GRAPH BASED HYBRID RECOMMENDER SYSTEMS

Zühal Kurt
Doktora Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

Mayıs 2019

GRAPH-BASED HYBRID RECOMMENDER SYSTEMS

Zühal KURT

Ph.D. Dissertation

Computer Engineering Department
Supervisor: Assoc. Prof. Dr. Kemal ÖZKAN

Eskişehir
Anadolu University
Graduate School of Sciences
May 2019

FINAL APPROVAL FOR THESIS

This thesis titled “Graph-Based Hybrid Recommender Systems” has been prepared and submitted by Zühal Kurt in partial fulfillment of the requirements in “Anadolu University Directive on Graduate Education and Examination” for the Degree of Doctor of Philosophy (Ph.D.) in Computer Engineering Department has been examined and approved on 09/05/2019.

Committee Members	**Title Name Surname**	**Signature**
Member (Supervisor)	: Assoc. Prof. Dr. Kemal ÖZKAN	.....
Member	: Prof. Dr. Ömer Nezih GEREK	.....
Member	: Assist. Prof. Dr. Alper BİLGE	.....
Member	: Prof. Dr. İdris DAĞ	.....
Member	: Assoc. Prof. Dr. Cihan KALELİ	.....

Director of Graduate School of Sciences

ABSTRACT

GRAPH-BASED HYBRID RECOMMENDER SYSTEMS

Zühal KURT

Department of Computer Engineering

Anadolu University, Graduate School of Sciences, May 2019

Supervisor: Assoc. Prof. Dr. Kemal ÖZKAN

Recommender systems provide recommendations about various products and services to their users by using other users' data. These systems depend on personal user preferences on items via ratings and recommend items based on choices of similar users. Their success is imperative for both users and the e-commerce vendors utilizing such systems. Since inaccurate and unreliable product recommendations make users search alternative sites for shopping. Hence, recommender systems are a challenging research field with many unresolved problems and many different hybrid recommendation algorithms have been proposed to overcome these problems. Hybrid models that use different information sources (text, images, ratings, etc.) for recommendation are getting more attention in recent years. In this dissertation, a graph-based hybrid recommender system is proposed that is incorporating numerical ratings and product images to learn items and the corresponding user's representations. Moreover, another graph-based recommender system, that utilizes only user-item ratings, is proposed. In the current literature, recommendation generation in a graph based model is a link prediction problem and link prediction approaches are used to distinguish between fundamental relational dualities of *like* or *dislike* and *similar* or *dissimilar*. However, *similar* and *dissimilar* relationships between users (or items) are mostly disregarded. Hence, a link prediction method is proposed that utilizes user-user and item-item *similar/dissimilar* relationships with *like/dislike* dualities in order to improve the accuracy of the system. Similarly, triangle closing model is expanded with similarity relationships, and then the number systems are investigated to represent each similarity entity as a number. The usage of link prediction algorithms is examined for the quaternion and the complex number systems. On the standard Amazon and MovieLens datasets, the proposed similarity-inclusive link prediction method performed empirically well compared to other methods operating in the quaternion and complex domain. The experimental results show that the proposed recommender system can be a plausible alternative to overcome the deficiencies in recommender systems.

Keywords: Complex Domain, Graph, Hybrid Recommender Systems, Link Prediction, Quaternion.

ÖZET

GRAF TABANLI MELEZ ÖNERİ SİSTEMLERİ

Zühal KURT

Bilgisayar Mühendisliği Anabilim Dalı
Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, May 2019

