set of nodes. There are certain metrics to evaluate the goodness of a community structure, and we will discuss it in later chapters.

1.2 Motivation

This thesis focuses on using matrix factorization framework to model the relationship between links and communities for overlapping community detection. Our motivation mainly consists of two parts: (1) previous work is problematic, and (2) we have new insights to come up with new models to overcome these problems.

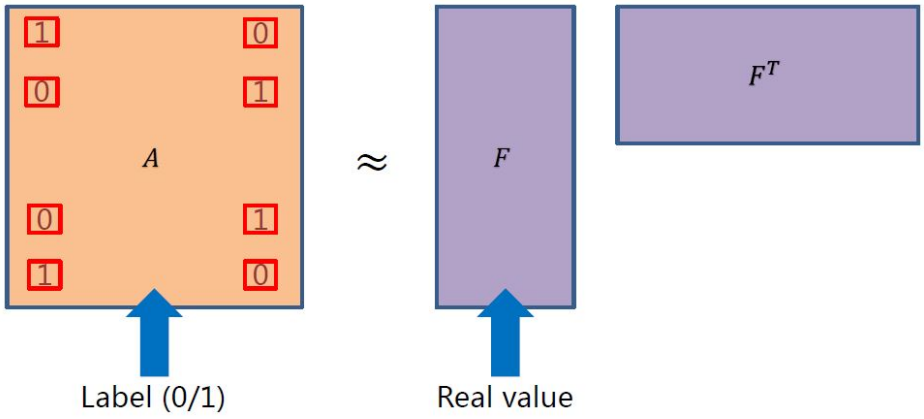


Figure 1.6: Mismatch between labels and real values in previous work.

Given that most of the previous work based on matrix factorization framework has the same or similar factorization form, it is the optimization function that matters the most. As Figure 1.6 shows, a natural thought is to let the product of factorized matrices be as close as the adjacency matrix of the network. This type of optimization function is based on value approximation. However, we notice that the adjacency matrix of an undirected network only consists of binary labels representing whether a link exists or not while the entries in the node-community mem-

bership matrix are real values. A label can always represent the same thing no matter what its actual value is but a value has a physical meaning. The problem of a value approximation based optimization function is the mismatch between a label and a real value. Thus, we need new optimization functions that are not based on value approximation to overcome the mismatch problem.

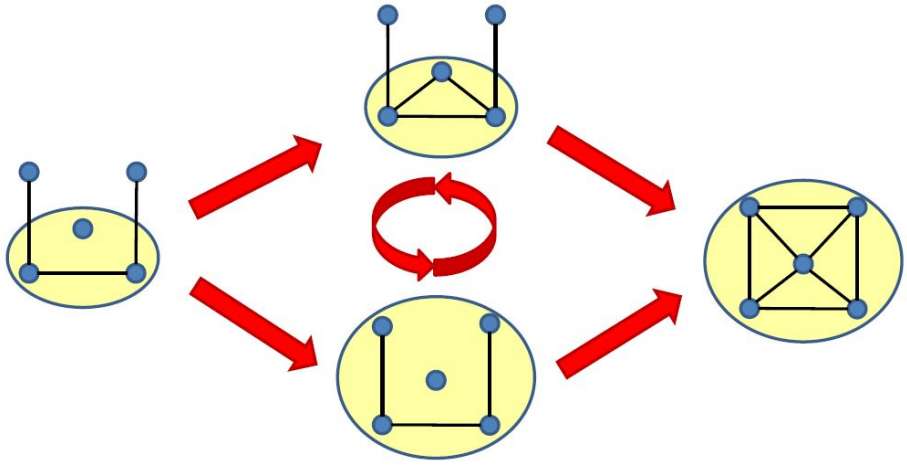


Figure 1.7: How a community evolves.

While previous work only sees the static existence of communities, the truth we cannot deny is that a community has its formation process. Figure 1.7 is a toy illustration on how a community evolves along the way. In this figure, a blue dot represents a node, a line represents a link, and a yellow circle represents a community. As the leftmost part shows, a community is relatively small at the beginning. Nodes inside the community may or may not know each other, and they may also know

nodes outside the community. As times go by, new nodes join the community mostly because of the recommendation of their friends inside the community. On the other hand, strangers inside the community start to make friends with each other due to the intrinsic property of a community that nodes inside it share a common interest. With multiple rounds of mutual effect between links and the community itself, the community becomes bigger and denser until reaching the status shown in the rightmost. Thus, a community can be viewed as a result of mutual enhancement between links and the community itself. From the community-to-link perspective, the intuition is that nodes in the same community have a higher chance to become friends. From the link-to-community perspective, the intuition is that friends usually have similar community structures.

Providing the above two motivations, this thesis proposes multiple novel optimization functions in the matrix factorization framework to capture the relationship between links and communities and eventually solve the problem of overlapping community detection.

1.3 Thesis Contributions

The main contributions of this thesis can be summarized as follows:

1. **A Preference-based Non-negative Matrix Factorization Model for Overlapping Community Detection [110]**

We propose a Preference-based Non-negative Matrix Factorization (PNMF) model to incorporate implicit link preference information, i.e., a node's preference reflected by the links it associates with. This information has a substantial impact on overlapping community detection since a node tends to build links with nodes inside its community than those outside its community but has been ignored in previous work. Different from conventional matrix factorization approach using objective functions to approximate the given adjacency matrix in value, our new objective function maximizes the likelihood of the preference order for each node by following the intuition that a node prefers its neighbors than other nodes. Our objective function overcomes the indiscriminate penalty problem in which non-linked pairs inside one community are equally penalized with those across two communities. We use stochastic gradient descent with bootstrap sampling to learn the node-community membership matrix. Evaluations on several real-world networks show that our PNMF model outperforms state-of-the-art approaches on both modularity and F_1 score and is scalable for large datasets.

2. **A Locality-based Non-negative Matrix Factorization Model for Overlapping Community Detection [111]**

   Based on our PNMF model, we propose a Locality-based Non-negative Matrix Factorization (LNMF) model to re-

fine the preference system by incorporating locality into the learning objective. Our motivation is that "local non-neighbors" (e.g., my friend's friend but not my direct friend) have been ignored in previous work but are helpful when detecting overlapping communities. After defining a subgraph called "k-degree local network" to set a boundary between local non-neighbors and other non-neighbors, we assume that the preference of neighbors is larger than the preference of local non-neighbors and the preference of local non-neighbors is larger than the preference of other non-neighbors. With a refined objective function reflecting our new assumptions, the LNMF model can detect overlapping community in a more precise manner. We employ a fast sampling strategy with stochastic gradient descent as our learning algorithm. By comparing our LNMF model with state-of-the-art baseline methods including the PNMF model on various real-world networks, we show that our LNMF model can achieve higher modularity and F_1 score and detect more fine-grained communities than the PNMF model.

3. **A Mutual Density-based Non-negative Matrix Factorization Model for Overlapping Community Detection**

   Through observations on real-world networks with ground-truth communities, we find that compared with the existence of a link, the number of mutual friends between two

nodes can better reflect their similarity regarding community membership. Based on the concept of mutual friend, we introduce Mutual Density as a new indicator to infer the similarity of community membership between two nodes in the MF framework for overlapping community detection. We propose a Mutual Density-based Non-negative Matrix Factorization (MD-NMF) model by maximizing the likelihood that node pairs with larger mutual density are more similar in community memberships. The new objective function is quite similar to that of the PNMF model but the existence of a link between two nodes has been replaced by the value of mutual density. By conducting experiments on various real-world networks, we show that our MD-NMF model outperforms other state-of-the-art baselines and the PNMF model on both modularity and F_1 score.

4. **A Homophily-based Non-negative Matrix Factorization Model for Overlapping Community Detection [112]**

   Since most existing MF-based approaches only view links as consequences of communities (community-to-link) but fail to explore how nodes' community memberships can be represented by their linked neighbors (link-to-community), we propose a Homophily-based Non-negative Matrix Factorization (HNMF) to model both-sided relationships between links and communities. From the community-to-link perspective, the PNMF model is used since it assumes that

nodes with common communities are more likely to build links with each other. From the link-to-community perspective, we employ the Skip-gram model with network embedding by assuming that linked nodes have similar community representations. We combine both parts into the unified objective function. We conduct experiments on several real-world networks and the evaluations show that our HNMF model achieves higher modularity and F_1 score compared with state-of-the-art baselines including the PNMF model alone.

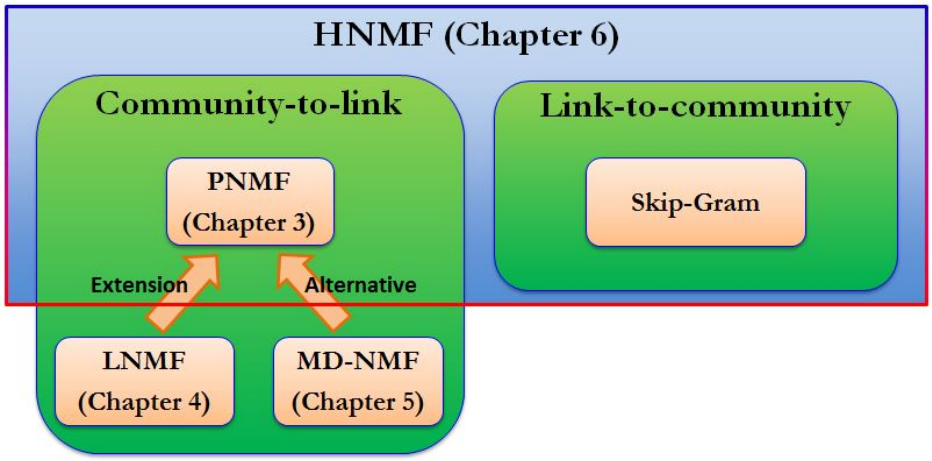


Figure 1.8: Thesis contributions.

Figure 1.8 is a summary of the relationships among the contributions of this thesis mentioned above. We can see that the HNMF model is a combination of both the PNMF model and the Skip-Gram model that will be introduced in Chapter 6. The LNMF model generalizes the preference system of the PNMF model and thus can be regarded as an extension of the PNMF

model. The MD-NMF model uses mutual density instead of link existence to be the indicator and thus can be regarded as an alternative to the PNMF model.

1.4 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 2, we review the background knowledge and previous work closely related to our contributions. Particularly, we first provide an overview of community detection, then conduct a literature review on overlapping community detection, finally dive into the matrix factorization framework for overlapping community detection. In Chapter 3, we propose a Preference-based Non-negative Matrix Factorization (PNMF) model for overlapping community detection. We first provide a brief review of related work. Then we demonstrate model formulation and parameter learning paradigm. In the end, we show our experimental results and conclude this work. In Chapter 4, we propose a Locality-based Non-negative Matrix Factorization (LNMF) model for overlapping community detection. We first define our problem and briefly review some related work. Then we illustrate model formulation and discuss the relationship between this model and the previous PNMF model. In the end, we show our experimental results and conclude this work. In Chapter 5, we propose a Mutual Density-based Non-negative Matrix Factorization (MD-NMF) model for overlapping community detection. We first give

some definitions and show our data observations. Then we demonstrate model formulation and parameter learning paradigm. After that, we show our experimental results and discuss the difference with the previous PNMF model. Finally, we provide a brief review of related work and conclude this work. In Chapter 6, we propose a Homophily-based Non-negative Matrix Factorization (HNMF) model for overlapping community detection. We first define our problem and show our data observations. Then we conduct a brief review of related work. After that, we illustrate model formulation and parameter learning paradigm. Finally, we show our experimental results and conclude this work. Chapter 7 summarizes this thesis and discusses some potential directions that can be explored in future work.

To make the chapters self-contained, we briefly reiterate critical definitions and models that are related in the following chapters.

□ **End of chapter.**

Chapter 2

Background Study

In this chapter, we will go over the background knowledge regarding the focus of this thesis from a general view to a specific view. We will follow the taxonomy in Figure 2.1 by starting with the general topic of community detection, followed by overlapping community detection, and ending with a specific topic of matrix factorization framework for overlapping community detection. In each part, we will first formally define the problem and then conduct a brief literature review.

In each of the following chapters, to make the chapter self-contained, we will reiterate critical literature reviewed in this chapter and talk about other literature which are highly related to that chapter but not mentioned in this chapter.

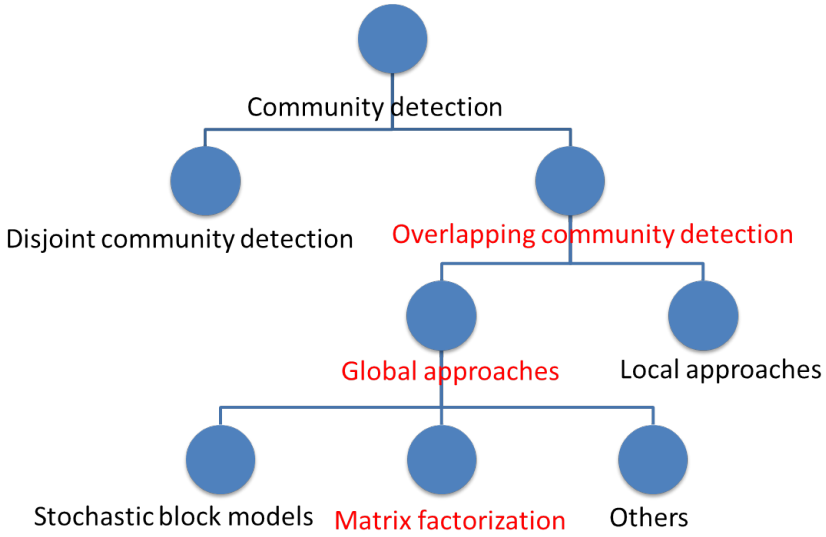


Figure 2.1: The taxonomy of community detection.

2.1 Community Detection

2.1.1 Problem Description

Suppose that we have a graph/network denoted as $G(V, E)$, where V is the node set and E is the link or edge set. We can formally define the concept of *community* as follows.

Definition 2.1 (**Community**). *A community C is a subset of V with a certain characteristic.*

Based on this definition, nodes in a community are more likely to make friends with each other. Therefore, a community usually has stronger internal connections and weaker external connection, which is directly proposed as an alternative definition of community in [35]. Following this, the problem of *community detection* can be defined as follows.

Definition 2.2 (**Community Detection**). *Given a graph* $G(V, E)$, *community detection is a task to find a set of communities* $\mathbb{S} = \{C_i | C_i \neq \emptyset, C_i \neq C_j, 1 \leq i, j \leq p\}$ *that maximizes a particular objective function* f, *i.e.,*

$$\arg\max_{\mathbb{S}} f(G, \mathbb{S}), \tag{2.1}$$

where p *is the number of communities.*

Classic community detection usually has two major assumptions:

- Completeness: every node should belong to one community, i.e., $V = \bigcup_{i=1,..,p} C_i$,
- Disjointness: no node can belong to more than one communities, i.e., $C_i \bigcap C_j = \emptyset$ for all $i \neq j$.

These two assumptions make classic community detection more tractable since now it can be converted to other well-solved problems.

2.1.2 Literature Review

Several surveys regarding classic community detection can be found in [20, 30, 50, 56]. Here we would like to briefly summarize some of the most representative approaches.

Traditional Approaches

A graph partitioning problem is to divide the nodes in a graph into n groups such that the number of edges across two groups

is minimal. The number of such edges is called *cut size*. This objective matches the weak external connection part in the definition of community. As a result, a community detection problem can be viewed as a graph partitioning problem. Since most variants of graph partitioning are NP-hard, many heuristics are proposed to find a good but not necessarily optimal solution, such as the Kernighan-Lin algorithm [43], the spectral bisection method [8], etc. Also, based on the well known max-flow min-cut theorem [29], efficient algorithms dealing with the maximum flow problem can be used to solve community detection at the same time [27, 28]. However, in a graph partitioning problem, the number of groups and the size of each group need to be determined beforehand to avoid trivial solutions. Also, partitions into more than two groups are usually achieved by performing a bisection of the graph multiple times. These are the main reason that algorithms for graph partitioning are bad for community detection.

Though the number and size of communities are usually unknown in advance, a network may display a hierarchical structure, i.e., small groups of nodes included in a large group of nodes. Hierarchical clustering is the technique to reveal such structure [32]. Hierarchical clustering algorithms require a similarity metric to be defined in the beginning and only use the similarity matrix S^{N*N} (S_{ij} - the similarity between node i and node j, N - the number of nodes) to identify groups with high similarity. They can be classified into two categories according to the ge-

neral strategy, which are agglomerative algorithms and divisive algorithms. Agglomerative algorithms iteratively merge small groups into large groups in a bottom-up fashion. Divisive algorithms, on the opposite side, iteratively divide large groups into small groups in a top-down fashion. The main weakness of hierarchical clustering is that the partitions are highly dependent on the similarity metric we choose but defining a similarity metric is not trivial in a network with only link information.

Other traditional techniques include k-means algorithm [60, 63] and spectral clustering[22, 26]. K-means algorithm requires that data can be embedded into a metric space. Although the value of k needs to be determined beforehand, we can run this algorithm multiple times to find a good k. On the other side, similar to hierarchical clustering, spectral clustering requires the similarity matrix or other matrices derived from it as well. By using the eigenvectors of this matrix, spectral clustering can map the nodes in a network to a space whose basis are these eigenvectors. Not surprisingly, both k-means algorithm and spectral clustering also suffer from the issue that a network with only link information is difficult to be transformed into some data points in space or find a similarity metric.

Modularity-based Approaches

Modularity is by far the most popular quality function to measure how good the detected communities are. First proposed by Newman and Girvan [72], modularity is based on the as-

sumption that a null model, i.e., a graph where the choice of every node pair to have a link is regardless of any other node pair, does not have community structure. If the density of edges in subgraph is significantly larger than the expected density in the null model, we can identify this subgraph as a community. Specifically, modularity is defined as

$$Q = \frac{1}{2m} \sum_{u,v \in V} (A_{u,v} - P_{u,v}) \mathbb{I}_{u,v}, \tag{2.2}$$

where m is the number of links, V is the node set, A is the adjacency matrix, $P_{u,v}$ is the expected number of links between u and v in the null model, and $\mathbb{I}_{u,v}$ is an indicator function of whether u and v belong to the same community.

In fact, a link between u and v consists of two stubs, i.e., half-links. By maintaining the degree of each node, the probability p_u to pick a stub coming from node u is $\frac{d(u)}{2m}$ and the same applies on node v, where where $d(u)$ is the degree of node u. So the probability of a link between u and v is given by the product of p_u and p_v, since links are independent of each other. Thus, the expected number of links between u and v of the null model, i.e., $P_{u,v}$, can be computed as $P_{u,v} = \frac{d(u)d(v)}{4m^2} * 2m = \frac{d(u)d(v)}{2m}$. Putting it into Equation (2.2), the formula becomes

$$Q = \frac{1}{2m} \sum_{u,v \in V} (A_{u,v} - \frac{d(u)d(v)}{2m}) \mathbb{I}_{u,v}, \tag{2.3}$$

Due to the existence of $\mathbb{I}_{u,v}$, only node pairs in the same community contributes to Equation (2.3). As a result, we can

rewrite Equation (2.3) as a sum over the communities

$$Q = \sum_{c=C_1}^{C_p} [\frac{l_c}{m} - (\frac{d_c}{2m})^2], \tag{2.4}$$

where p is the number of communities in the network, l_c is the number of internal links in community c, and d_c is the sum of the degree of all the node in community c.

It can be inferred from Equation (2.4) that modularity will be larger if a community is denser or has fewer links connecting to the outside. The modularity of the whole network is always zero if no partition is made. In other words, if we cannot find a partition with positive modularity, the network has no community structure. Also, modularity is always smaller than one and can be negative.

In addition to being a quality function, modularity can be used as an optimization objective to directly detect communities. Modularity optimization is by far the most widely-used algorithms for classic community detection. As it has been proved that modularity optimization is an NP-complete problem [12], several heuristics have been proposed to find fairly good results in a reasonable amount of time.

The first heuristic of modularity optimization for community detection is a greedy algorithm of Newman [69]. It starts from n groups with each consisting of only one node. In each step, two groups will be merged if the modularity increase is maximum compared with the previous community structure. Since

the number of possible merging choice is massive, especially in early steps, computing modularity for all the choices requires a lot of time. Clauset et al. employs *heap* to reduce the time complexity from $O(n^2)$ to $O(n \log^2 n)$ [16]. However, this greedy strategy tends to merge two large groups instead of two small ones since it yields more modularity increase if two large groups are densely connected with each other. To alleviate this issue, Danon et al. introduce a normalizer when computing the modularity variation ΔQ [19]. Other tricks include (1) starting from some intermediate structure instead of the initial structure [23, 81, 104], (2) merging more than two communities at a time [92, 93]. Both turn out to dramatically improve the accuracy of the greedy algorithm. Also, Blondel et al. generalize the greedy algorithm into the case of weighted graph [10].

Since a greedy algorithm tends to be trapped into local optima, other methods, in fact, achieve higher accuracy. Guimerà et al make use of simulated annealing, a probabilistic process for global optimization [44], on modularity optimization [37]. Duch and Arenas, on the other hand, employ a local optimization method called *extremal optimization (EO)* [11] to obtain a comparable accuracy but slightly faster running time [24]. Another approach is called spectral optimization which defines a modularity matrix to rewrite the formula of modularity so that we can compute its eigenvalues and eigenvectors to optimize modularity on a graph of two communities [70].