Danışman: Doç. Dr. Kemal ÖZKAN

Öneri sistemleri, diğer kullanıcılarının verilerini kullanarak kullanıcılarına çeşitli ürün ve hizmet önerilerinde bulunmaktadır. Bu sistemler, kullanıcıların ürünlere ilişkin kişisel tercihlerini verdikleri oylamalara dayandırarak ve benzer kullanıcıların tercihlerine dayanan ürün önerileri sunarlar. Bu sistemlerin başarısı hem kullanıcılar için hem de bu tür sistemleri kullanan e-ticaret siteleri için önemlidir. Yanlış ve güvenilir olmayan ürün önerileri kullanıcıların alternatif alışveriş sitelerine yönelmesine neden olmaktadır. Bu nedenle, öneri sistemleri alanı birçok çözülmemiş problemi ihtiva eden zorlu bir araştırma alanıdır ve sorunların üstesinden gelmek için birçok farklı hibrit öneri algoritması önerilmiştir. Farklı bilgi kaynaklarını kullanan hibrit modeller son yıllarda fazla dikkat çekmektedir. Bu tezde, ürünleri ve ilgili kullanıcıları temsil edebilmek için oylamaları ve ürünlere ait imgeleri kullanan bir graf tabanlı hibrit öneri sistemi önerilmektedir. Ayrıca, sadece kullanıcı-ürün oylamalarından yararlanan başka bir graf tabanlı öneri sistemi önerilmiştir. Bir grafta öneri üretme bir bağlantı tahmin problemidir ve bağlantı tahmin yaklaşımları, *beğenme* veya *beğenmeme,* ve *benzer* veya *benzerolmayan* temel ilişkisel dualiteleri birbirinden ayırmak için kullanılmaktadır. Ancak, kullanıcılar (veya öğeler) arasındaki *benzer* veya *benzer olmayan* ilişkiler çoğunlukla göz ardı edilmektedir. Dolayısıyla, sistemin doğruluğunu arttırmak için kullanıcı-kullanıcı ve ürün-ürün *benzer/benzerolmayan* ilişkileri ile *beğenme*/*beğenmeme* ikiliklerini kullanan bir bağlantı tahmin metodu önerilmiştir. Benzer şekilde üçgen kapanış modeli bu benzerlik ilişkileriyle genişletilmiş ve her bir benzerlik ilişkisini bir sayı olarak temsil etmek için sayı sistemleri araştırılmıştır. Bağlantı tahmin algoritmalarının kuaterniyon ve kompleks sayı sistemlerinde kullanılışı incelenmiştir. Önerilen benzerlik içeren bağlantı tahmin yöntemi diğer kuaterniyon ve kompleks sayı sistemlerinde çalışan yöntemlerle karşılaştırıldığında standart Amazon ve MovieLens veri kümeleri üzerindeki deneylerde daha iyi bir performans göstermiştir. Deneysel sonuçlar, önerilen öneri sisteminin, öneri sistemlerindeki eksikliklerin üstesinden gelmek için tercih edilebilir alternatif bir sistem olabileceğini göstermektedir.

Anahtar Kelimeler: Kompleks Uzay, Graf, Melez Öneri Sistemi, Bağlantı Tahmini, Quaterniyon.

ACKNOWLEDGEMENTS

First, I would like to thank my dissertation committee members Assoc Prof. Dr. Kemal ÖZKAN, Prof. Dr. Ömer Nezih Gerek, and Assist. Prof. Dr. Alper Bilge for their guidance, advice and valuable knowledge; encouraging and motivating me to develop this work.

Especially, I would like to express my sincere thanks to Assoc Prof. Dr. Kemal ÖZKAN, Prof. Dr. Ömer Nezih Gerek, Assist. Prof. Dr. Alper Bilge, Prof. Dr. Idris Dağ, and Assoc Prof. Dr. Cihan Kaleli for their valuable contributions as well as serving on my committee.

I would like to thank the Scientific and Technological Research Council of Turkey (TUBITAK) for providing partial financial support throughout this dissertation with grant TUBITAK 2211-E. Moreover, I am also thankful to the Technological Research Council of Turkey for the support throughout this dissertation which is a part of the TUBITAK 3001 project number, 116E284.

Finally, thanks to my family and all friends for their emotional supports and inspired me to reach my dreams.

Zühal Kurt

May 2019

STATEMENT OF COMPLIANCE WITH ETHICAL PRINCIPLES AND RULES

I hereby truthfully declare that this thesis is an original work prepared by me; that I have behaved in accordance with the scientific ethical principles and rules throughout the stages of preparation, data collection, analysis and presentation of my work; that I have cited the sources of all the data and information that could be obtained within the scope of this study, and included these sources in the references section; and that this study has been scanned for plagiarism with "scientific plagiarism detection program" used by Anadolu University, and that "it does not have any plagiarism" whatsoever. I also declare that, if a case contrary to my declaration is detected in my work at any time, I hereby express my consent to all the ethical and legal consequences that are involved.

09/05/2019

Zühal KURT

CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

Page

Page

LIST OF ABBREVIATIONS

A	: Adjacency Matrix
B	: Biadjacency Matrix
G	: Graph
V	: Vertices
CBF	: Content-Based Filtering
CF	: Collaborative Filtering
CORLP	: Complex Representation-Based Link Prediction
SIMLP	: Similarity-Inclusive Link Prediction
Q-SIMLP	: Quaternion-Based Similarity-Inclusive Link Prediction
Q-Hybrid	: Quaternion-Based Hybrid Recommender System
CFKG	: Collaborative Filtering with Knowledge Graph
JRL	: Joint Representation Learning

1. INTRODUCTION

In general terms, in the context of data mining, information filtering is a term used to describe a variety of processes involving the delivery of information to people who need it [1]. Filtering includes eliminating data that seem needless. This process is vital in this information age since the amount of data that is accessible online has expanded exponentially. In recent years, it is often hard for users to cope with this information overload and access the data that is necessary without the help of data mining systems and artificial intelligence. Recommender systems are the application of filtering systems and they are used to recommend or help the user find an item among a collection of items that will most probably be liked by the user.

A recommendation system consists of a particular sort of information filtering method that provides recommendations about items based on the interests that a user states. Generally, recommender systems are employed in e-commerce sites and customer-adapted websites. Users demand comfort and convenience in their interactions, and the business demands a higher chance of commerce. Hence, the success of the recommendation system is imperative for both users and e-commerce sites. Selling more products/services and increasing user satisfaction depend on the generation of precise and dependable recommendations. In general, the prediction of ratings for items that have not been considered is achieved by using customer profiles [1]. Depending on the application domain, items can be movies, websites or other products discovered on an online store. For example, Amazon and Netflix use recommendation systems in the sense that Amazon typically suggests books and other articles (as well as many types of commercial items), and Netflix typically suggests movies and TV series to their customers. Even though various algorithms for recommender systems have been developed in recent years, there are still high levels of enthusiasm in this area caused by the growing requirement on functional processes, which can supply customized recommendations and help to deal with information overload problem [1, 2].

Recommender systems are generally categorized according to their approach to generation of a recommendation. In general, there exist two primary recommendation methods, i.e., content-based filtering (CBF) and collaborative filtering (CF) methods.

1.1. Content-Based Filtering Methods

Techniques of CBF [3] are usually dependent on the similarity of items to the objects that were previously preferred by the user. The recommender systems that use CBF methods suggest items to users by analyzing the item descriptions in order to identify which items are of interest to a particular user. The recommended items are similar in content to the items that the user was interested in earlier. Thus, item representation and user profiling are the major concerns of the CBF recommender systems [3]. There are several ways to represent items and users for content-based approaches. Content-based item representations have relied on item descriptions which have semantic information about the items. The description of an item is formed to calculate the similarity between items and user profiles. Similarly, user profiles are formed using the descriptions of users' previous item preferences.

Some of the main issues concerning CBF techniques are constrained content analysis, the new user problem and overspecialization [4]. Content-based recommender systems usually examine the characteristics of items that were automatically derived by information recovery techniques. However, there are complicated algorithms to tokenize textual documents, while methods of feature extraction can be much more difficult to use for multimedia data or items that have various/heterogeneous characteristics. Moreover, content-based recommender systems are likely to be overspecialized, since the only items that have a high likeness to those already rated will be recommended to the specific user. An additional issue of content-based recommender systems is that, firstly, a user has to rate an adequate number of items, then the system can predict recommendations.