Despite that modularity optimization is the most popular

class of approaches in classic community detection, it suffers from a severe problem called resolution limit [31]. The resolution limit prevents modularity optimization methods from detecting communities with small sizes compared to the whole graph. To be more precise, one cannot determine whether a community is a single community or a combination of smaller weakly-interconnected communities when the partition with maximum modularity consists of communities with a total degree of the order of $\sqrt{m}$ or smaller, where m is the number of links. The resolution limit has a strongly negative impact on practical applications. One possible way to overcome this issue is to conduct further partitions in large communities detected by modularity optimization algorithms [31, 89]. However, it is difficult to decide when to stop this process. Another solution is to use a different scoring function which incorporates both internal and external structures [91].

Other Approaches

In this part, we mainly explore two more classes of popular approaches for classic community detection.

One class is divisive algorithms. Similar to divisive hierarchical clustering methods, divisive algorithms identify communities by finding inter-community links and removing them, which makes the communities disconnected from each other. However, the main difference between divisive algorithms and divisive hierarchical clustering methods is that divisive algorithms aim to re-

move inter-community links instead of links between node pairs with low similarity. Thus, divisive algorithms need a more global metric than divisive hierarchical clustering methods to identify the border of communities. In the earliest algorithm proposed by Girvan and Newman [35, 72], link betweenness, i.e., the number of shortest paths that pass through the link, is used as the estimator of the importance of a link. In each step, the betweenness of all links is computed and then the link with the largest betweenness is removed. Tyler et al. proposed a modification of the Girvan-Newman algorithm by calculating edge betweenness only from a limited number of randomly sampled centers [100]. It improves the speed of the computation and can be applied to large networks such as networks of gene co-occurrences. Another possible way to detect inter-community links is to identify cycles. A community usually has a high density of links so that it is common to have cycles inside it. In contrast, inter-community links can hardly form cycles. Based on this idea, a new measure called edge clustering coefficient is proposed by Radicchi et al. [82] to measure the likelihood of being an inter-community link.

Another class employs label propagation as an efficient way to solve the community detection problem. Raghavan et al. first propose a *label propagation algorithm (LPA)* which can detect communities in a very fast speed [83]. This algorithm initially assigns a unique community label to each node in the network. At every propagation step, each node sequentially updates its

label to the most frequent label among its neighbors. A tie is broken randomly. The algorithm stops when all labels no longer change. Barber and Clark extend *LPA* by relating it to modularity [7] and Liu et al. further combine it with a greedy agglomerative algorithm to escape local maxima [59].

2.2 Overlapping Community Detection

2.2.1 Problem Description

Much of the focus in classic community detection lies in finding *disjoint* communities. However, multiple community memberships are quite common in real-world networks [86]. For example, a person in a social network has multiple social identities, an author in a collaboration network has publications in multiple venues, one kind of protein in a protein-protein interaction network has multiple biological functions, etc. To break the restriction brought by unique community membership, overlapping community detection becomes the main trend in the research of community detection.

The definition of overlapping community detection has no difference with classic community detection except that there is no disjoint constraint on detected communities. Despite little modification on the definition, most of the methods of classic community detection can no longer be directly applied to overlapping community detection anymore. Thus, the demand for

algorithms specifically designed for overlapping community detection is getting bigger and bigger in the last decade.

2.2.2 Literature Review

Several surveys in the area of community detection have discussed overlapping community detection as one of the subjects [17, 30] or compare both disjoint community detection algorithms and overlapping community detection algorithms [39, 50, 56]. A comprehensive survey dedicated for overlapping community detection can be found in [105]. Different from all existing surveys, we classify overlapping community detection approaches into local approaches and global approaches based on the breakthrough idea. We will introduce some important works in both categories in this part.

Local Approaches

Local approaches employ a divide-and-conquer strategy which usually consists of three main phases, i.e., dividing, conquering, and adjustment. In the dividing phase, a network is divided into multiple small subgraphs. In the conquering phase, a particular community detection algorithm is performed on each subgraph to obtain initial communities. In the end, an adjustment is conducted either locally or globally by merging densely overlapped initial communities into a new community in the final community structure. In some cases, the dividing phase and the con-

quering phase interact heavily with each other. In other cases, the adjustment phase is skipped.

Clique percolation (CP) is the earliest overlapping community detection algorithm [74]. The *CP* algorithm first identifies for all k-cliques, i.e., fully-connected subgraphs with size k, and then transform the original network into a clique graph where a node represents a k-clique and two nodes are connected if they share $n-1$ members. The final communities are all connected components of the clique graph. Through experiments on synthetic datasets, small values of k give good results. However, the polynomial time complexity of *CP* is still too large for large networks with millions of nodes. Kumpula et al. propose a two-phase *sequential clique percolation (SCP)* algorithm [49]. In the first phase, it finds k-cliques by checking $(k-2)$-cliques in the common neighbors of two endpoints when links are inserted sequentially to the network. In the second phase, it turns the original network into a bipartite graph with two types of nodes which denotes k-cliques and $(k-1)$-cliques respectively and obtains the final communities according to this bipartite graph. The running time of *SCP* grows linearly as the number of k-cliques and thus can deal with networks of larger sizes.

Baumes et al. propose an algorithm with two steps, *RankRemoval* and *Iterative Scan (IS)* [9]. *RankRemoval* calculates the rankings of all nodes and then continuously removes top nodes until the network is split into small, disjoint connected components. These connected components are regarded as seeds and

IS adjusts them by adding or removing nodes until a density function is maximized. The density function is defined as

$$f(c) = \frac{m_{in}^c}{m_{in}^c + m_{out}^c}, \tag{2.5}$$

where m_{in}^c and m_{out}^c are the total number of internal and external links of community c. Thus, the quality of detected communities is highly related to the quality of seeds.

Besides [9], we will discuss a few more local approaches based on seed expansion. *LFM* [51] randomly selects a seed node to construct a community by adding or removing nearby nodes until the fitness function

$$f(c) = \frac{k_{in}^c}{(k_{in}^c + k_{out}^c)^\alpha}, \tag{2.6}$$

reaches its local maximal, where k_{in}^c and k_{out}^c are the total internal and external degree of community c, and α is the resolution parameter deciding community size. After a community is discovered, *LFM* will randomly select a node from the rest of the network until all communities are detected. Havemann et al. propose a smoothness term on the fitness function, i.e.,

$$f(c) = \frac{k_{in}^c + 1}{(k_{in}^c + k_{out}^c)^\alpha}, \tag{2.7}$$

which allows a community to only consist of a single node [40]. Whang et al. employs a kernelized distance function to determine seeds and the personalized *PageRank* algorithm to expand a seed to a community [103]. *OSLOM* [52] employ statistical significance test on a cluster compared with a global null model

to determine when to cease expansion from a seed. However, the result of *OSLOM* tends to leave a large number of nodes with no community memberships.

Disjoint community detection algorithms can also be modified to detect overlapping communities. If we cluster links instead of nodes in a network and assign both endpoints of a link to the community this link belongs to, we can get overlapping communities. This idea is first proposed by Ahn et al. as *link clustering* [1]. In this algorithm, links are clustered via hierarchical clustering and the similarity metric between two links e_{ik} and e_{jk} (incident on node k) is defined as the Jaccard similarity between the neighborhood of node i and the neighborhood of node j, i.e.,

$$s(e_{ik}, e_{jk}) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}. \tag{2.8}$$

Link clustering is not suitable for detecting densely overlapping communities since a link cannot belong to more than one communities.

Other local approaches apply disjoint community detection algorithms on the conquering phase and merge initial disjoint communities into overlapping communities in the adjustment phase. For example, Coscia et al. apply label propagation algorithm on ego network of each node, i.e., a node with its neighbors, to detect initial communities and then merge communities with large overlap [18]. Li et al. propose a *Local Expansion via Minimum One Norm (LEMON)* algorithm to expand the

seeds by searching for a sparse vector in the span of the local spectra such that the seeds are in its support [57].

Global Approaches

Global approaches, on the other side, perform overlapping community detection from a global view. They usually assume null communities first and start to adjust the memberships of each node for multiple rounds.

As a popular framework, stochastic block model [73] has already been widely employed in disjoint community detection [42, 79]. In a stochastic block model, we have N nodes and K blocks, and each node belongs to only one of the K blocks. We define an indicator matrix $Z \in \{0,1\}^{N\times K}$, where Z_{ir} represents whether node i belongs to block r. We also define a relationship matrix $B \in [0,1]^{K\times K}$, where B_{pq} represents the probability of connections between nodes from block p and block q, respectively. Given B and Z, we can finally define a probability matrix $\Theta = ZBZ^T$, where Θ_{ij} denotes the link probability between node i and node j. With the adjacency matrix being available from data, the goal is to estimate Z. Several works manage to generalize the stochastic block model for overlapping community detection. For example, *mixed membership stochastic block model (MMSB)* [3] allows a fixed number of memberships for a node so that Z is a mixed membership matrix instead of an indicator matrix. A general variational inference algorithm is applied for fast approximate posterior inference of Z. *Overlap-*

ping stochastic block model (OSBM) [53] exploits a multivariate Bernoulli distribution instead of a Dirichlet distribution in [3] to generate Z. A very recent work of Jin et al. models Z as a probability matrix whose row sum is 1 and preserves node degree in each of the probabilistic communities to learn model parameters [41].

Apart from the stochastic block model, several global frameworks originally used in other areas also receive attention for overlapping community detection. Game theory has been a classic tool with various applications. Chen et al. propose a game-theoretical framework to identify overlapping communities in social networks [14]. In the strategic game this work plays, each node is a selfish agent and its actions include joining or leaving a community. The utility function is defined as the combination of a gain function based on modularity and a loss function related to the number of communities one node joins. The equilibrium of this game is interpreted as the targeted community structure. McAuley and Leskovec propose a generative node clustering framework to discover social circles in ego networks [64]. This framework first encodes both community characteristics and pairwise node similarities into features, which are used to model the link probability. Then a generative objective function is constructed for parameter learning according to the adjacency matrix. As another popular framework, matrix factorization has been widely applied in recommender systems and other areas. Since the models proposed by this thesis are all

built on the matrix factorization framework, we will introduce this framework in more detail in the following section.

2.3 Matrix Factorization Framework for Overlapping Community Detection

2.3.1 Problem Description

Matrix factorization (MF) has been a standard technique in areas such as recommender systems [45, 62], image processing [54], natural language processing [106], bioinformatics [13], etc. Although it has many variations [55, 66, 90, 95], the main mathematical form can be summarized as follows.

Definition 2.3 (**Matrix Factorization**)**.** Given a matrix $R^{m\times n}$, the objective of matrix factorization is to find two matrices $U^{m\times k}$ and $V^{n\times k}$ whose product can minimize a particular loss function l, i.e.,

$$\arg\min_{U,V} l(R, UV^T), \tag{2.9}$$

where m is the size of data, n is the dimension of data, and k is the dimension of latent space.

When m and n are large but R is sparse, matrix factorization is one of the most suitable frameworks to learn the unknown values in R with $k << m$ and $k << n$. What matters most is the choice of learning objective. Sometimes, the form of factorization can also be modified. For example, some models employ

matrix tri-factorization (MTF) which factorize R into the product of three matrices USV^T. Moreover, constraints can be added to the factorized matrices, e.g., the non-negative constraints on U and V making matrix factorization into non-negative matrix factorization (NMF). All the above modifications are based on the task we want to solve.

Matrix factorization is suitable for overlapping community detection due to the following advantages:

1. adjacency matrix can be used as the input matrix,
2. the communities can be regarded as the latent space, i.e., k becomes the number of communities,
3. the output matrix is naturally soft-partitioning, i.e., communities are allowed to overlap,
4. it does not suffer the resolution limit problem, which is one of most severe drawbacks in modularity optimization approaches.

2.3.2 Literature Review

There is no comprehensive survey by far on the matrix factorization framework for overlapping community detection. I will briefly review some of the most representative works in chronological order.

A Bayesian NMF Model

The Bayesian non-negative matrix factorization (BNMF) model is the first MF-based model utilized in overlapping community detection [80]. Based on a generative graphical model, the BNMF model assumes that there are K communities and a scale hyper-parameter $\beta = \{\beta_k\}$ for different communities. The adjacency matrix G is influenced by an unobserved *expectation network* $G' \in \mathbb{R}^{N\times N}$ and G' is composed of two non-negative matrices $W \in \mathbb{R}^{N\times K}$ and $H \in \mathbb{R}^{K\times N}$ so that $G' = WH$. k-th column in W and k-th row H are correspondent to the k-th community, thus are both affected by β_k. The joint distribution over all variables is

$$\mathcal{P}(G, W, H, \beta) = \mathcal{P}(G|W, H)\mathcal{P}(W|\beta)\mathcal{P}(H|\beta)\mathcal{P}(\beta), \qquad (2.10)$$

hence the objective function is to maximize the model posterior given the observations, i.e., $\mathcal{P}(W, H, \beta|V)$, or equivalently, to minimize the negative log posterior

$$U = -\log \mathcal{P}(G|W, H) - \log \mathcal{P}(W|\beta) - \log \mathcal{P}(H|\beta) - \log \mathcal{P}(\beta). \qquad (2.11)$$

Each part of Equation (2.11) is modeled with a certain distribution with some prior. The final objective function is written

as

$$U = \sum_i \sum_j [g_{ij} \log(\frac{g_{ij}}{g'_{ij}}) + g'_{ij}] + \frac{1}{2} \sum_k [(\sum_i \beta_k w_{ik}^2) + (\sum_j \beta_k h_{kj}^2) - 2N \log \beta_k] + \sum_k (\beta_k b_k - (a_k - 1) \log \beta_k) + c, \tag{2.12}$$

where c is a constant. A fast fix-point algorithm with consecutive updates is adopted for the optimization process for W, H, and β. The solution includes $W^* \in \mathbb{R}^{N \times K^*}$ and $H^* \in \mathbb{R}^{K^* \times N}$ for which $G' = W^* H^*$ and K^* is the inferred number of latent communities. When the graph is undirected, W^* is expected to be the transpose of H^*.

Although the BNMF model is a good attempt to employ matrix factorization into overlapping community detection, it requires many prior assumptions and the time complexity for an update is too high to deal with a large network with millions of nodes.

An NMF Model for Different Types of Networks

Wang et al. propose three NMF models to target three different types of networks (undirected, directed and compound) [101]. They directly apply the Euclidean loss $\mathcal{L}(A, B) = ||A - B||_F^2 = \sum_{i,j}(A_{ij} - B_{ij})^2$ as the objective function for all three models.

For the undirected case, the objective function is to minimize $||G - XX^T||_F^2$, where G is adjacency matrix and X is the scale partition matrix of the network whose i-row corresponds to the

community membership of the i-th node. X can be further normalized with $\sum_j X_{ij} = 1$ such that X_{ij} represents the posterior probability that i-th node is associated with k-th community. X can be solved by the multiplicative update rule

$$X_{ik} \leftarrow X_{ik}(\frac{1}{2} + \frac{(GX)_{ik}}{(2XX^TX)_{ik}}). \tag{2.13}$$

For the directed case, since the adjacency matrix is asymmetric, we need to introduce matrix tri-factorization. The new objective function is to minimize $||G - XSX^T||_F^2$. Under the alternative update rules

$$\begin{aligned} X_{ik} &\leftarrow X_{ik}(\frac{[G^TXS + GXS^T]_{ik}}{[XSX^TXS^T + XS^TX^TXS]_{ik}})^{\frac{1}{4}}, \\ S_{kl} &\leftarrow S_{kl}\frac{[X^TGX]_{kl}}{[X^TXSX^TX]_{kl}} \end{aligned} \tag{2.14}$$

the authors can guarantee that the loss in non-increasing.

For compound networks, the authors take a movie recommendation as an example, where U denotes the user-user matrix, D denotes the movie-movie matrix, and M denotes the user-movie matrix. The target is to find a latent X, which minimizes all three parts of the loss function $||M - X||^2 + \alpha||U - XX^T||^2 + \beta||D - X^TX||^2$ simultaneously, where $\alpha > 0$ and $\beta > 0$ are importance coefficients. The authors also guarantee that the loss in non-increasing under the multiplicative update rule

$$X_{ij} \leftarrow X_{ij}(\frac{[M + 2\alpha UX + X\hat{D}]_{ij}}{2(\alpha + \beta)[XX^TX]_{ij}})^{\frac{1}{4}}, \tag{2.15}$$

where $\hat{D} = 2\beta D - I$.

Although the objective functions of all three NMF models are straightforward without any data assumptions, it is obvious that the multiplicative update involves too many matrix multiplications and thus is computationally inefficient.

A Bounded Non-negative MTF Model

The bounded non-negative matrix tri-factorization (BNMTF) model uses three factors to learn the community membership of each node as we as the interaction among communities [113]. The BNMTF model considers a weighted graph with a non-negative matrix $G \in \mathbb{R}_+^{n \times n}$ as the adjacency matrix and assumes that the maximum number of possible communities k is given. It introduces a matrix $U \in \mathbb{R}_+^{n \times k}$ to denote the community membership of n nodes and $B \in \mathbb{R}_+^{k \times k}$ to denote the community interaction matrix. Each entry u_{ij} in U represents and thus is between 0 and 1. The objective is to use the product of UBU^T (denoted by $\hat{G}$) to approximate G. Two loss function, squared loss and generalized KL-divergence, are employed to measure the approximation. They are defined as

$$\begin{aligned} \mathcal{L}_{sq}(G, U, B) &= ||G - UBU^T||_F^2, \\ \mathcal{L}_{sq}(G, U, B) &= \sum_{i,j} (g_{ij} \ln \frac{g_{ij}}{\hat{g}_{ij}} - g_{ij} + \hat{g}_{ij}), \end{aligned} \tag{2.16}$$

where g_{ij} is the entry in G and $\hat{g}_{ij}$ is the entry in $\hat{G}$. In real-world networks, a node is not associated with too many communities so U is sparse. Thus, a l_1 norm is added to be the regularization

term, i.e., $||U||_1 = \mathbf{1}^T U \mathbf{1}$. The final optimization problem of the BNMTF model is formulated as

$$\min_{U,B} \mathcal{L}(G, U, B) + \lambda ||U||_1 \; s.t. \; \mathbf{0} \leq U \leq \mathbf{1}, B \geq \mathbf{0}, \tag{2.17}$$

where $\lambda > 0$ balances the trade-off between approximation error and the complexity of U.

For parameter learning, the BNMTF model uses coordinate descent methods which update one parameter at a time while fixing all the others. An auxiliary function is defined for KL-divergence loss since the original objective function has no closed-form solution. The computation requires a lot of matrix multiplication, so this model only works on networks with thousands of nodes.

An extension has been proposed by Pei et al. by taking consideration of graph regularization components including user similarity and message similarity in social networks on top of the non-negative matrix tri-factorization (NMTF) framework [78].

A Link Probability-based Model

The cluster affiliation model for big Networks (BigClam) is the first matrix factorization based model designed for large networks of millions of nodes and edges [109]. This model is built on a bipartite affiliation network $B(V, C, M)$ consisting of a node set V as one side and a community set C as the other side with M indicating node community affiliations. Then, a non-negative node community weight matrix F is used to parameterize the

affiliation between a node and a community. To be specific, each community c connects its member u and v with probability $1 - \exp^{(-F_{uc} \cdot F_{vc})}$. By assuming that each community c connects u and v independently, the probability that there is an edge between a node pair (u, v) is

$$\mathcal{P}(u, v) = 1 - \exp(F_u F_v^T). \tag{2.18}$$

Given an undirected network $G(V, E)$, the BigClam model aims to fit the underlying network G by generating exactly the same set of edges with maximum probability, i.e.,

$$\hat{F} = \arg\max_{F \geq 0} \mathcal{L}(F) = \sum_{(u,v) \in E} \log \mathcal{P}(u, v) + \sum_{(u,v) \notin E} \log(1 - \mathcal{P}(u, v)). \tag{2.19}$$

Combining Equation (2.18) and (2.19), the objective function can be written as

$$\mathcal{L}(F) = \sum_{(u,v) \in E} \log(1 - \exp(-F_u F_v^T)) - \sum_{(u,v) \notin E} F_u F_v^T. \tag{2.20}$$

To solve the above optimization problem, the BigClam model adopts a block coordinate gradient ascent algorithm which updates F_u for each u with the other F_v fixed. As a result, the problem of updating F_u becomes a convex optimization problem. The gradient of F_u can be computed straightforwardly by

$$\frac{\partial L}{\partial F_u} = \sum_{v \in \mathcal{N}(u)} F_v \frac{\exp(-F_u F_v^T)}{1 - \exp(-F_u F_v^T)} - \sum_{v \notin \mathcal{N}(u)} F_v, \tag{2.21}$$

where $\mathcal{N}(u)$ is u's neighbors. According to Equation (2.21), a single step of updating F_u takes linear time $O(N)$. However, by

replacing $\sum_{v \notin \mathcal{N}(u)} F_v$ with $(\sum_v F_v - F_u - \sum_{v \in \mathcal{N}(u)} F_v)$, the time complexity can be reduced to $O(|\mathcal{N}(u)|)$. After each update, F_u is projected to the non-negative space by $F_{uc} = \max(F_{uc}, 0)$ to maintain the non-negative constraint.

□ **End of chapter.**

Chapter 3

A Preference-based NMF Model

Community detection is an important technique to understand structures and patterns in complex networks. Recently, overlapping community detection becomes a trend due to the ubiquity of overlapping and nested communities in the real world. However, existing approaches have ignored the use of implicit link preference information, i.e., links can reflect a node's preference on the targets of connections it wants to build. This information has a high impact on community detection since a node prefers to build links with nodes inside its community than those outside its community. In this chapter, we propose a preference-based nonnegative matrix factorization (PNMF) model to incorporate implicit link preference information. Unlike conventional matrix factorization approaches, which simply approximate the original adjacency matrix in value, our model maximizes the likelihood of the preference order for each node by following the intuition that

a node prefers its neighbors than other nodes. Our model overcomes the indiscriminate penalty problem in which non-linked pairs inside one community are equally penalized in objective functions as those across two communities. We propose a learning algorithm which can learn a node-community membership matrix via stochastic gradient descent with bootstrap sampling. We evaluate our PNMF model on several real-world networks. Experimental results show that our model outperforms state-of-the-art approaches and can be applied to large datasets.