1.2. Collaborative Filtering Methods

CF methods can recommend items to the users based on similar users' interests or habits, without any need for content information about items. The recommender systems that use CF approaches are presented in [5]. These approaches assume that users who previously have the same idea, with regards to them, former item preferences are likely to agree on the future items again. The users' ratings on the same item are computed first, then similar users are determined and recommendation generation are made depend on similar users [5]. Hence, CF techniques [6] depend on the ratings provided by users with

similar likes and choices. On the contrary CBF recommender systems, predictions of CF recommender systems depend on items formerly rated by other users, [14,16].

Nevertheless, the new user problem is encountered in collaborative recommender systems, and formerly, the system had to learn choices of the user to predict accurate recommendations. On the other hand, in addition to the new user cold start issue in collaborative recommender systems, they also pose the problem of the new item, which implies that a new item is needed to be rated by an adequate number of users before being suggested precisely by the system. Furthermore, the performance of a system of collaborative recommendation is dependent on the degree of accessible rating information. Generally, the number of ratings acquired is fewer than in comparison to the number of ratings that is needed to be predicted. That is to say, the user-item matrix is generally highly sparse, which accordingly causes inaccurate recommendations [4].

Many different CF algorithms have been proposed to overcome these difficulties that are generally categorized into three classes: memory-based, model-based, and hybrid schemes [7-9]. Memory-based algorithms representatively operate on the overall data collection to generate recommendations. On the other hand, model-based schemes work on a prototype derived from the original user-item interaction matrix. For example, the model based on the user-item interaction graphs is aimed to improve recommendation accuracy [10–12]. Also, the hybrid schemes are constructed to combine the advantages of both memory- and model-based techniques.

In any case, the two filtering methods exhibit particular deficiencies. Firstly, a content-boosted CF algorithm is proposed to improve recommendation accuracy [13]. In addition to CF and CBF systems, hybrid schemes combining CBF along with CF are proposed and these approaches are also utilized product contents, [14, 15]. Then, the hybrid schemes are constructed to combine the advantages of both CF and CBF techniques, [14]. In spite of combining CBF with CF, techniques enhance recommendation accuracy. Such hybrid systems require a complex implementation.

1.3. Hybrid Recommendation Algorithms

Until now, various recommendation algorithms have been proposed along with their advantages and disadvantages. Incorporating richer information sources beyond

numerical ratings can improve recommendation accuracy/overall performance of recommender systems [14, 17]. Recommendation approaches, which utilize different information sources are proposed [14,17]. These studies lead to research on hybrid recommendation techniques. Then, the hybrid schemes are constructed to combine the advantages of both CF and CBF techniques. The category of hybrid recommender systems [14,17], which generally includes two explorations, are the hybridization of algorithms, and the hybridization of different information sources. In the beginning, hybrid recommendation is mostly referred to as the utilization of both CBF [3] and CF-based [20] techniques for a recommendation. Also, different algorithms can be assembled by various strategies such as weighting, switching, mixing, cascading, or meta-level hybridization [14, 21]. However, these approaches put less attention on leveraging various information sources. They also require significant efforts on model design and selection, since different strategies are needed for different algorithms, especially when CBF methods are involved.

More recently, many Web applications have accumulated a large number of various information sources about users, items, and their interactions, which helped to extend the concept of hybrid recommendation to the integration of different information sources. A popular research line is the joint modeling of numerical ratings and textual reviews for recommendation [22, 23]. Beyond numerical ratings and textual reviews, visual images exhibit users' visual preference towards different items and these information sources are combined to construct recommendation [22]. An image-based recommendation system is proposed to provide co-purchase recommendations with the top-N recommendation methodology [24, 25].

Though achieving better performance against modeling ratings alone, previous models (including deep approaches) are usually limited to pre-selected information sources or domain knowledge, hence different models for different types of user-item interactions are developed. For instance, a user-item interaction graph model is proposed to improve recommendation accuracy [10–12]. This model is able to improve top-N recommendation performance, which is closely related to the business values in real-world recommender systems.

1.4. Contribution of Dissertation

Most of the existing CF-based recommendation systems operate with various input data such as reviews, ratings, or images to profile the users for a personalized recommendation. Despite its success two major challenge can be defined as “sparsity problem” and “integration of multiple types of input data” for recommender systems. The first challenge is the sparsity of the user-item interaction/transaction data (will described in next chapter), this data is the most crucial input data for recommendation systems. Many existing algorithms can not solve this sparsity problem and fail to make appropriate/accurate recommendations when adding new users and new items to the system. The further discussion on the sparsity problem of CF algorithms are investigated in next chapter. Since the major motivations to proposed the graph-based recommendation algorithm is the sparsity problem. Moreover, this dissertation also examines the usage of graph-based models to design more effective recommendation algorithms that cope with the sparsity nature of the transaction data in the real-time applications. The second challenge is the lack of a unified framework to integrate different types of input data and a meta-level algorithm evaluation to determine the most applicable algorithm to make recommendation. The previous approaches cannot work with the explicit relationship between different knowledge that we know about users and items. In this dissertation, answer of this question is investigated, “can we improve the performance of CF techniques by integrating user’s large-scale structured behavior data?”. The major challenge to answer this question is how to effectively integrate multiple types of user behaviors and item properties while preserving the internal relationship between them to improve the performance of estimating personalized recommendation.

A new graph based hybrid framework for personalized recommendation is described in this dissertation. Firstly, a simple overview of the framework is provided and then two different information sources (visual image, and numerical rating) is adopted to describe how the framework can be developed in practice. Moreover, the recommendation generation in a graph based recommender system can be moderated as a sub-problem of link prediction, that is a primary issue attempting to predict the probability of occurrence of a connection between two nodes depending on the discovered features and other connections between nodes [26, 27]. In a framework for predicting

links, there are symmetrical nodes which ignore the classification of nodes as a user and an item. These graph-based models have particular nodes (items and users) and three categories of links (item-item, user-item, and user-user) based on varying endpoint combinations. In the current literature, the type of user-user or item-item links is described as *similar* or *dissimilar*, and the type of links between users and items is described as *like* or *dislike* [28, 29]. After such adjustment, it is much more appealing to project links of *like* or *dislike* since only items are suggested to users.

In this dissertation, in order to utilize relational dualities, the proposed model is formulated to depend on the representation of complex (also quaternion) numbers with real and imaginary parts in the complex form. In previous studies, *similar* or *dissimilar* links were weighted by real numbers, whereas *like* or *dislike* links were weighted by complex numbers [29]. Since a complex number provides a natural algebraic link between real and imaginary values, the problem of recommendation could be considered as a problem of link prediction. With the utilization of the proposed method, other available algorithms of predicting links can still be used by no means of change. The proposed representation copes with sparse nature of transaction data and the lack of integration of the multiple type of data. The validity and efficiency of the proposed representation are assessed by evaluating the performance of the proposed recommendation approach in two real-world datasets.

1.5. Dissertation Organization

Recommender systems provide recommendations about various products and services to their users while using other user's data. Their success is imperative for both users and e-commerce sites utilizing such systems. Providing accurate and dependable recommendations increases user satisfaction that results in selling more products and services. On the other hand, inaccurate and unreliable product recommendations make users search alternative cites for shopping. In this dissertation, graph-based hybrid recommendation algorithms have developed for recommender systems to make accurate and reliable recommendations. To plainly explain each proposed method, the rest of the dissertation is organized as follows.

In chapter 1, the existing theories for recommender systems have presented. The idea under the traditional studies are given and the improvements on previous algorithms

are touched by displaying the characteristics of recent methods. Currently, the primary recommendation methods are explained together with different recommender system models.

Chapter 2 presents a literature review of the recommendation algorithm research and top-N recommendation task employed in the experiments in the following chapters. In this chapter, we have provided further discussion on the challenges of recommendation algorithms.