3.1 Introduction

Discovering the community structure in complex networks has been extensively investigated in the past decade [30]. A community is intuitively regarded as a group of nodes with more links inside the group than between its member and outside the group [35]. In the real world, communities can be social circles manually categorized by users in ego networks [64], authors from the same institution in collaboration networks [68], proteins with the same functionality in biochemical networks [33], etc. The research issue of finding such groups is known as the *community detection* problem.

Classic methods for community detection assume that one node belongs to exactly one community. However, many complex networks we encounter in daily life allow multiple memberships. For example, two colleagues in the same department are

also in the same company (nested), one can join in several discussion groups in an online forum (overlapping), etc. Thus, the topic of overlapping community detection has attracted major attention recently [105].

Existing overlapping community detection approaches can be categorized into two classes: one is based on dense subgraph extraction [1, 74, 49], which uses certain criteria to find overlapping dense subgraphs or clusters in the network to be communities; the other is based on community affiliation model [80, 101, 109, 113], which determines the number of communities in advance and assigns each node to multiple communities according to some optimization function. However, both classes of approaches only focus on links themselves but ignore the implicit preference information in links. In fact, a link can reflect the preferences of both sides to some extent. For example, in a social network, if user A wants to make friend with user B, a typical way for A is to send a friend invitation to B and wait for him to accept it. They cannot be friends if either step goes wrong. Thus, when we see the fact that A and B are friends, it is reasonable to argue that A prefers B than other strangers to be his friend. Assuming B also receives other people's invitations and only accepts a few of them (this is very likely to happen in the real world), we can also argue that B prefers A than others who are still strangers to him. Following the intuition that a node is more likely to build links with other nodes in the same community than those outside its community, the implicit

preference information can be helpful for community detection.

For the second class of approaches, i.e., community affiliation based approaches, nonnegative matrix factorization (NMF) has been applied as a standard technique. The basic idea of NMF technique is to find a node-community membership matrix F ($F_{u,c}$ represents the weight of node u in community c) and approximate the adjacency matrix G via FF^T. Existing approaches use either the conventional least squares error or the generalized KL divergence as objective function [55]. However, both objective functions try to approximate the adjacency matrix G in value, which are inevitable to cause the indiscriminate penalty problem. Let us assume that there are two non-linked pairs (i, j) and (i, k), where i, j belong to the same community while i, k do not. Since i, j both have positive weights in some community c, $F_i F_j^T$ is positive. However, existing NMF-based approaches will penalize $F_i F_j^T$ for being positive since $G_{i,j} = 0$. Thus, there is no difference between j and k for i, which is against the intuition that for node i, node j in the same community is preferable than node k outside i's community. In fact, it is reasonable that $F_i F_j^T$ is higher than $F_i F_k^T$, and indiscriminately penalizing the two pairs are problematic.

In this chapter, we present a preference-based nonnegative matrix factorization (PNMF) model that not only fixes the indiscriminate penalty problem of previous NMF based models but also incorporates the implicit link preference information into the model formulation. Our model uses a new objective

function, which maximizes the likelihood of a pair-wise preference order for each node. In other words, from a node's perspective, we manage to ensure that the preference of any of its friends is higher than any of other nodes. When factorizing the adjacency matrix with node-community relationship matrix, our model gives no penalty to a non-zero value appearing in the position of a non-linked pair, as long as all the pairwise preferences are preserved. Thus, this objective function can be regarded as a relaxation of previous approaches. We exploit stochastic gradient descent with bootstrap sampling to solve the optimization problem. We conduct experiments in several real world datasets including some with ground-truth communities. By comparing our model with several state-of-art approaches, we show that our model can detect overlapping communities with higher quality on widely-used metrics in community detection. It can also be applied to large datasets.

3.2 Related Work

Baysian Personalized Ranking (BPR) [88] is proposed to rank items for a specific user in recommender systems while only implicit feedback (e.g. clicks) is available. The basic assumption is that a user prefers labeled items than unlabeled ones. While traditional methods replace missing values with zeros or negative ones, BPR uses pairwise preference as training data to learn the model parameters. Technically, it maximizes a posterior

probability $p(\Theta| >_u)$ where Θ is the parameter and $>_u$ is the latent preference structure for user u. We adopt this idea into our overlapping community detection task for a different learning goal of maximizing the probability $p(>_u |F)$, where F is a nonnegative matrix representing the latent node-community membership. BPR has become a classical model in one-class collaborative filtering, and there are several further works on top of it. For example, Rendle et al. extend the original matrix factorization to a tensor factorization to recommend personalized tags for a user given an item [87]. Zhao et al. leverage social connections to improve item recommendations by building a new preference system [114].

3.3 Community Detection via PNMF

In this section, we present our PNMF model in the context of overlapping community detection and propose a stochastic gradient descent method with bootstrap sampling to learn model parameters.

3.3.1 Preliminaries

Given an unweighted and undirected network $N(V, E)$, where V is the set of n nodes and E is the set of m edges, we can obtain its adjacency matrix $G \in \{0, 1\}^{n\times n}$ whose (i, j) entry $g_{i,j}$ is an indicator of whether node i and node j are connected. Since the network is undirected, G is a symmetric matrix.

We denote the set of communities by C and the number of communities by p. We use a nonnegative matrix $F \in \mathbb{R}_+^{n\times p}$ to denote the node-community membership for all the nodes. Each entry $F_{u,c}$ represents the weight between node $u \in V$ and community $c \in C$. The larger $F_{u,c}$ is, the more possible that u belongs to c. On the other hand, if $F_{u,c}$ is 0, u does not belong to c.

Given the information above, the objective is to recover G with some properties preserved by a nonnegative matrix factorization FF^T, i.e.,

$$G \approx FF^T. \tag{3.1}$$

Previous approaches simply approximate G in value. They expect $F_u F_v^t$ to be close to 1 if u, v are linked and to be 0 otherwise. In our model, we preserve the preference orders observed in G for all the nodes. We will discuss the details later.

The set of i's neighbors is denoted by $N^+(i)$. In addition, we define $N^-(i) := N^+(i)^c \backslash \{i\}$ to be "non-neighbors" of i, where $N^+(i)^c$ denotes the complement set of $N^+(i)$. By definition, $V = N^+(i) \cup N^-(i) \cup \{i\}$ for every i. Moreover, we define a learning set $S : V \times V \times V$ by

$$S = \{(i, j, k) | i \in V, j \in N^+(i), k \in N^-(i)\},$$

which consists of all the triples (i, j, k), where j is a neighbor of i while k is not.

In the end, we list three basic assumptions on implicit link preference to make model formulation clearer.

1. **Node independence**. Each node determines its preferences independently. The network can be regarded as a result after all the nodes make their decisions. Specifically, a link will be built between u and v if and only if u has a high preference on v and symmetrically v has a high preference on u.

2. **Higher preference on neighbors**. Let $u >_i v$ denote that node i prefers node u than node v. For a fixed node i, we have $j >_i k$ if $j \in N^+(i)$ and $k \in N^-(i)$, but no preference information between j and k is indicated if $j, k \in N^+(i)$ or $j, k \in N^-(i)$. Thus, the use of the learning set S is to record all the single triples (i, j, k) satisfying that i prefers to build a link with j than k.

3. **Pair independence**. For a fixed node i, its preference on j and k is independent with its preference on u and v when $j, u \in N^+(i)$ and $k, v \in N^-(i)$.

3.3.2 Model Formulation

Based on our motivation, we aim to find the node-community membership matrix, which maximizes the likelihood of observed preference order for all the nodes. According to the "node independence" assumption, the overall likelihood can be denoted as a product of likelihood of each node. Thus, our objective

function can be written as

$$\max_{F \in \mathbb{R}_+^{n \times p}} \prod_{i \in V} \mathcal{P}(>_i | F), \tag{3.2}$$

where $>_i$ denotes the observed preferences for node i and F is the node-community membership matrix.

According to the "high preference on neighbors" assumption and the "pair independence" assumption, the probability of preference order for a single node i can be written as

$$\begin{aligned} p(>_i | F) = & \prod_{(j,k) \in V \times V} \mathcal{P}(j >_i k | F)^{\delta(j \in N^+(i))\delta(k \in N^-(i))} \\ & \cdot (1 - \mathcal{P}(j >_i k | F))^{1 - \delta(j \in N^+(i))\delta(k \in N^-(i))} \\ = & \prod_{(j,k) \in V \times V} \mathcal{P}(j >_i k | F)^{\delta((i,j,k) \in S)} \\ & \cdot (1 - \mathcal{P}(j >_i k | F))^{\delta((i,j,k) \notin S)}, \end{aligned} \tag{3.3}$$

where S is the learning set mentioned in preliminaries and δ is the indicator function

$$\delta(a) = \begin{cases} 1 & \text{if } a \text{ is true,} \\ 0 & \text{else} \end{cases}.$$

For a triple (i, j, k), if $(i, j, k) \in S$, then $(i, k, j) \notin S$. Given $\mathcal{P}(j >_i k | F) + \mathcal{P}(k >_i j | F) = 1$, it is easy to see that $\mathcal{P}(j >_i k | F)^{\delta((i,j,k) \in S)} = (1 - \mathcal{P}(k >_i j | F)^{\delta((i,k,j) \notin S)}$. Applying this to Equation (3.3), maximizing $\mathcal{P}(>_i | F)$ is equivalent to

$$\max_{F \in \mathbb{R}_+^{n \times p}} \prod_{(j,k) \in V \times V} \mathcal{P}(j >_i k | F)^{\delta((i,j,k) \in S)}. \tag{3.4}$$

Combining Equation (3.2) and (3.4), our objective function can be rewritten as

$$\max_{F \in \mathbb{R}_+^{n \times p}} \prod_{(i,j,k) \in S} \mathcal{P}(j >_i k | F). \tag{3.5}$$

Based on the intuition that two nodes have a higher probability to be linked if they share more communities, we define the probability that i prefers j than k given the node-community membership matrix as

$$\mathcal{P}(j >_i k | F) = \sigma(F_i \cdot F_j^T - F_i \cdot F_k^T), \tag{3.6}$$

where σ is the sigmoid function $\sigma(x) := \frac{1}{1+e^{-x}}$.

The sigmoid function can map any real number into $(0, 1)$. We can see that the probability i prefers j than k is 0.5 when $F_i F_j^T = F_i F_k^T$. Also, this probability is close to 0 when $F_i F_j^T \ll F_i F_k^T$ and is close to 1 when $F_i F_j^T \gg F_i F_k^T$. These properties precisely characterize the requirements of our model.

For simplicity, we define $\hat{x}(i, j) := F_i \cdot F_j^T$. Equation (3.6) can be rewritten as

$$\mathcal{P}(j >_i k | F) = \sigma(\hat{x}(i, j) - \hat{x}(i, k)). \tag{3.7}$$

Now combining Equation (3.5), (3.6), and (3.7), the final

objective function of our PNMF model is

$$\begin{aligned} l(F) &:= \max_{F \in \mathbb{R}_+^{n \times p}} \ln \prod_{(i,j,k) \in S} \mathcal{P}(j >_i k|F) - \lambda \cdot reg(F) \\ &= \max_{F \in \mathbb{R}_+^{n \times p}} \sum_{(i,j,k) \in S} \ln \mathcal{P}(j >_i k|F) - \lambda \cdot reg(F) \\ &= \max_{F \in \mathbb{R}_+^{n \times p}} \sum_{(i,j,k) \in S} \ln \sigma(\hat{x}(i,j) - \hat{x}(i,k)) - \lambda \cdot reg(F), \end{aligned} \tag{3.8}$$

where $reg(F)$ is the regularization term we add to avoid overfitting and λ is the regularization parameter. We choose Frobenius norm as the regularization term, i.e., we set $reg(F) = ||F||_F^2$, since it is differentiable and fits our parameter learning process.

3.3.3 Parameter Learning

To make our model applicable to large datasets, we employ the widely used stochastic gradient descent (SGD) as our learning approach. In each update step, SGD randomly selects a triple in learning set S and updates the corresponding model parameters Θ by walking along the gradient direction,

$$\Theta^{t+1} = \Theta^t + \alpha \frac{\partial l}{\partial \Theta}, \tag{3.9}$$

where α is the learning rate. Specifically, the derivative of Equation (3.9) is calculated by

$$\begin{aligned} \frac{\partial l}{\partial \Theta} &= \frac{\partial}{\partial \Theta} \ln \sigma(\hat{x}(i,j) - \hat{x}(i,k)) - \lambda \frac{\partial}{\partial \Theta} reg(F) \\ &= \frac{-e^{\hat{x}(i,k) - \hat{x}(i,j)}}{1 + e^{\hat{x}(i,k) - \hat{x}(i,j)}} \cdot \frac{\partial}{\partial \Theta} (\hat{x}(i,j) - \hat{x}(i,k)) - \lambda \Theta \end{aligned} \tag{3.10}$$

and

$$\frac{\partial}{\partial \Theta}(\hat{x}(i,j) - \hat{x}(i,k)) = \begin{cases} F_{j,t} - F_{k,t} & \text{if } \Theta = F_{i,t} \\ F_{i,t} & \text{if } \Theta = F_{j,t} \\ -F_{i,t} & \text{if } \Theta = F_{k,t} \\ 0 & \text{else} \end{cases}, \quad (3.11)$$

where λ is the regularization parameter. Regarding the nonnegative constraints, we exploit the idea of projected gradient methods for NMF [58], which maps the value of a parameter back to nonnegativity.

Input: G, the adjacency matrix of original graph
Output: F, the node-community membership matrix

```
initialize F
compute initial loss
repeat
    for num_samples = 1 to |E| do
        sample node i from V uniformly at random
        sample node j from N+(i) uniformly at random
        sample node k from N−(i) uniformly at random
        for each entry Θ in F_i, F_j and F_k do
            update Θ according to Equation (3.9), (3.10), (3.11)
            Θ ← max(Θ, 0)
        end for
    end for
    compute loss
until convergence or max_iter is reached
```

Algorithm 1: Community detection via PNMF

The whole process of parameter learning is described in Algorithm 1. As we can see, the time complexity of each iteration

is $O(mp)$, where m is the number of links, and p is the number of community. The space complexity is $O(np)$, where n is the number of nodes since we need to save the node-community membership matrix into memory.

3.3.4 Other Issues

Choosing the number of communities. Before learning the parameters, we need to set the number of communities p in advance. However, we have no prior knowledge about it. Here we adopt the approach in [2]. We first reserve 10% of links as the validation set. Then we vary p and learn model parameters with the remaining 90% of links for each p. After that, we use the node-community membership matrix F to generate the adjacency matrix G and use G to predict the links in the validation set according to our motivation that linked pairs have a higher value than non-linked pairs in G. Finally, we pick the p with the best prediction score as our pre-assigned number of communities.

Setting membership threshold. After we learn F, we need to set a threshold δ in order to determine whether a node belongs to a community or not. If $F_{u,c} \geq \delta$, we say that node u belongs to community c. According to Equation (3.6), we need $p(j >_i k|F)$ to be closer to 1 than 0 if i prefer j than k. We assume that i, j share exactly one community and i, k do not share any communities. Thus $F_i F_k^T = 0$. Due to the symmetry

of i and j, we have

$$\sigma(F_i F_j^T - F_i F_k^T) = \sigma(\delta^2 - 0) = \frac{1}{1 + e^{-\delta^2}} = \beta,$$

where β is in the range of $(0.5, 1)$. When β is given, we can compute δ by

$$\delta = \sqrt{-\ln(\frac{1}{\beta} - 1)}. \tag{3.12}$$

3.4 Experiments

In this section, we conduct several experiments to compare our PNMF model with other state-of-the-art overlapping community detection approaches regarding community quality and scalability.

3.4.1 Datasets

We examine our model with several benchmark datasets available on the Internet. We separate them into two categories, one without ground-truth communities and the other with ground-truth communities. For the first category, we choose nine undirected networks collected by Newman[1] as our datasets. For the second category, three large datasets from SNAP[2] are used. Among them, *DBLP* is a co-authorship network in computer science, *Amazon* is a product co-purchase network, *YouTube* is an online social network with communities of various video

[1]http://www-personal.umich.edu/mejn/netdata

[2]http://snap.stanford.edu/data/

interests. Simple statistics for all the datasets can be found in Table 3.1, where **GT** represents whether this dataset has ground-truth communities, **V** is the number of nodes and **E** is the number of links.

Dataset	**GT**	**V**	**E**
Dolphins	N	62	159
Les Misérables	N	77	254
Books about US politics	N	105	441
Word adjacencies	N	112	425
American college football	N	115	613
Jazz musicians	N	198	2,742
Network science	N	1,589	2,742
Power grid	N	4,941	6,594
High-energy theory	N	8,361	15,751
DBLP	Y	317,080	1,049,866
Amazon	Y	334,863	925,872
YouTube	Y	1,134,890	2,987,624

Table 3.1: Statistics of twelve datasets (nine without ground-truth and three with ground-truth).

3.4.2 Baseline Methods

We select five state-of-the-art algorithms to be our baseline methods. The latter three are nonnegative matrix factorization based models, thus are highly comparable with our PNMF model.

SCP (Sequential Clique Percolation) [49]. Since the original Clique Percolation method [74] is slow when dealing with large datasets, we choose a sequential alternative, which

obtains the same performance but is much faster. For the choice of k-clique, we set k to be 4 or 5.

LC (Link Clustering) [1]. We do not manually set the threshold at which the dendrogram is cut. The algorithm automatically chooses the threshold where the maximum partition density is found. Among the detected communities, we get rid of all the communities whose size is smaller than 3 since these communities make no sense.

BNMF (Bayesian NMF) [80]. We use the classic squared loss $||G - WH^T||_F^2$ as the loss function, where G is the adjacency matrix, W and H are the results of nonnegative matrix factorization.

BNMTF (Bounded NM Tri-Factorization) [113]. To be consistent with BNMF, we also use squared loss $||G-FBF^T||_F^2$ as our loss function, where F and B are the results of nonnegative matrix tri-factorization.

BigCLAM [109]. For the number of communities, we set a minimum value and a maximum value and let the algorithm find the best choice between these two numbers based on cross-validation.

3.4.3 Metrics

We choose two well-known metrics to measure the performance of our model. The choice of metric depends on whether the specific dataset has ground-truth communities.

Modularity. We employ the most widely used modularity [71] as our measure for datasets without ground-truth communities. Since communities are overlapping in our case, we need to modify the original definition of modularity a bit. The new modularity Q is defined as

$$Q = \frac{1}{2m} \sum_{u,v \in V} (g_{u,v} - \frac{d(u)d(v)}{2m}) |C_u \cup C_v|,$$

where m denotes the number of links, V denotes the set of nodes, $g_{i,j}$ denotes the (i, j) entry of adjacency matrix G, $d(i)$ denotes the degree of node i, and C_u denotes the set of communities including u.

As we mentioned, two nodes are likely to link each other if they have common communities. Modularity matches our intuition very well in the way that more common communities two nodes have, more penalty they will receive if they do not build a link between them. $\frac{d(u)d(v)}{2m}$ can be regarded as the link probability between u and v.

$\mathbf{F_1}$ score. For datasets with ground-truth communities, we employ another criterion F_1 score to measure the quality of detected communities. We denote the set of ground-truth communities as C and the set of detected communities as $\hat{C}$. C_i represents the i-th community in C and $\hat{C}_i$ represents the i-th community in $\hat{C}$. We define F_1 score to be the average of the F_1 score of the best-matching ground-truth community to each

detected community, i.e,

$$F_1 = \frac{1}{|\hat{C}|} \sum_{\hat{C}_i \in \hat{C}} F_1(C_{b(i)}, \hat{C}_i),$$

where the best matching function $b(i)$ is defined as

$$b(i) = \arg\max_j F_1(C_j, \hat{C}_i),$$

and $F_1(\cdot, \cdot)$ is the harmonic mean of precision and recall.

3.4.4 Results

We compare our PNMF model with all the baseline methods listed above and show the results in Table 3.2. For the first nine datasets without ground-truth communities, we use modularity as our measurement. The results show that our model performs best on seven out of nine datasets. Especially, our model dominates other nonnegative matrix factorization based models (BNMF, BNMTF, BigCLAM) on all the datasets except "Jazz musicians". For the last three datasets with ground-truth communities, we use F_1 score as our measurement. We can see that our model significantly outperforms LC and is comparable with the other two methods with a fair overall advantage. Another advantage of our model is scalability. Some results are not shown because the corresponding baseline methods cannot scale to networks with such size. Only SCP, BigCLAM and our PNMF model can deal with the largest dataset, i.e. YouTube, which consists of more than one million nodes.

Dataset	**Metric**	**SCP**	**LC**	**BNMF**	**BNMTF**	**BigCLAM**	**PNMF**
Dolphins	M	0.3049	0.6538	0.5067	0.5067	0.4226	**0.9787**
Les Misérables	M	0.3066	0.7730	0.1247	0.1031	0.5395	**1.1028**
Books about US politics	M	0.4955	0.8507	0.4613	0.4924	0.5290	**0.8640**
Word adjacencies	M	0.0707	0.2705	0.2539	0.2677	0.2312	**0.6680**
American college football	M	0.6050	0.8907	0.5584	0.5733	0.5175	**1.0492**
Jazz musicians	M	0.0114	1.1424	0.1133	0.1118	**1.1438**	0.9357
Network science	M	0.7286	0.9558	0.6607	0.7413	0.5026	**1.6570**
Power grid	M	0.0439	0.3713	0.3417	0.3682	1.0097	**1.1051**
High-energy theory	M	0.5427	**0.9965**	0.5648	0.6004	0.9636	0.9725
DBLP	F_1	0.0967	0.0402	-	-	0.0390	**0.0985**
Amazon	F_1	0.0315	0.0070	-	-	**0.0441**	0.0419
YouTube	F_1	0.0445	-	-	-	0.0194	**0.0605**

Table 3.2: Experimental results in terms of modularity (M) and F_1 score (F_1).