Chapter 3 introduces background information related to the proposed recommendation approach. The reasons for developing such methods are also touched with more detail. Moreover, the contributions of graph-based recommender systems are expressed by concentrating on their theoretical aspects. Furthermore, components of graph-based systems including bipartite graphs, link prediction methods, triadic closure model, and similarity measurements are explained in the related places.

Chapter 4 explains the detailed representation of the proposed recommendation algorithm. A similarity inclusive link prediction approach is proposed for item recommendation in the complex domain. Moreover, the potential impacts of each (user-user, item-item) links for graph-based recommendation systems are exhibited throughout the subjective and objective performance evaluation stages. Furthermore, the information about utilized similarity metrics for performance evaluation is given for readers.

The proposed recommendation algorithm is implemented by using different resources in chapter 5. Additionally, the potential impacts of each visual feature of items are used as links in the proposed graph-based recommendation systems. Chapter 5 experimentally examines the implementation of the proposed recommendation approach on Amazon (real-world) datasets and provides a discussion on the experimental results.

In chapter 6, the similarity inclusive link prediction approach for item recommendation is expanded and applied in quaternion domain. Furthermore, the triangle closing model is implemented/expanded in quaternion domain. Hence, potential impacts of this model for graph-based recommendation systems are exhibited. Chapter 6 experimentally analyses the proposed recommendation approach in real-world datasets that are used in chapter 4-5 and provides a discussion on the experimental results.

Finally, the dissertation has finalized with a concise conclusion and summarized obtained results and conclude on the contributions of the proposed algorithm.

2. RELATED WORK

The input data of a recommender system can be categorized as in three forms according to relevance of users or items and transactions between users and items [74]. The users related data can be sampled as users' name, gender, and address, etc. The items related data can be sampled as items' price, images and product brand, etc. The transactional data are captured through users' explicit ratings or implicit feedback observed from users' behaviors. Transactional data can be sampled as to which item was purchased, rated, and wanted by a user. In the literature, many recommender systems have employed users' explicit feedback in the form of ratings [15, 61, 62]. For example, user ratings of a webpage are implemented as input data in a recommender system [61], and also user ratings on movies are employed as input to another recommendation system [62]. These systems that utilize explicit feedback from users as input data, are based on users' significant contribution. Hence the explicit feedback could be costly for users, a good system design is needed to reveal the feedback from the users initially to gather the important information as well as on a continued basis to ensure good recommendation performance on newly added items.

Moreover some systems have utilized implicit feedback automatically captured during the user's usage session to deal with this problem [63-65]. These systems analyze system logs or transaction records to reveal users' preferences. The explicit feedback data is generally in a form of ratings, ranging from 0 to 5 star rating schemes. However, the implicit feedback data are demonstrated by binary values in literature. Such as, sales transactions data can be represented in binary values as an implicit feedback form. Hence, the occurrence of sale transaction can be indicated by 1 and the non-observation of sale transaction can be indicated by 0. On the other hand, the binary input data are modified as a special case of the multi-scale rating data in many existing recommendation algorithms, there is fundamental difference for these algorithms design. For example, the rating data that utilized in these recommendation algorithms can be indicated by three rating values for each user-item pair, +1, −1, and 0. The rating of *like* is indicated by (+1), *dislike* is indicated by (−1) and non-observation of rating is represented by 0. These recommendation algorithms take the observations with +1 and −1 rating as training data

to derive models for prediction of potential values of test data or unobserved ratings. The only information regarding a user-item pair for recommendation with binary transaction data is whether a transaction has been observed (1) or not (0). Likewise, recommendation algorithms that utilized typical implicit feedback data is only benefited from positive instances to generate a recommendation. Hence, typical learning algorithms are not applicable for these recommendation algorithms. Indeed, these lacks/differences are fundamental motivations to proposed the graph-based approaches for recommendation systems in this dissertation. The problem of the transaction-based recommendation generation is the prediction of the positive instances depend only on the previously observed positive instances since there are no negative instances. This problem is deeply connected with the link prediction problems when representing the user item interactions utilizing by a graph representation/model [29, 30, 66]. This representation have utilized in many places in the following chapters.