For the choice of membership threshold β, we examine different values from 0.5 to 1 to find a reasonable range. It is clear that a community will contain fewer nodes if we set a higher value to β. According to our experiments, $[0.7, 0.8]$ appears to be a suitable range for candidates since a community may contain nearly half of the nodes when β is less than 0.7, while many nodes may not belong to any communities when β is larger than 0.8. To determine the final value of β, we again use the cross-validation paradigm with several candidates in this range and pick the one with the best performance on validation data.

3.4.5 Convergence Issues

Since our PNMF model applies stochastic gradient descent as the learning technique, we also observe convergence rate and convergence speed while conducting experiments. For convergence rate, as long as the learning rate and the regularization parameter are appropriate, all the datasets can converge before reaching a maximum number of iteration. For convergence speed, Figure 3.1 shows the results on five UMich datasets and Figure 3.2 shows the results of three SNAP datasets. Here the y-axis represents the ratio of current loss to initial loss. From both figures, we can see that loss drops quickly in the beginning and starts to slow down significantly after it reaches 20% of the initial loss. Comparing these two figures, we can also find that, although SNAP datasets need more time for one iteration than

UMich datasets, the total number of iteration is smaller, which proves the scalability of our model from another perspective.

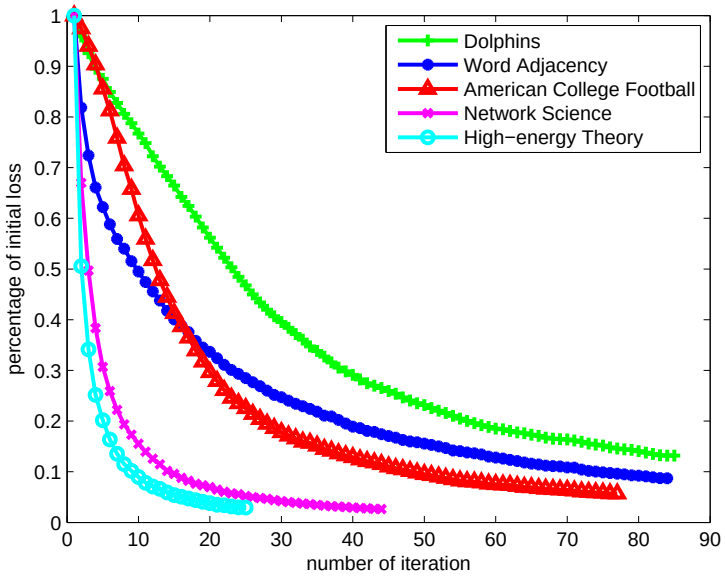


Figure 3.1: Convergence speed of learning algorithm on UMich datasets

3.5 Conclusion and Future Work

In this chapter, we have presented a *Preference-based Non-negative Matrix Factorization* model for overlapping community detection. The most significant contribution of our model is to incorporate implicit link preference information into the model formulation. By following the intuition that a node prefers any of its neighbors than any of its "non-neighbors", we maximize the likelihood of a preference order for each node instead of simply approximating the original adjacency matrix in value. Our model can eliminate the unreasonable indiscriminate penalty on pairs inside

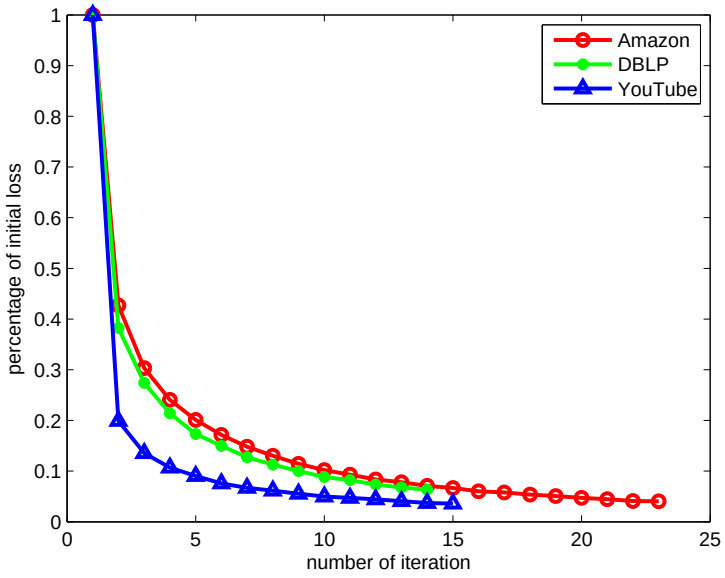


Figure 3.2: Convergence speed of learning algorithm on SNAP datasets

and between communities. In the learning process, we choose stochastic gradient descent with bootstrap sampling to learn the parameters of node-community membership matrix. We apply our PNMF model on several real-world datasets both with and without ground-truth communities. Our results show that our PNMF model outperforms state-of-art approaches in two metrics, namely modularity and F_1 score, and is scalable for large datasets.

Our current work only focuses on the difference between neighbors and "non-neighbors". We assume that all the "non-neighbors" have the same preference. However, this assumption may not hold in real-world networks. Considering two nodes A and B with no link between them, if there are other nodes which are

neighbors of both A and B, from the perspective of A it is reasonable to assign higher preference on B than nodes which have no common neighbors with A. We plan to employ the concept of common neighbors to enhance our preference system in our future work.

□ **End of chapter.**

Chapter 4

A Locality-based NMF Model

Community detection is of crucial importance in understanding structures of complex networks. In many real-world networks, communities naturally overlap since a node usually has multiple community memberships. One popular technique to cope with overlapping community detection is *Matrix Factorization (MF)*. However, existing MF-based models have ignored the fact that besides neighbors, "local non-neighbors" (e.g., my friend's friend but not my direct friend) are helpful when discovering communities. In this chapter, we propose a *Locality-based Non-negative Matrix Factorization (LNMF)* model to refine a preference-based model by incorporating locality into learning objective. We define a subgraph called "k-degree local network" to set a boundary between local non-neighbors and other non-neighbors. By discriminately treating these two class of non-neighbors, our model can capture the process of community formation. We propose a fast sampling strategy within

the stochastic gradient descent based learning algorithm. We compare our *LNMF* model with several baseline methods on various real-world networks, including large ones with ground-truth communities. Results show that our model outperforms state-of-the-art approaches.

4.1 Introduction

An individual in a social network can not only be regarded as an individual. One's behaviors are influenced by people around her, especially close friends. And her activities will influence others as well. A person always appears in a social network with multiple social identities, e.g., a (former) graduate student, a family member, a club member, a star fan, a company employee, etc. In most cases, her behaviors are related to one or several of these identities. Since identities can be defined by communities, discovering such overlapping communities in social networks becomes an important task for understanding social relationships and activities. This task is known as *overlapping community detection* [30, 33, 68].

Unlike classic community detection assuming that communities are mutually exclusive, overlapping community detection cannot be directly turned into the traditional graph clustering (i.e., node clustering) problem. Thus, many heuristic methods have been proposed in the past decade to deal with this task. Early approaches pay most of the attention to links. *Clique*

Percolation [74, 49] tries to find all k-cliques (a complete graph with k nodes) and combine those sharing $k-1$ nodes to be communities. *Link clustering* [1], on the other hand, cluster links instead of nodes and assign each node to all communities that its corresponding links belong to. Other recent works such as [18, 103] select some seed node and use links to expand communities. These methods aim to seek communities via links but do not address the issue that communities are the actual reason behind links (see Figure 4.1). Considering a user's ego network [64], i.e., a network of connections between her friends where communities are social circles categorized manually, the reason for two nodes to build a link is that they are in the same category. For example, the probability of one's college mates to be friends are usually much higher than that of one's random friends.

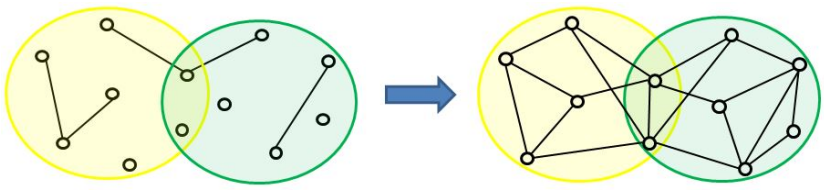


Figure 4.1: Community is the reason behind links.

Based on the idea that "communities generate links", *Matrix Factorization* based model has been employed for overlapping community detection. To apply this model, we need to set the number of communities and randomly assign users to each community in advance. Then a particular objective function

will be adopted to update the community membership for each node. Previously, a typical objective function is to minimize $||A - FF^T||$, where A is the adjacency matrix of the network and F is the node-community membership matrix. However, this value-approximation based objective function is problematic in that A only has 0 or 1 in its entry, which is more like a label (i.e., whether there is a link or not) than a real value. To tackle this issue, we have proposed a *Preference-based Non-negative Matrix Factorization (PNMF)* model in this thesis. Instead of approximating the value, it maintains a pairwise preference order for each node. To be specific, we assume that $A_{i,j} = 1$ and $A_{i,k} = 0$. Previous models try to make $F_i F_j^T$ close to 1 and $F_i F_k^T$ close to 0 while the preference based model only expects $F_i F_j^T$ to be larger than $F_i F_k^T$ without considering their actual values.

However, *PNMF* simply separates nodes into two parts, i.e., neighbors and non-neighbors, ignoring the fact that all non-neighbors are not supposed to be treated equally. Inspired by the famous saying "my friend's friend is also my friend", in this chapter, we propose a *Locality-based Non-negative Matrix Factorization (LNMF)* model to refine the *PNMF* model by further splitting the non-neighbors into two parts, namely "local non-neighbors" and "distant non-neighbors". We define a "k-degree local network" to distinguish these two kinds of non-neighbors. Given the two assumptions that (1) neighbors are preferred to local non-neighbors and (2) local non-neighbors are preferred to distant non-neighbors, we obtain the objective function by

maximizing a product of likelihood. We use the traditional stochastic gradient descent as our learning method and provide an efficient sampling strategy. Experiments conducted on real-world datasets show that our *LNMF* model does outperform the state-of-the-art approaches, indicating that our model assumption makes sense.

4.2 A Locality-based Non-negative Matrix Factorization (LNMF) Model

In this section, we first define the concept of k-degree locality and then formalize our LNMF model in the scenario of community detection. We will also briefly talk about the process of parameters learning and provide several candidates of the sampling strategy.

4.2.1 Preliminaries

Definition 4.1 (**k-Degree Local Network**). *Given an undirected and unweighted graph G, for a node $u \in G$, u's k-degree local network $L_k(u)$ is the subgraph consisting of all nodes whose shortest path length to u is less than or equal to k.*

According to the definition above, $L_0(u)$ consists of only node u, $L_1(u)$ is the subgraph including node u and all its neighbors, $L_\infty(u)$ is the whole graph, etc. We denote the node set of $L_t(u)$ except u itself as $V_t(u)$, where $t = 1, 2, \cdots$.

Now we further define the terms of "local non-neighbors" and "distant non-neighbors".

Definition 4.2 (**k-Degree Local Non-neighbors**). *Given a k-degree local network $L_k(u)$, the set of k-degree local non-neighbors $S_k(u)$ is defined as $S_k(u) := L_k(u) \backslash L_1(u)$, where $k \geq 1$.*

Definition 4.3 (**k-Degree Distant Non-neighbors**). *Given a k-degree local network $L_k(u)$, the set of k-degree distant non-neighbors $T_k(u)$ is defined as $T_k(u) := L_\infty(u) \backslash L_k(u)$, where $k \geq 1$.*

We can see that when $k = 1$, $S_k(u) = \emptyset$ and $T_k(u) = N^-(u)$. In this case, our model degrades to the *PNMF* model. When $k \geq 2$, our model will have a new class of nodes in preference system. Thus, our model is actually a generalization of the *PNMF* model.

Notation	**Meaning**
$G(V, E)$	Graph G with node set V and edge set E
$L_k(u)$	u's k-degree local network in G
$V_k(u)$	node set of $L_k(u)$ except u itself
$S_k(u)$	node set of u's k-degree local non-neighbors
$T_k(u)$	node set of u's k-degree distant non-neighbors
$N^+(u)$	node set of u's neighbors
$N^-(u)$	node set of u's non-neighbors

Table 4.1: A summary of notations.

A summary of notations is shown in Table 4.1. Four simple propositions can be drawn from the above notations.

Proposition 4.1. $V_k(u) = N^+(u) \bigcup S_k(u)$.

Proposition 4.2. $N^+(u) \bigcap S_k(u) = \emptyset$.

Proposition 4.3. $N^-(u) = S_k(u) \bigcup T_k(u)$.

Proposition 4.4. $S_k(u) \bigcap T_k(u) = \emptyset$.

4.2.2 Model Assumption

Recall the basic assumption of *PNMF* in Equation **??**. Incorporating the concept of k-degree local network, we can exploit k-degree local non-neighbors to enhance the old model assumption. The new model assumption for our *LNMF* model can be represented as

$$r_{u,i} \geq r_{u,j}, r_{u,j} \geq r_{u,d}, i \in N^+(u), j \in S_k(u), d \in T_k(u), \quad (4.1)$$

where $r_{u,p}$ is still the preference of node u on node p. It means (1) neighbors are preferred to local non-neighbors; (2) local non-neighbors are preferred to distant non-neighbors. These two assumptions are quite intuitive. Notice that when $k = 1$, the new model assumption degrades to the old one.

We also adopt two independence assumptions of our *PNMF* model, i.e., node independence and pair independence assumptions, to formalize our new model.

- **Node independence**. The preference order of each node is independent with that of any other node. There will be a link between u and v if and only if u prefers to build a

relationship with v and symmetrically v prefers to build a relationship with u.

- **Pair independence**. For a fixed node i, its preference on j and k is independent with its preference on u and v when $j, u \in N^+(i)$ and $k, v \in N^-(i)$.

4.2.3 Model Formulation

Given the above model assumptions, we are ready to present our *LNMF* model formally. Since nodes are independent of each other, we can consider one node at first.

For each node u, the optimization criterion is to maximize the likelihood of preference order which can be represented as a product of pairwise preferences, i.e.,

$$\begin{aligned} \prod_{i,j \in V_k(u)} & [\mathcal{P}(r_{u,i} \geq r_{u,j}|F)^{\delta(u,i,j)} \\ & (1 - \mathcal{P}(r_{u,i} \geq r_{u,j}|F))^{1-\delta(u,i,j)}] \cdot \\ \prod_{j,d \in N^-(u)} & [\mathcal{P}(r_{u,j} \geq r_{u,d}|F)^{\xi(u,j,d)} \\ & (1 - \mathcal{P}(r_{u,j} \geq r_{u,d}|F))^{1-\xi(u,j,d)}], \end{aligned} \tag{4.2}$$

where $\delta(\cdot)$ and $\xi(\cdot)$ are two indicator functions that

$$\delta(u,i,j) = \begin{cases} 1 & \text{if } i \in N^+(u) \text{ and } j \in S_k(u), \\ 0 & \text{otherwise} \end{cases}$$

and

$$\xi(u,j,d) = \begin{cases} 1 & \text{if } j \in S_k(u) \text{ and } d \in T_k(u), \\ 0 & \text{otherwise} \end{cases}.$$

Recall the four propositions in preliminaries that $V_k(u)$ and $N^-(u)$ can be split into two disjoint sets with different levels of preference. Following the scheme argued in [88, 114], we can simplify Equation 4.2 to

$$\frac{\sum_{i\in N^+(u), j\in S_k(u)} \mathcal{P}(r_{u,i} \geq r_{u,j}|F)}{|N^+(u)| \cdot |S_k(u)|} + \frac{\sum_{j\in S_k(u), d\in T_k(u)} \mathcal{P}(r_{u,j} \geq r_{u,d}|F)}{|S_k(u)| \cdot |T_k(u)|}. \tag{4.3}$$

Applying the sigmoid function $\sigma(x) := \frac{1}{1+e^{-x}}$ to interpret $\mathcal{P}(r_{u,i} \geq r_{u,j}|F)$, i.e., $\mathcal{P}(r_{u,i} \geq r_{u,j}|F) = \sigma(\hat{x}(u,i) - \hat{x}(u,j))$, we sum up the log-likelihood functions of all nodes:

$$\sum_u [\sum_{i\in N^+(u), j\in S_k(u)} \ln \sigma(\hat{x}(u,i) - \hat{x}(u,j)) + \lambda(u) \cdot \sum_{j\in S_k(u), d\in T_k(u)} \ln \sigma(\hat{x}(u,j) - \hat{x}(u,d))], \tag{4.4}$$

where $\hat{x}(u,i) := F_u \cdot F_i^T$ can be regarded as the correlation between u and i, and $\lambda(u) := \frac{|N^+(u)|}{|T_k(u)|}$ can be regarded a coefficient of local influence.

In the end, to prevent our model from overfitting, we add a regularization term $reg(F) = ||F||_F^2$, which is the Frobenius norm of the node-community membership matrix. The final objective function l is

$$l(F) = \sum_u [\sum_{i\in N^+(u), j\in S_k(u)} \ln \sigma(\hat{x}(u,i) - \hat{x}(u,j)) + \lambda(u) \cdot \sum_{j\in S_k(u), d\in T_k(u)} \ln \sigma(\hat{x}(u,j) - \hat{x}(u,d))] - \lambda_r reg(F), \tag{4.5}$$

where λ_r is a regularization coefficient.

4.2.4 Parameter Learning

As an efficient and widely-used paradigm for parameter learning, *stochastic gradient descent (SGD)* is employed as our learning algorithm. Distinguished from the traditional batch gradient descent which computes Equation 4.5 in each iteration, *SGD* only picks a small number of random samples to perform update. In our case, a sample is a (source, neighbor, local non-neighbor, distant non-neighbor) quadruple. Mathematically, we calculate the derivative of our final objective function l by

$$\Theta^{t+1} = \Theta^t + \alpha \frac{\partial l}{\partial \Theta}, \tag{4.6}$$

where Θ can be any entry of the node-community membership matrix F. For the non-negative constraints, we apply a projected gradient method proposed in [58], which maps the parameter vector back to the nearest point in projected space, in our case, the non-negative space.

The whole process is described in Algorithm 2. Let sample size be t. The time complexity of each iteration is $O(tp)$ and the space complexity is $O(np)$, where n is the number of nodes and p is the number of communities.

4.2.5 Sampling Strategy and Other Issues

Due to the nature of stochastic gradient descent, sampling strategy matters to both running time and performance. More than what *PNMF* did, we need to sample a set of quadruples for each

Input: G, the adjacency matrix of original graph
Output: F, the node-community membership matrix
initialize F
compute initial loss
repeat
  for *num_samples* $= 1$ to *sample_size* **do**
    sample (u, i, j, d) according to Algorithm 3
    for each entry Θ in F_u, F_i, F_j and F_d **do**
      update Θ according to Equation (4.6)
      $\Theta \leftarrow \max(\Theta, 0)$
    end for
  end for
  compute loss
until convergence or *max_iter* is reached

Algorithm 2: Community Detection via *LNMF*

learning step. The process is described in Algorithm 3.

Input: G, the adjacency matrix of original graph
Output: (u, i, j, d), a quadruple to perform a step in stochastic gradient descent
sample node u from V uniformly at random
sample node i from $N^+(u)$ uniformly at random
sample node j from $S_k(u)$ uniformly at random
sample node d from $T_k(u)$ uniformly at random

Algorithm 3: Sampling Strategy

For the sampling of j, we need to pre-process the whole graph to record a set of local nodes of each u in the graph. By using the fact that $N^-(u) = S_k(u) \bigcup T_k(u)$, we keep sampling a random node until we get a node neither in $N^+(u)$ nor in $S_k(u)$ and let

d be this node.

Moreover, there are several remaining issues to be discussed.

- **The number of communities.** The nature of matrix factorization needs us to set the number of communities which are unknown in advance. A cross-validation paradigm is used. In detail, we reserve 10% of nodes as the validation set at first. After learning the node-community membership matrix F, we compute the sum of log-likelihood function for all nodes in the validation set via Equation 4.3. Since the computational cost is enormous for cross-validation, only a small set of quadruple will be sampled.

- **The community membership threshold.** Obtaining the node-community membership matrix F is still one step away from getting the final node-community correspondence. We need to set a threshold to decide whether a community accepts a node. Similar to what we employ in the *PNMF* model, we set a probability threshold to $\mathcal{P}(r_{u,i} \geq r_{u,j}|F)$ and use the sigmoid function to reversely compute the lower bound of community membership weight assuming that u and i share one community but u and j do not share any community.

- **The convergence criterion.** First, we randomly generate a subset of quadruples to be our loss sample and compute initial loss on this set according to Equation 4.5. After each iteration, we compute loss again and stop when the absolute

difference between the current loss and the previous loss is smaller than a small percentage, say ϵ, of the initial loss.

4.3 Experiments

In this section, we compare our $LNMF$ model with both classic and state-of-the-art overlapping community detection methods on various real-world datasets. We will show our experimental results with two metrics, namely modularity and F_1 score, and have a brief discussion.

4.3.1 Data Description

Six benchmark networks collected by Newman[1] are used as our datasets. These networks are relatively small and have no ground-truth communities. Basic information of these datasets can be found in Table 4.2, where **V** is the number of nodes and **E** is the number of links.