Different data representations are utilized in many existing recommender systems. These representations are denoted as in three forms in a recommender system and they are user representation, item representation, and transaction representation. User representation can be constituted in four different ways utilized by user attributes, or by associated items or by transactions and also by item attributes associated with the user preferences in the feedback data. User attributes may usually represented as demographic data such as gender, birth date, salary, etc., Associated items can be denoted as the products the user has expressed interest in, has given ratings to or actually purchased/bought. Transactions can be denoted as attributes extracted from the user's transaction history such as time, frequency, and amount, can partially represent a user's behavior pattern. Items or item attributes associated with the user preferences in the feedback data can be explained that a user might be characterized as liking romantic movies and depend on the attributes of the movies that user has already bought/watched or simply as a set of movies that user has watched/bought. Most recommendation algorithms denote users as a set of associated items [15, 61, 62, 64, 67]. Such as, a recommender system that utilized web pages the user had rated or previously visited to represent the user [64]. Likewise, a recommendation algorithm that used in GroupLens [67] utilized movies and books to represent the user interests. Besides that, another important factor in recommendation algorithms is item representation. Items are usually

represented by item attributes they are price, content, brand, etc. or by associated users who have rated/bought this item before. There are also several recommendation algorithms that utilize the user-item interaction matrix and do not explicitly benefit from any user or item representations [29, 30, 66]. Several recommender systems also utilize transactions as a recommendation generation factor, and transaction attributes (such as time, amount, etc.) or items in the transactions are utilized to representing them, [12, 23, 24]. Some transaction attributes are included such as time and place as additional dimensions and support a different type of recommendation that may be depend on different combinations of dimensions. For example the recommendation of web content is generated to a specific user on weekends, or the recommendation of the best time to promote certain products to a specific user is determined [4]. Whereas, the most existing recommendation algorithms concentrate on the analysis of the two dimensions of users and items.

Users are typically represented by the items they have bought or rated in the CF systems. Such as, in an online shopping sites selling 1 million products, each user is represented by a Boolean user-item vector of 1 million components. The value for each component is determined by whether a particular user has bought the corresponding item in past transactions. When a user bought an item typically the value is denoted as 1 and if this user did not buy or did not see the item, the component of the user-item vector is denoted as 0. A matrix consists of all user-item vectors that are representation of all users can constituted with capturing past transactions. This matrix is called as user-item interaction matrix and is briefly described in the next chapter. The general term “interaction” is used to refer to this matrix as opposed to the more particular “purchasing” or “transaction” since there are other types of relations such as explicit and implicit ratings between users and items for general recommender systems, [74].

In many large-scale implementations for instance major e-commerce sites, the number of items, and the number of users are both large. Many transactions have been recorded in that cases, hence the user-item interaction matrix can be extremely sparse, in other words, there are very few components in this matrix is denoted as 1. The sparse user-item interaction matrix has a major negative impact on the effectiveness of a CF recommendation approach, this negative impact is commonly referred to as the sparsity problem. The similarity value between two given users or items is zero because of the

sparsity, hence that makes CF approaches as useless [61]. Even for pairs of users or items that are really similar, such similarity values may not be reliable.

To counter this major sparsity problem, a new graph-theoretic approach based on CF algorithm is firstly proposed in [68]. The main idea of this approach is to form and maintain a directed graph whose nodes are only users and whose directed edges correspond to a new concept referred as predictability. This predictability term is in one sense stronger than the various concept of closeness, and since in another sense more generic hence this concept more likely to occur. This concept is stronger when more common items are already rated by the source and target users. It is more generic in the sense that graph consist not only pairs of users whose ratings are really close, but also user pairs whose ratings are similar except that a specific user's ratings are more same with a user's ratings than another user [68]. This concept also contains user pairs who rate items more or less oppositely, additionally combinations of the above two scenarios. A linear transformation is generated to translates ratings from a user to the other, whenever a user predicts another user. The principle idea is that estimated rating of item j for user u can be computed as weighted averages computed via a few reasonably short directed paths joining several users. These directed paths have connected user u at one end with another user k that has rated item j at the other end. Along the directed path other users have not rated item j. Whereas, each directed arc in the graph is provided the property that there is some real rating predictability from one user to the other. Then, the overall rating for this path is evaluated by starting with the rating r_{kj} and translating it via the combinations of the different transformations corresponding to the directed path [68].