Dataset	**V**	**E**
Dolphins	62	159
Les Misérables	77	254
Books about US politics	105	441
Word adjacencies	112	425
American college football	115	613
Coauthorship in network science	1,589	2,742

Table 4.2: Statistics of six Newman's datasets.

[1]http://www-personal.umich.edu/ mejn/netdata/

Moreover, we choose three large networks with ground-truth communities collected by SNAP[2] [108] to test the scalability of our model. These networks are of different types:

- **YouTube** dataset: a social network of a video-sharing web site.
- **DBLP** dataset: a collaboration network of research paper authors in computer science.
- **Amazon** dataset: a products co-purchasing network based on Customers Who Bought This Item Also Bought feature of the Amazon website.

Simple statistics for these three datasets are shown in Table 4.3, where **V** is the number of nodes, **E** is the number of links, **C** is the number of ground-truth communities, and **U** is the average number of nodes per community.

Dataset	**V**	**E**	**C**	**U**
DBLP	317k	1.0M	2.5k	429.8
Amazon	335k	926k	49k	100.0
YouTube	1.1M	3.0M	30k	9.7

Table 4.3: Statistics of three SNAP datasets.

4.3.2 Experimental Setup

We conduct our experiments on a computer with a Xeon 2.60GHz CPU and 64GB memory.

[2]http://snap.stanford.edu/data/

Comparison methods. We select both classic and state-of-the-art methods to compare with our model. The latter four are *Non-negative Matrix Factorization (NMF)* based models.

- **SCP** [49] accelerates the original **CP** method [74] in a sequential manner. We set k to be 4 or 5 when finding k-cliques.
- **LC** [1] clusters link instead of node to get overlapping communities. We ignore all communities with only one or two nodes since they are meaningless.
- **BNMF** [80] is one of the earliest work which applies *MF* into community detection. The squared loss is used as loss function.
- **BNMTF** [113] incorporates a community interaction matrix into the classic *MF* to become a *Matrix Tri-Factorization* model. Squared loss is used as loss function.
- **BigCLAM** [109] is claimed by its authors as a scalable model. It can search for the best number of communities given a range.
- **PNMF** is the model we propose in Chapter 3.

Evaluation metrics.

- **Modularity**. We use the classic modularity as our metric

for Newman's datasets. Modularity Q is defined as

$$Q = \frac{1}{2m} \sum_{u,v \in V} (A_{u,v} - \frac{d(u)d(v)}{2m}) |C_u \cup C_v|,$$

where m is the number of links, V is the node set, A is the adjacency matrix, $d(u)$ is the degree of node u, and C_u is the set of communities to which node u belongs. This definition indicates that for each node pair (u, v) which shares communities, its contribution to modularity is positive if u, v are linked and is negative otherwise. It matches our intuition that nodes inside one community tends to build links with each other.

- **$\mathbf{F_1}$ score**. For SNAP datasets with ground-truth communities, F_1 score is obviously one of the best measurements. The F_1 score of a detected community S_i is defined as the harmonic mean of precision(S_i) and recall(S_i), where precision(S_i) and recall(S_i) are defined as

$$\text{precision}(S_i) = \max_j \frac{|C_j \bigcap S_i|}{|C_j|},$$

and

$$\text{recall}(S_i) = \max_j \frac{|C_j \bigcap S_i|}{|S_i|},$$

where C_j is the node set of a ground-truth community. The average F_1 score for the set of detected communities S is

$$\overline{F_1}(S) = \frac{1}{|S|} \sum_{S_i \in S} F(S_i).$$

Setting the k. Remember that if we set $k = 1$ in k-degree local network, our model will degrade to the *PNMF* model. According to our observation on several datasets, if k is set to be larger than 2, the average number of common communities two nodes in a k-degree local network share is not significantly larger than that two random nodes in the whole network share. Thus, we set k to be 2, which means only a friend's friends are considered as local non-neighbors.

4.3.3 Results

We set the regularization coefficient to be 0.5 and the convergence parameter ϵ to be 0.001 for all experiments. The sample size t is determined according to data size. For Newman's datasets, we set $t = m$, i.e., the number of links. For SNAP datasets, we set $t = 10\sqrt{n}$ to finish one iteration without taking too much time, where n is the number of nodes. The maximum times of iteration are set to 100, though in fact, all datasets converge before reaching the limit.

Table 4.4 shows the performance of our *LNMF* model on Newman's datasets, where **RI** denotes the relative improvement over PNMF. From the results, we find that under the metric of modularity, our *LNMF* model outperforms all baseline methods on all datasets.

Table 4.5 shows the our experimental results on SNAP datasets, where **RI** denotes the relative improvement over PNMF.

Dataset	**SCP**	**LC**	**BNMF**	**BNMTF**	**BigCLAM**	**PNMF**	**LNMF(RI)**
Dolphins	0.305	0.654	0.507	0.507	0.423	0.979	**1.086(10.9%)**
Les Misérables	0.307	0.773	0.125	0.103	0.540	1.103	**1.184(7.3%)**
Books about US politics	0.496	0.851	0.461	0.492	0.529	0.864	**1.270(47.0%)**
Word adjacencies	0.071	0.271	0.254	0.268	0.231	0.668	**0.701(4.9%)**
American College football	0.605	0.891	0.558	0.573	0.518	1.049	**1.235(17.7%)**
Coauthorships in network science	0.729	0.956	0.661	0.741	0.503	1.657	**2.310(39.4%)**

Table 4.4: Comparison in terms of modularity.

Dataset	BigCLAM	PNMF	LNMF(RI)
DBLP	0.039	0.098	**0.107(9.2%)**
Amazon	0.044	0.042	**0.048(11.4%)**
YouTube	0.019	**0.060**	0.057(0.0%)

Table 4.5: Experimental results on SNAP datasets in terms of F_1 score.

The other baselines methods are not listed here since none of them can finish all three datasets in time. This fact can reflect the scalability of our *LNMF* model to some extent. It can be seen that our model outperforms *BigCLAM* on all datasets and has an improvement over $PNMF$ on two of three datasets. For *YouTube*, we find its community formation pattern quite random due to the small size of communities and the large variety of users. In other words, our model assumption does not fit the community pattern of this dataset so well, which explains why *LNMF* fails to improve on it. The running time of one iteration is about one or two hours for *DBLP* and *Amazon*. For *YouTube*, it takes about four to five hours to finish an iteration.

The convergence speed of our learning algorithm on *Amazon* and *DBLP* is illustrated in Figure 4.2. A point in the figure represents the ratio of current loss to initial loss after i-th iteration. The results show that our *LNMF* can converge within a fair number of iterations. In fact, if we do not consider the regularization term, the final losses of both datasets are less than 10% of the initial loss.

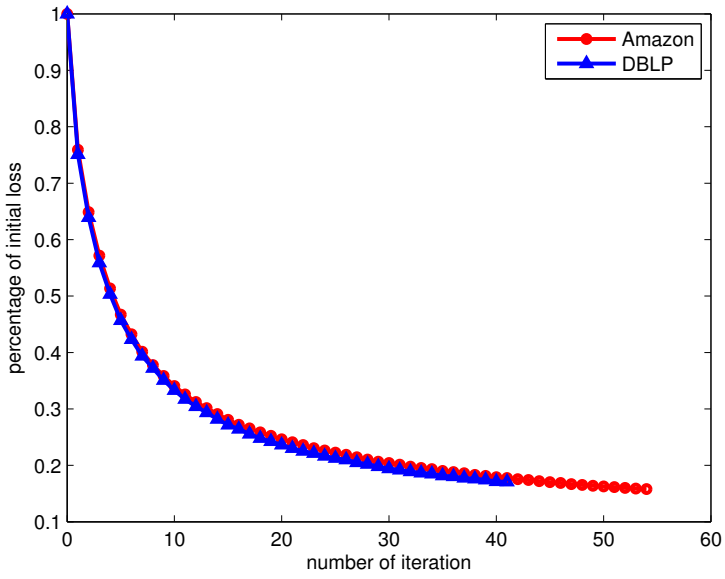


Figure 4.2: Convergence speed of learning algorithm.

4.4 Conclusion

In this chapter, we propose a *Locality-based Non-negative Matrix Factorization* model to improve the performance of existing work on overlapping community detection. Our *LNMF* model is based on a pairwise preference learning scheme. We exploit local area around a node formally defined as a k-degree local network to enhance the previous preference system. In detail, we extend a two-level preference system which only distinguishes neighbors and non-neighbors to a three-level preference system which split the set of non-neighbors into local non-neighbors and distant non-neighbors. Experiments on several real-world datasets including large ones with ground-truth communities show that this extension can indeed improve the quality of overlap-

ping community detection.

□ **End of chapter.**

Chapter 5

A Mutual Density-based NMF Model

Community detection provides a way to unravel complicated structures in complex networks. Overlapping community detection allows nodes to be associated with multiple communities. *Matrix Factorization (MF)* is one of the standard tools to solve overlapping community detection problems from a global view. Existing MF-based methods only exploit link information revealed by the adjacency matrix, but ignore other critical information. In fact, compared with the existence of a link, the number of mutual friends between two nodes can better reflect their similarity regarding community membership. In this chapter, based on the concept of mutual friend, we introduce *Mutual Density* as a new indicator to infer the similarity of community membership between two nodes in the MF framework for overlapping community detection. We conduct data observation on real-world networks with ground-truth communities to validate

an intuition that mutual density between two nodes is correlated with their community membership cosine similarity. According to this observation, we propose a *Mutual Density-based Non-negative Matrix Factorization (MD-NMF)* model by maximizing the likelihood that node pairs with larger mutual density are more similar in community memberships. Our model employs stochastic gradient descent with sampling as the learning algorithm. We conduct experiments on various real-world networks and compare our model with other baseline methods. The results show that our MD-NMF model outperforms the other state-of-the-art models on multiple metrics in these benchmark datasets.

5.1 Introduction

In complex networks, there usually exist groups inside which nodes are connected more densely with one another than with the nodes outside. These groups of nodes are called *communities* [35]. In reality, these groups usually have physical meanings such as members of the same organization, scientists with publications in the same area, or proteins sharing the same function [47]. Thus, uncovering such latent communities in complex networks has attracted great research interests in the past decade [30]. Classic methods assume communities are mutual exclusive, i.e., each node of a network belongs to one and only one community. However, in real-world complex networks like social

networks and biological networks, such community membership restriction does not apply because a node may have multiple characteristics and thus belongs to multiple communities. As a result, a more challenging problem named *overlapping community detection* has been introduced in recent years [105].

Matrix Factorization (MF), as one of the standard frameworks to solve the problem of overlapping community detection, detects communities from a global view [105]. Taking the adjacency matrix G of the given network as input, MF-based models assign the number of communities in advance and seek out a node-community weight matrix F, which matches the information revealed by the input as accurately as possible. Early work [80, 101] simply aims to approximate G entry by entry with FF^T, which only makes use of the mathematical representation of adjacency matrix, but ignores its physical meaning. The most obvious information an adjacency matrix provides is the link information. Thus, recent work [107] assumes that nodes sharing more communities have a higher probability to be linked and formulates the problem with a generative objective function. In other words, a link can be regarded as an indicator to reflect the similarity of community membership between two nodes.

However, a link is not a perfect indicator for two major reasons. First, it is common that two nodes sharing several communities do not have a link between them, or two nodes with no common community are connected. A survey conducted on Facebook [21] shows that edges between two individuals from diffe-

rent communities outnumber edges connecting users in the same community. For example, a salesperson may make connections with many strangers to sell his products, and the establishment of links between salespeople and customers does not indicate any similarity between their community memberships. In cases like these, links become noise instead of evidence. Second, a link is a binary indicator in an unweighted network. Given two linked node pairs with no other information at all, it is impossible to distinguish which one is more similar.

Inspired by the definition of tie strength [34], we introduce a more powerful indicator, which is the number of mutual friends between two nodes, to reflect their community membership similarity. The definition of tie strength reveals that the stronger tie the two nodes own, the larger overlap in their friendship circles they will have. This idea can be incorporated into our matrix factorization framework for overlapping community detection, which meets the common sense that the more communities two nodes share, the more mutual friends they will have. For example, if two individuals attended the same class in high school, joined the same basketball team, and work in the same company now, they should know many mutual friends in different communities, i.e., their ego-networks (friend circles) are densely overlapped. Compared to a link, the number of mutual friends is no longer a binary indicator and it provides more confidence to predict the similarity of community membership between two nodes. However, it still suffers from several issues: the lack of

friends of two nodes may limit the number of mutual friends between them, and communities with different sizes may contribute different numbers of mutual friends to each node pair. To handle these limitations, we introduce *Mutual Density* as a more consistent indicator, which is defined as the Jaccard similarity of two nodes' ego-networks. Under the general description of "neighborhood similarity", the concept of mutual density has been applied in community detection under different assumptions [1, 4, 67, 96, 99]. However, none of these methods are based on matrix factorization, and none of them use mutual density to measure the similarity of community membership between two nodes.

In this chapter, we introduce mutual density and the number of mutual friends as the new indicators instead of links themselves for inferring community membership similarity in the matrix factorization framework. We conduct data observation on real-world networks with ground-truth communities to validate that mutual density is more consistent with community memberships similarity than the other two indicators. Thus, we formulate our *Mutual Density-based Non-negative Matrix Factorization (MD-NMF)* model, which incorporates mutual density as the community similarity indicator and employs a novel objective function to ensure that a node pair with higher mutual density is more likely to have a higher community membership similarity. From a node's perspective, we ensure that it is more likely to join the same communities with its acquaintances than with its stran-

gers. To solve the optimization problem, we apply projected stochastic gradient descent with sampling. By using our model to real-world and open-source network datasets, we find that our new MD-NMF model outperforms several state-of-the-art methods on either modularity or F_1 score.

The main contributions of this chapter are:

1. We introduce *Mutual Density* as a new indicator to reflect the community membership similarity between two nodes in substitution for a link within the matrix factorization framework for overlapping community detection.

2. We find that there is consistency between the mutual density of two nodes and their community memberships similarity by empirically studying real-world networks with ground-truth communities.

3. We propose a novel *Mutual Density-based Non-negative Matrix Factorization (MD-NMF)* model for overlapping community detection by formulating mutual density properly in the matrix factorization framework. Our model outperforms state-of-the-art baselines.

In the rest of this chapter, we first list out indicator definitions and show our data observations in Section 5.2. Then we formulate our MD-NMF model and discuss parameter learning in Section 5.4. Experimental results are illustrated in Section 5.5, followed by related work in Section 5.3 and conclu-

sion in Section 5.6.

5.2 Definition and Data Observation

In this section, we first define the community detection problem. Then we define three indicators mentioned in Introduction for indicating community membership similarity. Finally, we conduct data observation experiments on two real-world networks with ground truth communities and examine which indicator has the best consistency.

5.2.1 Indicator Definitions

To infer the community membership similarity between two nodes, we have mentioned three indicators in Introduction. They are link existence $l(u, v)$, the number of mutual friends $m(u, v)$ and mutual density $d(u, v)$, where u and v are both nodes in V. We formally define each of them as follows.

Definition 5.1 (Link Existence)**.** Given a graph $G(V, E)$ and two nodes $u, v \in V$, the link existence between u and v is

$$l(u, v) = \begin{cases} 1 & \text{if } G_{uv} = 1, \\ 0 & \text{else} \end{cases} . \tag{5.1}$$

Definition 5.2 (The Number of Mutual Friends)**.** Given a graph $G(V, E)$ and two nodes $u, v \in V$, the number of mutual friends between u and v is

$$m(u, v) = \mid \{i | (u, i) \in E \; and \; (v, i) \in E\} \mid. \tag{5.2}$$

Definition 5.3 (Mutual Density)**.** Given a graph $G(V, E)$ and two nodes $u, v \in V$, the mutual density between u and v is

$$d(u, v) = \frac{\mid \{i | (u, i) \in E \text{ and } (v, i) \in E\} \mid}{\mid \{j | (u, j) \in E \text{ or } (v, j) \in E\} \mid}. \tag{5.3}$$

5.2.2 Data Observation

To validate (1) the number of mutual friends is better than a link in inferring community membership similarity, and (2) mutual density is more stable compared with the number of mutual friends, we conduct two experiments on two large real-world networks with ground-truth communities [108]. Table 5.1 shows the statistics of these two networks, where $|V|$ is the number of nodes, $|E|$ is the number of edges, $|C|$ is the number of ground-truth communities, D is the average degree of nodes, M is the average number of nodes per community, and A is the average number of joined communities per node.[1]

Dataset	**|V|**	**|E|**	**|C|**	**D**	**M**	**A**
Amazon	335k	926k	49k	3.38	100.0	14.83
DBLP	317k	1.0M	2.5k	4.93	429.8	2.57

Table 5.1: Dataset statistics.

To quantify the community membership similarity between two nodes, we use *cosine similarity* as our measurement, which is defined as follows.

[1] http://snap.stanford.edu/data/

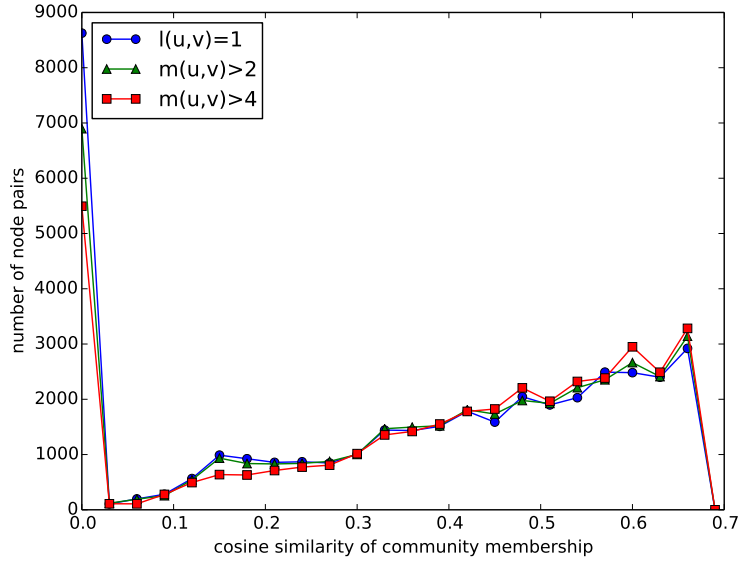
Definition 5.4 (Cosine similarity of community membership)**.** Given a graph with p ground-truth communities $\{C_i | i = 1, 2, \cdots, p\}$, the cosine similarity of community membership $s(u, v)$ between u and v is

$$s(u, v) = \frac{\vec{u} \cdot \vec{v}^T}{\|\vec{u}\|_2 \|\vec{v}\|_2} , \tag{5.4}$$

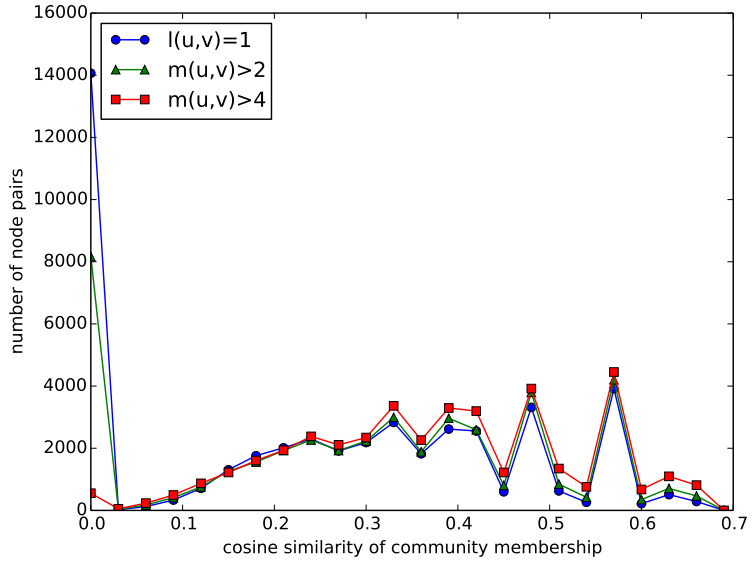
where $\vec{u} \in \mathcal{R}^p$ is the community membership vector of node u and u_i represents the weight u belongs to community C_i.

First, we randomly sample 100,000 node pairs with links as well as 100,000 node pairs with at least two or four mutual friends and compute the cosine similarity of community membership for each node pair. Figure 5.1 plots the number of 3 different types of node pairs with the same value of cosine similarity. We expect all three types of node pairs to share at least one community and thus to have non-zero cosine similarity. However, nearly 14,000 node pairs with links do not share any communities. The error rate is about 14%. On the other side, less than 8% of the node pairs with at least two mutual friends and only about 1% of the node pairs with at least four mutual friends are out of our expectation. When the value of cosine similarity is nonzero, all three types are pretty similar, and the number of node pairs with four mutual friends is slightly greater than the other types.

In contrast, we randomly draw 50,000 non-linked node pairs and the same number of node pairs that have no mutual friends. We expect both sets of node pairs to have $s(u, v) = 0$. Table 5.2



(a) Amazon



(b) DBLP

Figure 5.1: The number of sampled node pairs having a same value of cosine similarity

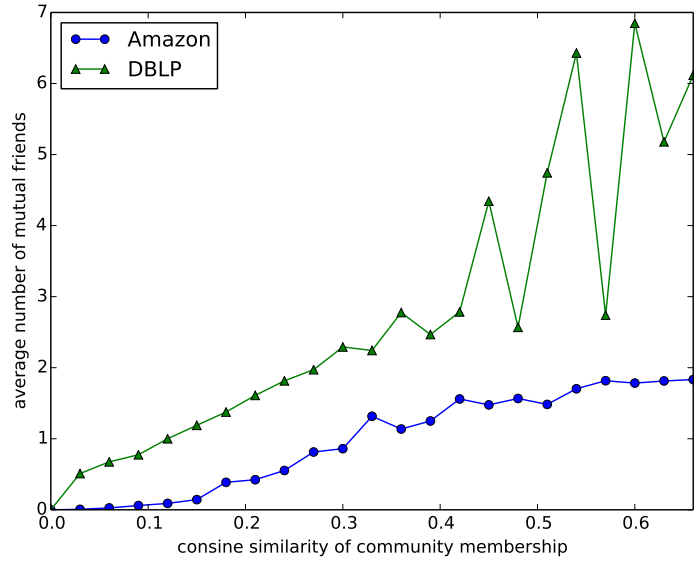
Indicator	$s(u,v) = 0$	$s(u,v) > 0$
l(u,v) = 0	45793	4207
m(u,v) = 0	49816	184

Table 5.2: Comparison of error rate for 50,000 non-linked node pairs between the number of mutual friends and the existence of links.

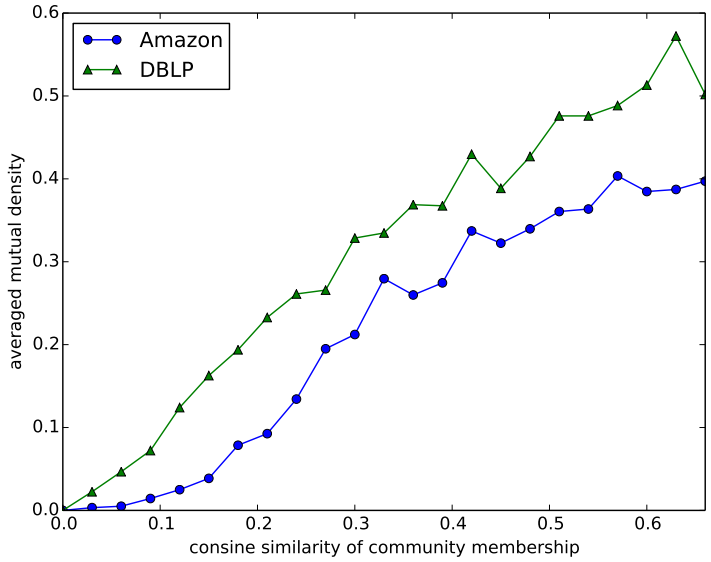
shows the number of node pairs with $s(u,v) = 0$ or $s(u,v) > 0$. We can see that the number of mutual friends has a smaller error rate than the existence of links. Combining the observations from both Figure 5.1 and Table 5.2, we can conclude in strong confidence that the number of mutual friends is a more accurate and more flexible indicator compared to the existence of links.

Second, we compare the stability of indicator between the number of mutual friends and mutual density. A stable indicator is expected to be monotonic while community membership similarity increases. We sample 10,000 node pairs each time with a certain value of cosine similarity and calculate the average number of mutual friends and average mutual density of these node pairs. The result is shown in Figure 5.2. We can see that on the DBLP data, the average number of mutual friends vibrates up and down while average mutual density is almost monotonic as cosine similarity increases. Thus, mutual density is a more stable indicator than the number of mutual friends to infer community membership similarity.

In summary, mutual density is the best indicator among all three indicators we mentioned with highest accuracy and stabi-



(a) the number of mutual friends



(b) mutual density

Figure 5.2: Averaged value of each indicator as a function of cosine similarity in community membership

lity.

5.3 Related Work

In this section, we will review several works related to this chapter. We first investigate the applications of mutual friends in community detection. Then, we introduce the background of the learning objective of our MD-NMF model.

5.3.1 Mutual Friends

Mutual friend as a strong factor to indicate the closeness between two nodes has been investigated in many social-related tasks. Friend recommender systems provide the potential friends list through discovering the latent information behind network topology and friends in common [5, 94]. Link prediction models in complex networks use common neighbors to evaluate the probabilities of link establishments [6, 61]. Online social rating networks make use of the co-commenting and co-rating behaviors of users to recommend products and predict new rating [97]. In community detection problem, mutual friends have also been employed to measure the strength of connections between nodes. Newman defines connection strength as the normalized term of mutual friends and uses it to cluster nodes [67]. Tang and Liu directly interpret Jaccard similarity as node similarity to fit into K-means algorithm for community detection [99]. Steinhaeuser and Chawla exam Jaccard coefficient as an

edge weighting method and employ it in community detection. However, this algorithm fails to detect any community structure without the addition of node attribute [96]. Alvari et al. regard neighborhood similarity, i.e., the number of common neighbors, as a similarity measure and incorporate it into a game theory framework [4]. Ahn et al. explicitly define link similarity and hierarchically cluster links accordingly [1].

In this chapter, mutual density has the same mathematical form as Jaccard similarity or link similarity but is used for measuring the community membership similarity. Thus, we can still calculate mutual density between two nodes even if they are not linked. Also, our model is built on the matrix factorization framework instead of link clustering.

5.3.2 Bayesian Personalized Ranking

The pairwise objective function of our model is based on the Bayesian Personalized Ranking [88]. This method and its extensions are originally proposed to solve the ranking problem in recommender systems [75, 87, 114]. Also, in the *PNMF* model in Chapter 3 and the *LNMF* model in Chapter 4, we employ *BPR* on the overlapping community detection problem. We focus on the link indicator and assume that each node shares more common communities with its neighbors than its non-neighbors, which is more realistic both conceptually and experimentally.

5.4 Mutual Density-based NMF Model

In this section, we formally define our model assumption and formulate our pairwise learning objective in a matrix factorization framework. We apply stochastic gradient descent with sampling to learn model parameters.

5.4.1 Model Assumption

From the data observation, we can see that the cosine similarity of community membership between two nodes is correlated with their mutual density. It leads to the intuition of our model that two nodes with larger mutual density are more likely to have higher cosine similarity of community membership.

To formally illustrate our model assumption, we need to define two relationships between two nodes in the first place: α-acquaintance and β-stranger.

Definition 5.5 (α-acquaintance)**.** Given $\alpha \in [0, 1]$, for two nodes $u, v \in V$, v is u's α-acquaintance if and only if

$$d(u, v) \geq \alpha.$$

By the symmetry of $d(u, v)$, u is also v's α-acquaintance.

Definition 5.6 (β-stranger)**.** Given $\beta \in [0, 1]$, for two nodes $u, v \in V$, v is u's β-stranger if and only if

$$d(u, v) \leq \beta.$$

By the symmetry of $d(u, v)$, u is also v's β-stranger.

In both definitions, $d(u, v)$ is the mutual density between u and v defined in Equation (5.4). Moreover, for a node u, we define its set of α-acquaintances as $A(u, \alpha) = \{i | d(u, i) \geq \alpha\}$ and its set of β-strangers as $B(u, \beta) = \{j | d(u, j) \leq \beta\}$.

Following our intuition, our model assumption can be formally defined as

$$\begin{aligned} &s(u, i) > s(u, j), \\ &\text{if } i \in A(u, \alpha), j \in B(u, \beta), \text{and } \alpha > \beta, \end{aligned} \tag{5.5}$$

where $s(u, i)$ is the cosine similarity of community memberships between u and i.

In other words, we expect that the cosine similarity between u and any of its α-acquaintances should be greater than the cosine similarity between u and any of its β-strangers. Adjusting α and β for different graphs enables us to make sure that the difference of cosine similarity is significant. If α is only slightly greater than β, we are not confident enough to make such assumption.

5.4.2 Model Formulation

In the MD-NMF model, we aim to find the node-community weight matrix F which maximizes the likelihood that every node in the graph has higher cosine similarity in community membership with all its α-acquaintances than with all its β-strangers. For each node u, we want to maximize

$$\mathcal{P}(\underset{u}{>} | F, \alpha, \beta) = \prod_{i \in A(u,\alpha)} \prod_{j \in B(u,\beta)} \mathcal{P}(s(u, i) > s(u, j) | F). \tag{5.6}$$

Given any two nodes $u, v \in V$, we can obtain their node-community weight vectors F_u, F_v from F. From the observation that the higher cosine similarity of community membership vectors between two nodes, the greater mutual density they will have, we define the probability that $s(u,i) > s(u,j)$ given the node-community membership matrix as

$$\mathcal{P}(s(u,i) > s(u,j)|F) = \sigma(\frac{F_u F_i^T}{\|F_u\|_2 \|F_i\|_2} - \frac{F_u F_j^T}{\|F_u\|_2 \|F_j\|_2}), \quad (5.7)$$

where σ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$. For simplicity, we define $\phi(i,j) = \frac{F_i F_j^T}{\|F_i\| \|F_j\|}$, so we have

$$\mathcal{P}(s(u,i) > s(u,j)|F) = \sigma(\phi(u,i) - \phi(u,j)). \quad (5.8)$$

Since the sigmoid function maps any real value into $(0,1)$, this probability approaches to 1 when $\phi(u,i) \gg \phi(u,j)$ and approaches to 0 when $\phi(u,i) \ll \phi(u,j)$.

By multiplying Equation (5.6) for each node and combining Equation (5.7) and (5.8), we can derive the final learning objective of the MD-NMF model, which is

$$\begin{aligned} l(F) &= \max_{F \in R_+^{n \times p}} \log \prod_{u \in V} \mathcal{P}(\underset{u}{>} |F, \alpha, \beta) - \lambda \cdot reg(F) \\ &= \max_{F \in R_+^{n \times p}} \sum_{u \in V} \sum_{i \in A(u,\alpha)} \sum_{j \in B(u,\beta)} \log \mathcal{P}(s(u,i) > s(u,j)|F) \\ &\quad - \lambda \cdot reg(F) \\ &= \max_{F \in R_+^{n \times p}} \sum_{u \in V} \sum_{i \in A(u,\alpha)} \sum_{j \in B(u,\beta)} \log \sigma(\phi(u,i) - \phi(u,j)) \\ &\quad - \lambda \cdot reg(F), \end{aligned} \quad (5.9)$$

where $reg(F)$ is a regularization term in order to prevent overfitting of F, and λ is the regularization parameter. For the simplicity of differentiation, we set $reg(F) = \|F\|_F^2$, which is the Frobenius norm of F.

In Equation (5.7), maximizing the difference of cosine similarity gives the model a geometrical meaning. If we set the number of communities as p, then each row vector F_i is a node-community weight vector in R^p. When optimizing the learning objective, the cosine similarity between $\alpha-$acquaintances is maximized, which means angles between their node-community weight vectors will be narrowed; likewise, angles between the weight vectors of β-acquaintances will be enlarged. In this way, F_i will finally converge to the community membership vector of node i.

5.4.3 Parameter Learning

To make our model scalable to large datasets, we employ the widely used paradigm of *Stochastic Gradient Descent (SGD)* as our learning algorithm. Also considering the non-negativity constraint, we apply a projected gradient method [58] which maps the vector with negative parameters back to the nearest point in the projected space. Following the learning objective l, we update the matrix F by

$$\Theta_{t+1} = \max\{\Theta_t + \delta\frac{\partial l}{\partial \Theta}, 0\} \ , \tag{5.10}$$

where δ is the learning rate and Θ can be any entry of matrix F. Defining $\hat{x} := \frac{F_u F_i^T}{\|F_u\|_2\|F_i\|_2} - \frac{F_u F_j^T}{\|F_u\|_2\|F_j\|_2}$, the partial derivatives can be calculated by

$$\begin{cases} \dfrac{\partial \hat{x}}{\partial F_{u,t}} = \dfrac{\|F_u\|_2\|F_i\|_2 \cdot F_{i,t} - F_u F_i^T \cdot \frac{\|F_i\|_2}{\|F_u\|_2} \cdot F_{u,t}}{\|F_u\|_2^2\|F_i\|_2^2} \\ \qquad\qquad - \dfrac{\|F_u\|_2\|F_j\|_2 \cdot F_{j,t} - F_u F_j^T \cdot \frac{\|F_j\|_2}{\|F_u\|_2} \cdot F_{u,t}}{\|F_u\|_2^2\|F_j\|_2^2} \\ \dfrac{\partial \hat{x}}{\partial F_{i,t}} = \dfrac{\|F_u\|_2\|F_i\|_2 \cdot F_{u,t} - F_u F_i^T \cdot \frac{\|F_u\|_2}{\|F_i\|_2} \cdot F_{i,t}}{\|F_u\|_2^2\|F_i\|_2^2} \\ \dfrac{\partial \hat{x}}{\partial F_{j,t}} = -\dfrac{\|F_u\|_2\|F_j\|_2 \cdot F_{u,t} - F_u F_j^T \cdot \frac{\|F_u\|_2}{\|F_j\|_2} \cdot F_{j,t}}{\|F_u\|_2^2\|F_j\|_2^2} \end{cases} . \tag{5.11}$$

Algorithm 4 describes the whole iterative process of parameter learning. In each iteration, the time complexity is $O(|E|p)$, where $|E|$ is the number of edges and p the number of communities. Because we need to save the whole node-community weight matrix F in memory, the space complexity of the algorithm is $O(|V|p)$, where V is the number of nodes. When V becomes too large, the algorithm needs huge memory to store the whole matrix F, which is the limitation of the algorithm. To scale this algorithm to billions of nodes, distributed storage and update of F should be considered.

Choosing the number of communities.

Before running Algorithm 4, we need to set the number of communities p in advance. After conducting some experiments on small datasets, we find that if we set p to be larger than the

Input: G, the adjacency matrix of original graph; α, the acquaintance threshold; β, the stranger threshold
Output: F, the node-community weight matrix
initialize F
compute initial loss
repeat
  for *num_samples* $= 1$ to $|E|$ **do**
    sample node u from V uniformly at random
    sample node i from u's α-acquaintances set $A(u, \alpha)$ uniformly at random
    sample node j from u's β-strangers set $B(u, \beta)$ uniformly at random
    for each entry Θ in F_i, F_j and F_k **do**
      update Θ according to Equation (5.10)
    end for
  end for
  compute loss
until convergence or *max_iter* is reached

Algorithm 4: Overlapping community detection using *MD-NMF*

intended p and learn the parameters accordingly, our detected communities contain the results we obtain with the intended p as well as some duplicated communities or trivial communities with few nodes. Thus, our strategy is to pick a relatively large p based on the number of nodes and edges in the network and further refine our results via merging or deletion.

Acquaintances and strangers sampling

For node $u \in V$ and any of its α-acquaintances i, if $\alpha > 0$, then it is guaranteed that u and i have mutual friends. To find i, we first do a breadth-first search and group all u's neighbors as well as friends of these neighbors into a set. Then we filter out any node k with $d(u, k) < \alpha$ in this set and sample i from the remaining nodes uniformly at random. If u does not have any α-acquaintance, we sample another u and repeat the above process until we get a valid u. To sample the β-stranger of u, we simply sample a random node from graph until we get the β-stranger. From Table 1 we can see that in each graph, the average degree of nodes is much smaller than the number of edges. Thus the time complexity of sampling acquaintances and strangers for a node remains constant.

Different values of α and β may affect the result, so we need to choose the thresholds empirically. For α, we sample ten thousand node pairs with mutual friends and observe the distribution of mutual density of these node pairs. We can choose the average mutual density among these pairs or any value that with 30% or 40% of the samples greater than it. For β, we sample ten thousand node pairs without mutual friends and make the same observation.

Setting membership threshold

For each node, since its final community membership should be binary, our strategy is to set a membership threshold t for each entry of its node-community weight vector, i.e., if $F_{u,k} \geq t$, we say that node u is associated with community t. To determine the threshold, we assume that the node-community weight vector is a unit vector and each node can join at most n communities, with all the weights being the same. Thus, the membership threshold can be set as $t = \frac{1}{\sqrt{n}}$. Since our raw output, i.e., the node-community weight matrix F is not normalized, we need to normalize each row of it and then apply the membership threshold for each entry.

5.5 Experiments

In this section, we conduct experiments on our model with various real-world datasets. We compare the result of our model with other classic or state-of-the-art overlapping community detection methods. To evaluate the quality of the result, we use two metrics, which are modularity and F_1 score.

5.5.1 Dataset

The real-world datasets we use include the two large networks we have described in the data observation section, as well as six

Dataset	|V|	|E|
Dolphins	62	159
Books about US politics	105	441
American college football	115	613
Network science	1,589	2742
Power grid	4,941	6,594
High-energy theory	8,361	15,751

Table 5.3: Statistics of six Newman's datasets.

benchmark networks collected by Newman[2]. Table 5.3 lists the basic information of the six benchmark datasets, where $|V|$ is the number of nodes and $|E|$ is the number of edges. They are relatively small compared to the two large networks and have no ground-truth communities.

5.5.2 Comparison Methods

For comparison, we select the following seven baseline approaches:

- *Sequential Clique Percolation (SCP)* [49]. This method improves the original Clique Percolation method [74] in a sequential manner. We set k to be 4 or 5 when finding k-cliques.

- *Link Clustering (LC)* [1]. This method uses the concept of link similarity that has the same definition as our mutual

[2]http://www-personal.umich.edu/mejn/netdata

density to cluster links instead of nodes and finally obtain overlapping communities. We ignore all the trivial communities with only one or two nodes.

- *Demon* [18]. This method employs label propagation to detect small communities on ego network of each node and merge communities with large overlap.

- *Bayesian Non-negative Matrix Factorization (BNMF)* [80]. This method is the first MF-based model utilized in overlapping community detection, which is based on a generative graphical model.

- *Bounded Non-negative Matrix Tri-Factorization (BNMTF)* [113]. This method uses three factors to learn the community membership of each node as well as the interaction among communities. We use the squared loss as its loss function.

- *BigCLAM* [109]. This method is the first MF-based model designed for large networks. It is built on a bipartite affiliation network and aims to fit the underlying network by generating the same set of edges with maximum probability.

- *Preference-based Non-negative Matrix Factorization (PNMF).* The method we propose in Chapter 3, which uses the same framework as the BigCLAM model but incorporates the implicit preference information inside a link to come up with

a novel objective function.

Notice that the latter four approaches are also based on matrix factorization.

5.5.3 Evaluation Metrics

We use modularity as the evaluation metric for small datasets without ground-truth communities and F_1 score for large datasets with ground truth communities.

Modularity

The classic modularity is defined as

$$Q = \frac{1}{2|E|}\sum_{u,v\in V}(G_{u,v} - \frac{d(u)d(v)}{2|E|})I_{u,v},$$

where $d(u)$ is the degree of node u, $G_{u,v}$ is the (u, v) entry of the adjacency matrix G, and $I_{u,v} = 1$ if u, v are in the same community otherwise 0 [71].

In the overlapping scenario, since a node pair may share more than one communities, a minor modification has been made by replacing $I_{u,v}$ with $|C_u \cap C_v|$, i.e., the number of overlapped community between u and v:

$$\hat{Q} = \frac{1}{2|E|}\sum_{u,v\in V}(G_{u,v} - \frac{d(u)d(v)}{2|E|})|C_u \cap C_v|.$$

From the definition, we can see that greater value of modularity reveals denser connectivity within the detected communities because only linked node pairs sharing common communities

contribute positively to the value. This metric has also been frequently used in previous MF-based works [109].

As we know, modularity has been directly used as an optimization objective in community detection, and those approaches are called modularity-based methods [16, 24, 37, 69]. However, when we compare the quality of detected communities among non-modularity-based models, modularity can still be served as a useful metric.

F_1 score

The F_1 score of a detected community S_i is defined as the harmonic mean of

$$precision(S_i) = \max_j \frac{|S'_j \cap S_i|}{|S_i|}$$

and

$$recall(S_i) = \max_j \frac{|S'_j \cap S_i|}{|S'_j|},$$

i.e.,

$$F_1 = \frac{precision(S_i) \cdot recall(S_i)}{precision(S_i) + recall(S_i)},$$

where S'_j is one of the given ground-truth communities. The overall F_1 score of the result of detected communities is the average F_1 score of all communities in the detected communities set.

Dataset	SCP	BigCLAM	PNMF	MD-NMF
Amazon	0.0315	0.0441	0.0419	**0.0961**
DBLP	0.0967	0.0390	0.0985	**0.1013**

Table 5.4: Comparison of experiment results in terms of F_1 score.

5.5.4 Results

For the small networks, we set the learning rate θ as 0.5 and p ranging from 10 to 50. We assume each node joins at most 3 to 10 communities and set the threshold based on this assumption. For the large network datasets, we set θ much greater because the normalized term in cosine similarity limits the altered amount of weight in each gradient descent iteration. We set p ranging from 1,000 to 5,000 and assume that each node joins at most 100 communities. The maximum number of iteration is set to be 100, while in most cases F converges before reaching the iteration limit.

Figure 5.3 shows the results regarding modularity on six small benchmark networks without ground-truth communities. We can see that our MD-NMF model outperforms all baseline methods on all datasets on modularity, including LC that leverages the general concept of "neighborhood similarity" as well and $PNMF$ that is also based on a pairwise objective function but employs links as the indicator.

Table 5.4 shows the results on two large benchmark networks with ground-truth communities. We can see that only three of

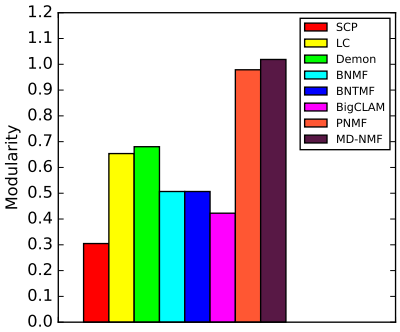
our comparison methods can scale to networks of such size. On both Amazon and DBLP dataset, our MD-NMF model prevails on the metric F_1 score.

5.5.5 Discussion

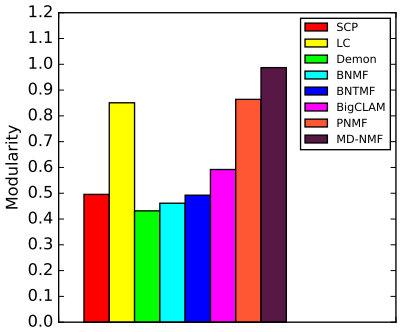
The learning objective of our MD-NMF model is similar to the PNMF model regarding formulation. The main difference between these two model is that the PNMF model uses link existence as the indicator of community membership similarity while our MD-NMF model uses mutual density. Thus, by comparing the results of both models on our benchmark networks, we can validate whether the results of MD-NMF model is consistent with its model assumption and whether mutual density is a better indicator than link existence.

The first issue we want to validate is the correctness of our MD-NMF model. From all the node pairs where both nodes belong to the same community, we count the number of pairs with no mutual friends (see Table 5.5). As we expect, the results of MD-NMF model have fewer node pairs with no mutual friends than the results of PNMF model, which illustrates the tendency of MD-NMF model to cluster nodes with mutual friends into same community.

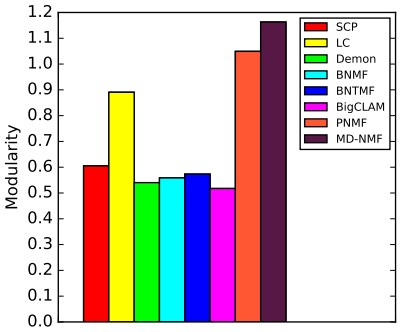
The second issue we want to validate is the superiority of our MD-NMF model. From all the node pairs with no links where both nodes belong to the same community, we count the number



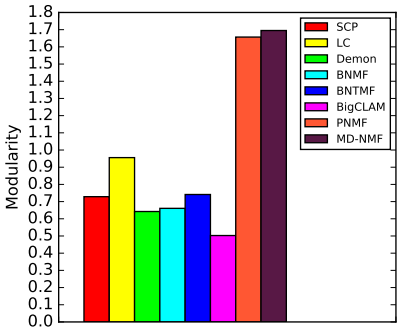
(a) Dolphins



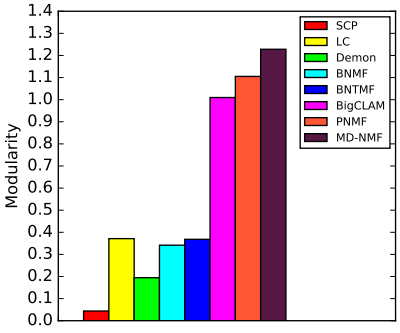
(b) Books about US politics



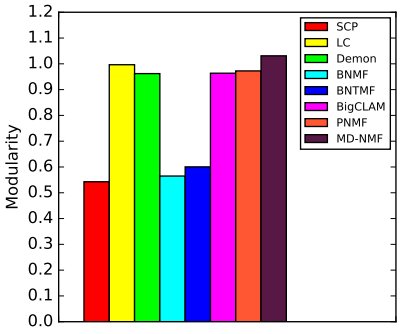
(c) American College football



(d) Network science



(e) Power grid



(f) High-energy theory

Figure 5.3: Comparison in terms of modularity.

Dataset	# of (u, v)		# of $m(u, v) = 0$	
	PNMF	**MD-NMF**	**PNMF**	**MD-NMF**
Dolphins	1,241	1388	588 (47.38%)	430 (30.98%)
Books about US politics	6,494	5,092	2,990 (46.04%)	809 (15.89%)
American college football	6,415	6,763	2,539 (39.58%)	1,950 (28.83%)
Network science	2,715,675	2,676,075	2,684,442 (98.85%)	2,568,619 (97.08%)
Power grid	6,605,705	6,109,728	6,586,528 (99.71%)	6,019,546 (98.52%)
High-energy theory	12,783,805	9,475,517	12,729,504 (99.58%)	9,255,685 (97.68%)

Table 5.5: The validation of correctness of MD-NMF model. u and v must be in the same community.

Dataset	# of $l(u,v)=0$		% of $m(u,v)\geq 1$		% of $m(u,v)\geq 2$	
	PNMF	**MD-NMF**	**PNMF**	**MD-NMF**	**PNMF**	**MD-NMF**
Dolphins	974	1,052	45.59%	63.41%	16.12%	28.71%
Books about US politics	5,391	3,939	45.32%	80.02%	26.64%	51.03%
American college football	5,081	4,959	53.61%	63.03%	21.94%	29.60%
Network science	2,695,401	2,594,222	0.48%	0.99%	0.10%	0.29%
Power grid	6,586,663	6,086,658	0.23%	1.28%	0.02%	0.13%
High-energy theory	12,756,611	9,226,311	0.21%	3.71%	0.04%	0.86%

Table 5.6: The validation of superiority of MD-NMF model. u and v must be in the same community.

of pairs with one or more mutual friends (see Table 5.6). The statistics show that the results of MD-NMF model have fewer node pairs without neither links nor mutual friends than results of PNMF model, which means the communities detected by the MD-NMF model is denser than those detected by the PNMF model. Another interesting phenomenon we observe is that even though our MD-NMF model focuses more on the mutual friend, it has more linked pairs inside communities than the PNMF model except in Dolphin. Thus, we are confident to say that mutual friend is a better indicator of community similarity than link existence.

5.6 Conclusion

In this chapter, we propose a *Mutual Density-based Non-negative Matrix Factorization* model for overlapping community detection. We introduce mutual density as a more consistent indicator of community membership similarity than links in traditional methods. The formulation of our model is based on empirical findings that mutual density correlates with the cosine similarity of community membership. Our learning objective maximizes the likelihood that each node has a more similar community membership with its acquaintances than its strangers. Experiment results show that our new model outperforms the other baseline methods as well as the link-based *PNMF* model in real-world datasets.

□ **End of chapter.**

Chapter 6

A Homophily-based NMF Model

Overlapping community detection has drawn much attention recently since it allows nodes in a network to have multiple community memberships. A standard framework to deal with overlapping community detection is *Matrix Factorization (MF)*. Although all existing MF-based approaches use links as input to identify communities, the relationship between links and communities is still under-investigated. Most of the approaches only view links as consequences of communities (community-to-link) but fail to explore how nodes' community memberships can be represented by their linked neighbors (link-to-community). In this chapter, we propose a *Homophily-based Non-negative Matrix Factorization (HNMF)* to model both-sided relationships between links and communities. From the community-to-link perspective, we apply a preference-based pairwise function by assuming that nodes with common communities have a higher

probability to build links than those without common communities. From the link-to-community perspective, we propose a new community representation learning with network embedding by assuming that linked nodes have similar community representations. We conduct experiments on several real-world networks, and the results show that our *HNMF* model can find communities with better quality compared with state-of-the-art baselines.

6.1 Introduction

A network is an abstraction representing relationships among real-world objects. A typical pattern of a network is that there are groups of nodes closely connected within the group but rarely making connections with nodes outside the group. Such groups are defined as *communities* [35]. The task of finding such communities from complex networks is referred as *community detection*, an important research topic in web mining for more than a decade. Usually, the more complex a network is, the more challenging it will be to identify such communities. It is mainly due to the infeasibility of visualization and the variety of community structure. Classic graph-partition-based community detection approaches assume that a node belongs to one and only one community, which contradicts with the fact that a node often appears with multiple memberships. To relax this unrealistic constraint, several new algorithms for *overlapping community detection* have been proposed in recent years.

A majority of existing methods for overlapping community detection is based on *Matrix Factorization (MF)* [80, 101, 113], which has been a standard technique in other areas such as recommender systems and natural language processing. The basic idea of MF here is to use low-dimensional latent vectors to represent nodes' features in networks. MF naturally fits into overlapping community detection since the dimensions of factorized latent vectors of nodes can be interpreted as their community memberships and hence are no longer latent. The MF-based overlapping community detection can be summarized into three steps: (1) assign the number of communities, (2) compute the node-community weight matrix F through a learning objective, and (3) obtain the final community set according to F. Here the most important part is the selection of learning objective. The simplest way is to recover the adjacency matrix of original network A by F with minimum error, i.e., to minimize $||A - FF^T||$ [80, 101]. However, an entry in A is a label (either 0 or 1) whereas an entry in F is a real value. The mismatch between label and entry does not make sense. To fix it, recent approaches such as [109] adopt generative objectives, which are based on the intuition that a node is more likely to build a link with another node inside its community than outside.

When we look into this intuition, it implicitly reveals that links are the consequence of communities (community-to-link), i.e., if two nodes share common communities, they will have a higher probability to be linked. However, the investigation in

reverse perspective (link-to-community) is largely ignored, i.e., whether a node's community membership can be represented by its neighbors' community membership. Taking MF as an example, the link-to-community perspective can be interpreted as to learn a node's community representation via the community representations of its neighbors. Here we use the word **homophily**, the tendency of an individual node to associate with similar others [98], to recapitulate both perspectives.

In this chapter, we propose a *Homophily-based Non-negative MF (HNMF)* to explicitly model the effect of homophily from both perspectives. From the community-to-link perspective, we apply a pairwise objective function in the *Preference-based Non-negative MF (PNMF)* model. From the link-to-community perspective, we develop a novel generative objective function based on unsupervised representation learning and network embedding. We combine both objective functions into a joint learning objective, in which parameter learning can be easily parallelized using asynchronous stochastic gradient descent. Through experiments on various real-world datasets, we demonstrate that our model can identify communities with better quality compared with state-of-the-art baselines and can be applied to large datasets.

Contributions. We summarize our main contribution of this chapter as follows,

1. Our work is the first to explore the link-to-community side

of homophily effect between links and communities in overlapping community detection. We justify it via observation on real-world datasets with ground-truth communities;

2. We propose a new learning objective to model both perspectives of homophily within the non-negative MF framework. Experiments show that our HNMF model can detect overlapping communities with better quality.

6.2 Data Observation

To validate the link-to-community perspective, we observe two large network datasets with ground-truth communities[1] [108] to see whether linked node pairs have more similar community representations than non-linked ones. These two datasets are:

- **Amazon**: a products co-purchasing network based on Customers Who Bought This Item Also Bought feature of the Amazon website.
- **DBLP**: a collaboration network of research papers authors in computer science;

Dataset	$\|V\|$	$\|E\|$	$\|S\|$	M	A
Amazon	335k	926k	49k	100.0	14.83
DBLP	317k	1.0M	2.5k	429.8	2.57

Table 6.1: Data statistics.

[1] http://snap.stanford.edu/data/

A simple statistics can be found in Table 6.1, where $|V|$ is the number of nodes, $|E|$ is the number of links, $|S|$ is the number of ground-truth communities, M is the average number of nodes per community, and A is the average community memberships per node.

We exploit the average number of shared communities (SC) and average Jaccard similarity of community memberships (JS) for all linked node pairs as our measurements. They are calculated by

$$SC = \frac{1}{2|E|} \sum_{i \in V} \sum_{j \in N^+(i)} |C_i \cap C_j|, \tag{6.1}$$

and

$$JS = \frac{1}{2|E|} \sum_{i \in V} \sum_{j \in N^+(i)} \frac{|C_i \cap C_j|}{|C_i \cup C_j|}, \tag{6.2}$$

respectively, where $N^+(i)$ is the set of i' neighbors and C_i represents the set of communities containing i. We also draw ten thousand random node pairs that do not need to be linked and compute the same measurements for these pairs. The comparison results are shown in Table 6.2, where **SC** is the average number of shared communities per linked node pair, $\mathbf{SC_r}$ is average number of shared communities per random node pair, **JS** is the average Jaccard similarity of community memberships per linked node pair, and $\mathbf{JS_r}$ is the average Jaccard similarity of community memberships per random node pair. The huge gap between linked ones (bold) and random ones (normal) reveals that two linked nodes share much more communities than two

random nodes in average and thus strongly supports the necessity of link-to-community perspective.

Dataset	SC	SC_r	JS	JS_r
Amazon	**6.767**	0.178	**0.490**	0.010
DBLP	**2.078**	0.009	**0.347**	0.002

Table 6.2: Data observations.

Moreover, we count the number of linked node pairs that share a particular number of communities in Figure 6.1 and Figure 6.2. In both networks, the number of linked node pairs reaches the peak near their average number of shared communities for linked node pairs and starts to decrease when this number continues to increase. This observation shows that average number of shared communities can be used to measure how strong the link-to-community side of homophily effect is. For example, we can claim that the link-to-community side of homophily effect in Amazon is much stronger than that in DBLP.

6.3 A Homophily-based Non-negative Matrix Factorization (HNMF) Model

In this section, we first introduce our model assumptions. Then we formalize our *HNMF* model from both perspectives and combine them into a unified model. In the end, we exhibit our parameter learning algorithm and discuss some more detailed issues. All the notations are listed in Table 6.3.

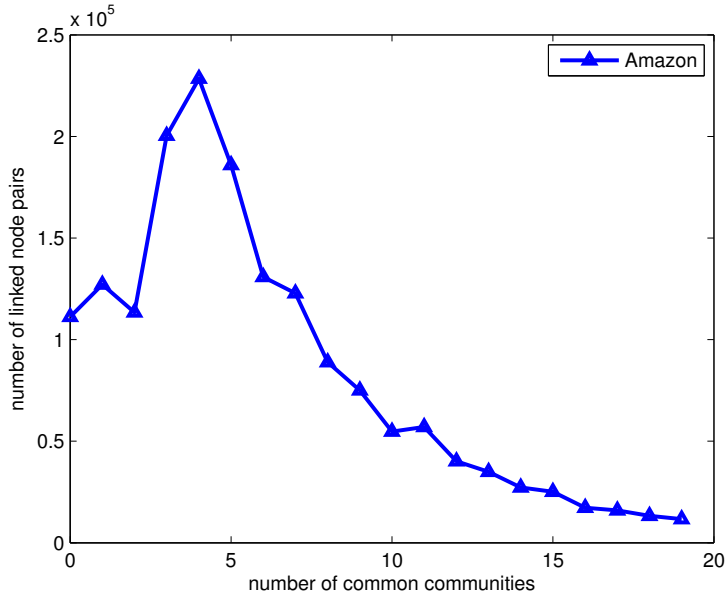


Figure 6.1: The number of linked node pairs sharing a particular number of communities for Amazon.

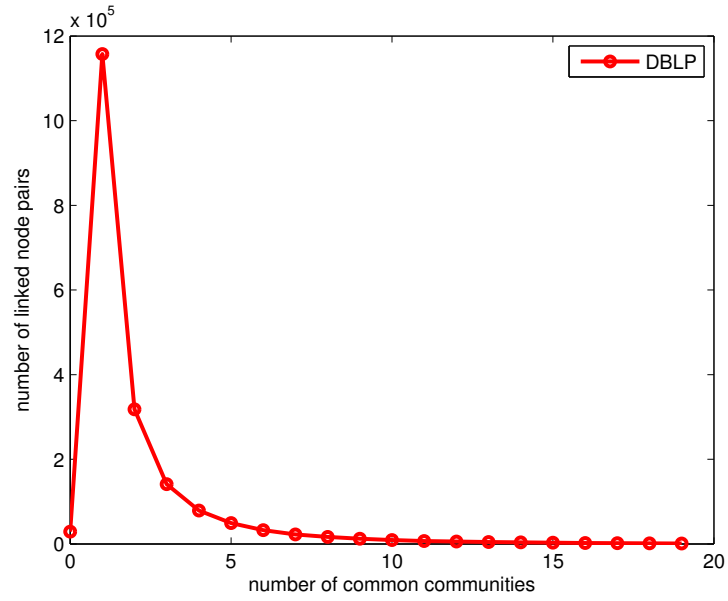


Figure 6.2: The number of linked node pairs sharing a particular number of communities for DBLP.

Notation	Meaning
$G(V, E)$	graph G (node set V, edge set E)
$A \in \{0,1\}^{\|V\|\times\|V\|}$	adjacency matrix of G
S	the set of detected communities
C_u	the set of communities containing u
$F \in \mathbb{R}^{\|V\|\times\|S\|}$	node-community weight matrix
F_u	u's community representation
$N^+(u)$	node set of u's neighbors
$N^-(u)$	node set of u's non-neighbors

Table 6.3: A summary of notations.

6.3.1 Model Assumption

Since we model homophily from both community-to-link and link-to-community perspectives, our model assumption will be introduced in two separate parts as well.

For the community-to-link perspective, the basic assumption is that two nodes should have higher probability to build links with each other if they share more communities, i.e.,

$$\mathcal{P}(A_{u,i} = 1) > \mathcal{P}(A_{u,j} = 1), \text{if } |C_u \cap C_i| > |C_u \cap C_j|. \tag{6.3}$$

Since we apply the *PNMF* model for this part, we also need to adopt the preference assumption, i.e.,

$$r_{u,i} > r_{u,j}, \text{if } i \in N^+(u) \text{ and } j \in N_u^-, \tag{6.4}$$

where $r_{u,i}$ is the preference of node u on node i. It indicates that a node prefers to build links with neighbors over non-neighbors.

For the link-to-community perspective, we assume that two linked nodes are more similar than two non-linked nodes. It is

formally denoted as:

$$sim_{u,i} > sim_{u,j}, \text{if } i \in N^+(u) \text{ and } j \in N_u^-, \tag{6.5}$$

where $sim_{u,i}$ is the similarity between node u and node i.

6.3.2 Modeling Community-to-link Perspective

We demonstrate our learning objective of community-to-link perspective by following the formulation of the *PNMF* model. For each node u, the objective of *PNMF* is to maximize the likelihood of a pairwise preference order, which can be denoted as $\mathcal{P}(>_u)$. According to the preference assumption, $\log \mathcal{P}(>_u)$ can be represented as:

$$\sum_{i \in N^+(u)} \sum_{j \in N^-(u)} \log \mathcal{P}(i >_u j). \tag{6.6}$$

Following the core idea of the community-to-link assumption, we use the community representations of node i, j, and k to model $\mathcal{P}(i >_u j)$. It can be written as

$$\mathcal{P}(i >_u j) = \sigma(F_u^T(F_i - F_j)), \tag{6.7}$$

where $\sigma(\cdot)$ is the sigmoid function $\sigma(x) := \frac{1}{1+e^{-x}}$. We choose sigmoid function because it is a differentiable function which can map any real number into the range between 0 and 1.

Based on Eq. (6.6) and Eq. (6.7), the learning objective of the community-to-link perspective can be derived by summing

up log-likelihoods of all the nodes, i.e.,

$$\begin{aligned}\mathcal{C}(F) :=& \sum_{u} \sum_{i \in N^+(u)} \sum_{j \in N^-(u)} \log \mathcal{P}(i >_u j) \\ =& \sum_{u} \sum_{i \in N^+(u)} \sum_{j \in N^-(u)} \log \sigma(F_u^T (F_i - F_j)).\end{aligned} \tag{6.8}$$

6.3.3 Modeling Link-to-community Perspective

Motivated by the success of *Skip-Gram* model [65] where word representations are learned in terms of representations of surrounding words in the same context, we here adopt a similar idea to learn a node's community representation from other nodes in its local scope. In our case, for a node u, its local scope is constrained within u's' neighbors. According to our link-to-community assumption, u's neighbors should have similar community representations with u. Formally, given a node u and its neighbors, our learning objective for the link-to-community perspective is to maximize the sum of log-likelihoods for a node to represent its neighbors as follows,

$$\sum_{i \in N^+(u)} \log \mathcal{P}(i|u). \tag{6.9}$$

Following the formulation in *Skip-Gram*, we apply a softmax function to define $\mathcal{P}(i|u)$ as

$$\mathcal{P}(i|u) = \frac{\exp({F'}_i^T F_u)}{\sum_{i'=1}^{|V|} \exp({F'}_{i'}^T F_u)}. \tag{6.10}$$

Note that F' needs to be introduced into our model and should be regarded as the latent community representation matrix which

is corresponding to the 'output' vector representations. Likewise, our learning target F is corresponding to the 'input' vector representations.

A computationally efficient approximation of the full softmax function is *Negative Sampling (NEG)*, which is simplified version of *Noise Contrastive Estimation (NCE)* [38]. It substitutes Eq. (6.10) with

$$\mathcal{P}(i|u) = \sigma({F'}_i^T F_u) + h\mathbb{E}_{i' \sim P_{N^-(u)}}\big[\sigma(-{F'}_{i'}^T F_u)\big], \tag{6.11}$$

where $\sigma(\cdot)$ is also the sigmoid function, h is the number of negative samples, and $P_{N^-(u)}$ is the unigram distribution raised to the power $\frac{3}{4}$.

Thus, we can obtain the learning objective of the link-to-community perspective as follows,

$$\begin{aligned}\mathcal{L}(F, F') := \sum_u \sum_{i \in N^+(u)} \big(& \log \sigma({F'}_i^T F_u) \\ & + h\mathbb{E}_{i' \sim P_{N^-(u)}}\big[\log \sigma(-{F'}_{i'}^T F_u)\big]\big).\end{aligned} \tag{6.12}$$

6.3.4 The Unified Model

Now we can combine the two perspectives, i.e., Eq. (6.8) and Eq. (6.12), into one unified model. The final learning objective

of our *HNMF* model is to maximize

$$\begin{aligned}\mathcal{U}(F, F') &:= \mathcal{C}(F) + \beta\mathcal{L}(F, F') - \lambda\mathcal{R}(F) \\ &= \sum_{u} \sum_{i \in N^+(u)} \Big(\sum_{j \in N^-(u)} \log \sigma(F_u^T (F_i - F_j)) \\ &+ \beta \log \sigma({F'}_i^T F_u) + \beta h \mathbb{E}_{i' \sim P_{N^-(u)}} \big[\log \sigma(-{F'}_{i'}^T F_u) \big] \Big) \\ &- \lambda ||F||_F,\end{aligned} \tag{6.13}$$

where $\mathcal{R}(F)$ is a regularization term, where we employ the Frobenius norm of F, β is the homophily coefficient used to adjust the importance of one perspective compared with the other, and λ is the regularization coefficient.

6.3.5 Parameter Learning

Considering time efficiency and the non-negativity constraint, we use projected stochastic gradient descent [55, 58] as our parameter learning method. It updates the corresponding parameters whenever a single sample or a small batch of samples arrive and maps the parameters back to the nearest point in the projected space, in our case, the non-negative space. The update rule for a parameter Θ is

$$\Theta^{t+1} = \max\{\Theta^t + \alpha \frac{\partial \mathcal{U}}{\partial \Theta}, 0\}, \tag{6.14}$$

where α is the learning rate.

The whole process of our learning method is shown in Algorithm 5. Here we discuss some of the steps in more detail.

Input: A, the adjacency matrix of original graph.

Output: F, the node-community weight matrix.

Initialization:

Initialize F (uniformly at random);

for *each node* u **do**

  Construct $N^+(u)$;

end

Training:

Compute initial loss;

repeat

  for *each node* u **do**

    Uniformly sample node i from $N^+(u)$;

    Community-to-link:

    Uniformly sample node j from $N^-(u)$

    $F_u = F_u + \alpha \frac{\partial \mathcal{C}}{\partial F_u}$;

    $F_i = F_i + \alpha \frac{\partial \mathcal{C}}{\partial F_i}$;

    $F_j = F_j + \alpha \frac{\partial \mathcal{C}}{\partial F_j}$;

    Link-to-community:

    Sample h negative nodes $i' \sim P_{N^-(u)}$;

    $F_u = F_u + \alpha\beta \frac{\partial \mathcal{L}}{\partial F_u}$; $F'_i = F'_i + \alpha\beta \frac{\partial \mathcal{L}}{\partial F'_i}$;

    for *each node* i' **do**

      $F'_{i'} = F'_{i'} + \alpha\beta \frac{\partial \mathcal{L}}{\partial F'_{i'}}$;

    end

    Regularization and Projection:

    $F_u = \max\{F_u - \alpha\lambda \frac{\partial \mathcal{R}}{\partial F_u}, 0\}$;

    $F_i = \max\{F_i - \alpha\lambda \frac{\partial \mathcal{R}}{\partial F_i}, 0\}$;

    $F_j = \max\{F_j - \alpha\lambda \frac{\partial \mathcal{R}}{\partial F_j}, 0\}$;

  end

  Compute loss;

until *Convergence or max_iter is reached*;

Algorithm 5: Overlapping community detection using *HNMF*

- Initialization. We initialize each entry of F to be a random real value between 0 and 1 divided by the number of communities, i.e., the number of columns in F.

- Negative sampling. For the negative sample j from $N^-(u)$, we keep sampling j from V until $j \notin N^+(u)$.

- Convergence criterion. We randomly sample a number of triples (u, i, j) and use them to compute the initial loss on according to Eq. (6.13) without considering the regularization term. After each iteration, we repeat the same process with a different set of samples and stop when the difference between the current loss and previous loss is less than a small value, say ϵ, of the initial loss.

- Setting the number of communities. We adopt a cross-validation paradigm by reserving 10% of nodes as a validation set. Since the computational cost on the validation set is still huge, sampling will be used as well.

6.3.6 Other Issues

Scalability. To scale up our *HNMF* model on large networks, we employ an asynchronous version of stochastic gradient descent to update the parameters. Since most updates only modify a small part of all the parameters, the chance that a parameter is simultaneously being updated by more than one worker is very small. Thus, a lock-free approach [84] can be adopted to

parallelize our parameter learning process. We will show in the experiments that the convergence speed is satisfactory.

Community membership threshold. After we obtain the node-community weight matrix F, we still need to figure out community memberships for each node. A standard solution is to set a threshold and discard all the nodes whose weights are below the threshold. Here we employ the same approach as the *PNMF* model.

6.4 Experiments

In this section, we compare our *HNMF* model with six baselines on various real-world datasets, including large networks with ground-truth communities. We measure the quality of communities with two metrics, modularity and F_1 score. Our experimental procedures and results are described as follows.

6.4.1 Data Description

Apart from the two large networks with ground-truth communities introduced in Section 6.2, we also use six benchmark networks collected by Newman[2] as our datasets. These networks are relatively small and have no ground-truth communities. We list the basic information of these datasets in Table 6.4, where $|V|$ is the number of nodes, $|E|$ is the number of links.

[2]http://www-personal.umich.edu/ mejn/netdata/

Dataset	$\|V\|$	$\|E\|$
Dolphins	62	159
Les Misérables	77	254
Books about US politics	105	441
Word adjacencies	112	425
American college football	115	613
High-energy theory	8,361	15,751

Table 6.4: Statistics of six Newman's datasets.

6.4.2 Experimental Setup

Comparison methods. We select two local approaches, namely *Sequential Clique Percolation (SCP)* [49] and *Demon* [18], and four state-of-the-art global approaches, namely *BNMF* [80], *BNMTF* [113], *BigCLAM* [109], and *PNMF*, to compare with our *HNMF* model.

Evaluation metrics. We use modularity for datasets without ground-truth communities and F_1 score for datasets with ground-truth communities.

- **Modularity.** The well-known modularity [71] Q is defined as

$$Q = \frac{1}{2|E|} \sum_{u,v \in V} (A_{u,v} - \frac{d(u)d(v)}{2|E|}) |C_u \cup C_v|,$$

  where $d(u)$ is u's degree. We can see that a node pair (u, v) positively contributes to modularity if they are linked and negatively contributes otherwise.

- **$\mathbf{F_1}$ score.** F_1 score of a detected community S_i is defined

as the harmonic mean of

$$\text{precision}(S_i) = \max_j \frac{|\hat{S}_j \bigcap S_i|}{|\hat{S}_j|}$$

and

$$\text{recall}(S_i) = \max_j \frac{|\hat{S}_j \bigcap S_i|}{|S_i|},$$

where $\hat{S}_j$ is one of ground-truth communities. The overall F_1 score of the set of detected communities S is the average F_1 score of all communities in S.

6.4.3 Results

Results on Newman's networks in terms of modularity are shown in Table 6.5. Despite that *PNMF* already has a large improvement over other baselines, our *HNMF* model further outperforms *PNMF* on all datasets, which reflects the significance of the link-to-community perspective in overlapping community detection.

Results on two large networks in terms of the F_1 score are shown in Table 6.6. We notice that the improvement on Amazon is much larger than that of DBLP. Recall our claim in data observation that the link-to-community side of homophily effect in Amazon is much stronger than that in DBLP. This explains such difference of improvement between these two datasets. With asynchronous stochastic gradient descent, the running time of our learning algorithm is about 4 hours for Amazon and about

Dataset	**SCP**	**Demon**	**BNMF**	**BNMTF**	**BigCLAM**	**PNMF**	**HNMF**
Dolphins	0.305	0.680	0.507	0.507	0.423	0.979	**1.021**
Books about US politics	0.496	0.432	0.461	0.492	0.529	0.864	**0.988**
Word adjacencies	0.071	0.032	0.254	0.268	0.231	0.668	**0.699**
American college football	0.605	0.540	0.558	0.573	0.518	1.049	**1.113**
Power grid	0.044	0.195	0.342	0.368	1.010	1.105	**1.135**
High-energy theory	0.543	0.962	0.565	0.600	0.964	0.973	**1.060**

Table 6.5: Experimental results on Newman's networks in terms of modularity.

6 hours for DBLP on a computer with a Xeon 24-core 2.60GHz CPU and 128GB memory.

Dataset	Demon	BigCLAM	PNMF	HNMF
Amazon	0.082	0.044	0.042	**0.122**
DBLP	0.102	0.039	0.098	**0.104**

Table 6.6: Experimental results on two large networks in terms of F_1 score.

Figure 6.3 illustrates the convergence speed of our learning algorithm on Amazon and DBLP. Since our computation of loss employs a global sampling strategy, we can directly sort the losses from all workers according to the time sequence. We set the ϵ in Section 4.5 to be 0.001. We can see that our learning algorithm can converge within a small number of iterations.

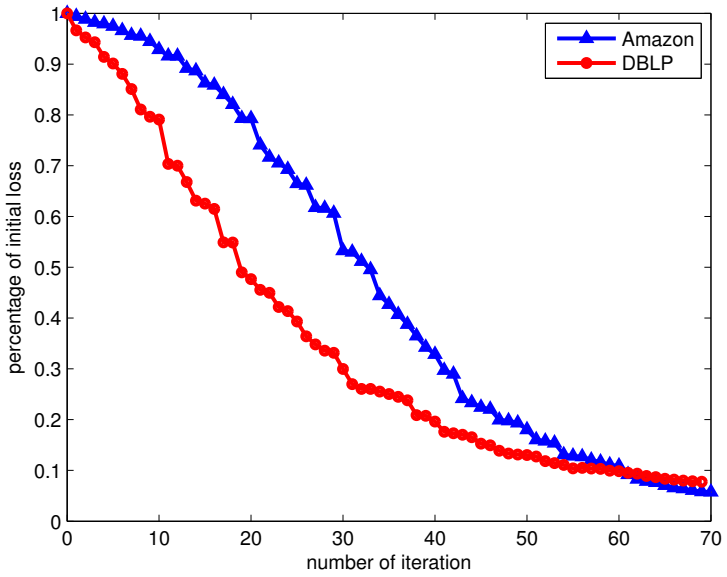


Figure 6.3: Convergence speed of our learning algorithm.

6.5 Conclusion

In this chapter, we propose a *Homophily-based Non-negative Matrix Factorization* model to capture both sides of homophily effect for overlapping community detection. Our unified learning objective is a combination of a preference-based pair-wise learning objective for the community-to-link perspective and a generative community representation learning with network embedding for the link-to-community perspective. We adopt an asynchronous stochastic gradient descent to learn model parameters efficiently. Experiments on real-world networks show that this model can indeed improve the quality of detected overlapping communities.

□ **End of chapter.**

Chapter 7

Conclusion

In this chapter, we summarize the main contributions of this thesis and discuss several potential research directions.

7.1 Summary

This thesis mainly focus on the matrix factorization framework for overlapping community detection. We propose several non-negative matrix factorization models with novel learning objectives which incorporate various concepts including link preference, locality, mutual density, and homophily.

In Chapter 3, we present a *Preference-based Non-negative Matrix Factorization (PNMF)* model which incorporates implicit link preference information into model formulation. We make a intuitive assumption that a node prefers its neighbors than its "non-neighbors" and thus build a novel learning objective of maximizing the likelihood of a preference order for each node. Our *PNMF* model eliminates indiscriminate pen-

alty issue caused by on pairs inside and between communities. We employ stochastic gradient descent with bootstrap sampling to learn the node-community membership matrix. By applying our *PNMF* model on several real-world datasets both with and without ground-truth communities, we show that our *PNMF* model outperforms state-of-art approaches in multiple metrics and is scalable for large datasets.

In Chapter 4, we present a *Locality-based Non-negative Matrix Factorization (LNMF)* model to further improve the performance of the *PNMF* model. Same as the *PNMF* model, our *LNMF* model is also based on a pairwise preference learning scheme. The main contribution is that we exploit local area around a node, formally defined as a k-degree local network, to enhance the previous preference system. To be specific, we extend the two-level preference system of *PNMF* which only distinguish neighbors and non-neighbors to a three-level preference system which further splits non-neighbors into local non-neighbors and distant non-neighbors. Experiments on several real-world datasets including large ones with ground-truth communities illustrate the effectiveness of this extension.

In Chapter 5, we present a *Mutual Density based Non-negative Matrix Factorization (MD-NMF)* model which incorporates mutual density to replace link existence as the indicator of community membership similarity. The formulation of our *MD-NMF* model is based on empirical observations that mutual density correlates with the cosine similarity of community membership.

A novel learning objective is proposed by maximizing the likelihood that each node has a more similar community membership with its acquaintances than its strangers. Experiment results on multiple real-world datasets show that our *MD-NMF* model outperforms baseline methods including those using link existence as the indicator of community membership similarity.

Finally, in Chapter 6, we present a *Homophily-based Nonnegative Matrix Factorization (HNMF)* model to capture both sides of homophily effect. We propose a unified learning objective which combines a preference-based pair-wise learning objective for the community-to-link perspective and a generative community representation learning with network embedding for the link-to-community perspective. We adopt an asynchronous stochastic gradient descent to learn model parameters in parallel. Experiments on various real-world networks show that our *HNMF* model successfully captures the homophily effect.

7.2 Future Work

Although the *LNMF* model is already an extension of our *PNMF* model, we can further generalize the preference system from three-level to an n-level. However, according to the *six degrees of separation* theory that all living things and everything else in the world are six or fewer steps away from each other, it is meaningless to set n larger than six. Also, a natural extension of our sampling strategy employed in the *LNMF* model suffers from

scalability issue. For each node, processing the whole network to save node sets of all preference levels in advance is equal to a breadth-first search starting from this node, which is too time-consuming for large networks.

The *MD-NMF* model can be combined with the *PNMF* model so that the indicator of mutual density and the indicator of link existence together can reveal more comprehensive preferences for a node. For example, we can simply unite learning objectives of these two models in a linear joint model with hyper-parameters controlling the weights.

The *HNMF* model employs the *PNMF* model to represent the community-to-link perspective. In fact, the *PNMF* model can be replaced by several alternatives, including our *LNMF* model and the *BigClam* model proposed by Yang et al. [109]. Comparison among different choices may give interesting insights on both datasets and models.

□ **End of chapter.**

Bibliography

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(1981-2014):3, 2008.

[3] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40, 2009.

[4] H. Alvari, S. Hashemi, and A. Hamzeh. Detecting overlapping communities in social networks by game theory and structural equivalence concept. In *International Conference on Artificial Intelligence and Computational Intelligence*, pages 620–630. Springer, 2011.

[5] M. G. Armentano, D. L. Godoy, and A. A. Amandi. A topology-based approach for followees recommendation in twitter. In *Workshop chairs*, page 22, 2011.

[6] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[7] M. J. Barber and J. W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2):026129, 2009.

[8] E. R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM Journal on Algebraic Discrete Methods*, 3(4):541–550, 1982.

[9] J. Baumes, M. K. Goldberg, M. S. Krishnamoorthy, M. Magdon-Ismail, and N. Preston. Finding communities by clustering a graph into overlapping subgraphs. *IADIS AC*, 5:97–104, 2005.

[10] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[11] S. Boettcher and A. G. Percus. Optimization with extremal dynamics. *complexity*, 8(2):57–62, 2002.

[12] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. *On modularity-np-*

completeness and beyond. Univ., Fak. für Informatik, Bibliothek, 2006.

[13] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004.

[14] W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Mining and Knowledge Discovery*, 21(2):224–240, 2010.

[15] C. M. Cheung, P.-Y. Chiu, and M. K. Lee. Online social networks: Why do students use facebook? *Computers in Human Behavior*, 27(4):1337–1343, 2011.

[16] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[17] M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(5):512–546, 2011.

[18] M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Demon: a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD*

international conference on Knowledge discovery and data mining, pages 615–623. ACM, 2012.

[19] L. Danon, A. Díaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(11):P11010, 2006.

[20] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[21] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti. On facebook, most ties are weak. *Communications of the ACM*, 57(11):78–84, 2014.

[22] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, 1973.

[23] H. Du, M. W. Feldman, S. Li, and X. Jin. An algorithm for detecting community structure of social networks based on prior knowledge and modularity. *Complexity*, 12(3):53–60, 2007.

[24] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2):027104, 2005.

[25] P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[26] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[27] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM, 2000.

[28] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization and identification of web communities. *Computer*, 35(3):66–70, 2002.

[29] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian journal of Mathematics*, 8(3):399–404, 1956.

[30] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[31] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[32] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[33] A.-C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A.-M. Michon, C.-M. Cruciat, et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.

[34] E. Gilbert and K. Karahalios. Predicting tie strength with social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 211–220. ACM, 2009.

[35] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[36] R. Gross and A. Acquisti. Information revelation and privacy in online social networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 71–80. ACM, 2005.

[37] R. Guimera, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004.

[38] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.

[39] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova. Community detection in large-scale networks: a survey and empirical evaluation. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(6):426–439, 2014.

[40] F. Havemann, M. Heinz, A. Struck, and J. Gläser. Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(01):P01023, 2011.

[41] D. Jin, Z. Chen, D. He, and W. Zhang. Modeling with node degree preservation can accurately find communities. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[42] B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1):016107, 2011.

[43] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell system technical journal*, 49(2):291–307, 1970.

[44] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[45] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8), 2009.

[46] B. Krishnamurthy and J. Wang. On network-aware clustering of web clients. *ACM SIGCOMM Computer Communication Review*, 30(4):97–110, 2000.

[47] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440(7084):637–643, 2006.

[48] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *Link mining: models, algorithms, and applications*, pages 337–357. Springer, 2010.

[49] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2):026109, 2008.

[50] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[51] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in

complex networks. *New Journal of Physics*, 11(3):033015, 2009.

[52] A. Lancichinetti, F. Radicchi, J. J. Ramasco, S. Fortunato, et al. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

[53] P. Latouche, E. Birmelé, and C. Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336, 2011.

[54] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[55] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[56] J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

[57] Y. Li, K. He, D. Bindel, and J. E. Hopcroft. Uncovering the small community structure in large networks: A local spectral approach. In *Proceedings of the 24th international conference on world wide web*, pages 658–668. ACM, 2015.

[58] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[59] X. Liu and T. Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.

[60] S. Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[61] L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.

[62] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 931–940. ACM, 2008.

[63] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[64] J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, volume 272, pages 548–556, 2012.

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[66] A. Mnih and R. R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[67] M. Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, 2012.

[68] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[69] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[70] M. E. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[71] M. E. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[72] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[73] K. Nowicki and T. A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[74] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[75] W. Pan and L. Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *IJCAI*, volume 13, pages 2691–2697, 2013.

[76] V. Pancaldi, Ö. S. Saraç, C. Rallis, J. R. McLean, M. Převorovskỳ, K. Gould, A. Beyer, and J. Bähler. Predicting the fission yeast protein interaction network. *G3: Genes— Genomes— Genetics*, 2(4):453–467, 2012.

[77] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3):515–554, 2012.

[78] Y. Pei, N. Chakraborty, and K. Sycara. Nonnegative matrix tri-factorization with graph regularization for community detection in social networks. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[79] C. Peng, Z. Zhang, K.-C. Wong, X. Zhang, and D. Keyes. A scalable community detection algorithm for large graphs using stochastic block models. In *IJCAI*, pages 2090–2096, 2015.

[80] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*, 83(6):066114, 2011.

[81] J. M. Pujol, J. Béjar, and J. Delgado. Clustering algorithm for determining community structure in large networks. *Physical Review E*, 74(1):016107, 2006.

[82] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[83] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[84] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

[85] K. Reddy, M. Kitsuregawa, P. Sreekanth, and S. Rao. In dnis'02: Proceedings of the second international workshop on databases in networked information systems, 2002.

[86] F. Reid, A. McDaid, and N. Hurley. Partitioning breaks communities. In *Mining Social Networks and Security Informatics*, pages 79–105. Springer, 2013.

[87] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 727–736. ACM, 2009.

[88] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[89] J. Ruan and W. Zhang. Identifying network communities with a high resolution. *Physical Review E*, 77(1):016104, 2008.

[90] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.

[91] V. Sankar, B. Ravindran, and S. Shivashankar. Ceil: a scalable, resolution limit free approach for detecting communities in large networks. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[92] P. Schuetz and A. Caflisch. Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, 77(4):046112, 2008.

[93] P. Schuetz and A. Caflisch. Multistep greedy algorithm identifies community structure in real-world and computer-generated networks. *Physical Review E*, 78(2):026112, 2008.

[94] N. B. Silva, R. Tsang, G. D. Cavalcanti, and J. Tsang. A graph-based friend recommendation system using genetic algorithm. In *IEEE Congress on Evolutionary Computation*, pages 1–7. IEEE, 2010.

[95] N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2005.

[96] K. Steinhaeuser and N. V. Chawla. Community detection in a large real-world social network. In *Social computing, behavioral modeling, and prediction*, pages 168–175. Springer, 2008.

[97] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Product recommendation and rating prediction based on multi-modal social networks. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 61–68. ACM, 2011.

[98] J. Tang, H. Gao, X. Hu, and H. Liu. Exploiting homophily effect for trust prediction. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 53–62. ACM, 2013.

[99] L. Tang and H. Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1):1–137, 2010.

[100] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. In *Communities and technologies*, pages 81–96. Springer, 2003.

[101] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.

[102] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[103] J. J. Whang, D. F. Gleich, and I. S. Dhillon. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2099–2108. ACM, 2013.

[104] B. Xiang, E.-H. Chen, and T. Zhou. Finding community structure based on subgraph similarity. *Complex Networks*, pages 73–81, 2009.

[105] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys (CSUR)*, 45(4):43, 2013.

[106] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[107] J. Yang and J. Leskovec. Community-affiliation graph model for overlapping network community detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1170–1175. IEEE, 2012.

[108] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3. ACM, 2012.

[109] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[110] H. Zhang, I. King, and M. R. Lyu. Incorporating implicit link preference into overlapping community detection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 396–402. ACM, 2015.

[111] H. Zhang, M. R. Lyu, and I. King. Exploiting k-degree locality to improve overlapping community detection. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pages 2394–2400. ACM, 2015.

[112] H. Zhang, T. Zhao, I. King, and M. R. Lyu. Modeling the homophily effect between links and communities for overlapping community detection. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016*, pages 3938–3944. ACM, 2016.

[113] Y. Zhang and D.-Y. Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 606–614. ACM, 2012.

[114] T. Zhao, J. McAuley, and I. King. 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*, pages 261–270. ACM, 2014.