Moreover, the cold-start problem demonstrates the importance of addressing the sparsity problem. The cold-start problem can be explained with the situation when a new user or item has just entered the system [69]. CF cannot constitute useful recommendations for the new user because of the lack of sufficient previous ratings or bought information. Similarly, CF systems will not recommend new item to many users because very few users have yet rated or bought this item that has just entered the system. Generally, the cold-start problem can be considered as an illustration of the sparsity problem, since the most components in certain rows or columns of the user-item

interaction matrix ($\boldsymbol{A}$) for the new user or item are indicated as 0. Some recommendation algorithms have proposed to cope with this such sparsity and cold-start problem. For instance, an item-based recommendation approach proposed to alleviating both the sparsity and scalability problems [18]. Firstly items which are similar as a particular user's bought before are examined depend on the transactional data, and then recommended to the user. Item similarities are computed to find relationship between the corresponding item (column) vectors in the user-item interaction matrix. It is observed that this item-based approach improve the quality of recommendation compared to the user-based approach which depends on relationship between user (row) vectors, [18]. Moreover another recommendation approach based on dimensionality reduction is proposed to deal with sparsity problem. This approach aims to reduced the dimensionality of the user-item interaction matrix directly. Firstly items or users are clustered and then these clusters are used as basic units to generate recommendations. Also, the commonly known techniques can be implemented to reduced the dimensionality. For example statistical techniques such as Principle Component Analysis (PCA), [65] and information retrieval techniques such as Latent Semantic Indexing (LSI) are utilized to reduced dimensionality in recommender systems [61,18]. The experimental results of this studies demonstrate that dimensionality reduction can improve recommendation quality significantly in some applications, but performs poorly when compared the others [18]. The dimensionality reduction approach are proposed to solve the sparsity problem by removing insignificant or unrepresentative users or items to generate the compact user-item interaction matrix. Unlike, potentially useful information can be lost during this reduction process.

The combination of CF with CBF recommendation approaches are proposed to overcome the sparsity problem in some researches [61,62, 66]. These approaches are take into account past user-item interactions and also similarities between users or items directly derived from their intrinsic properties or attributes. These approaches are referred as the hybrid recommendation approach. The hybrid approaches that utilized in previous studies have improved the quality of recommendation compared to the user-based recommendation approaches. Whereas, the hybrid approach needs additional information regarding the items and utilize a measurement to compute meaningful similarities among

them. Obtainin such item representation can be difficult or expensive and a related similarity measurement can not be readily available.

The general recommendation approach in this dissertation are proposed as a different graph-based framework and same as in [30,66] to deals with the sparsity problem. Instead of reducing the dimension of the user-item interaction matrix $\boldsymbol{A}$, since this method makes $\boldsymbol{A}$ less sparse, the graph based models are investigeted. The transitive interactions between users and items are explored to generate the matrix $\boldsymbol{A}$ and make it meaningfully "dense" for recommendation purposes. The common sense behind transitive interactions can be explained by the following example. Suppose that movie i_1 is watched by users u_1 and u_2 and movie i_2 is watched by users u_2 and u_3. In the standard CF approaches can associate u_1 with u_2 and also u_2 with u_3 but can not take into account the transitive interactions u_1 with u_3. Besides that an approach that utilizes transitive interactions, can reach the associative relationship between u_1 and u_3 and can embed such transitive interactions into the user-item interaction matrix $\boldsymbol{A}$ to generate recommendations.

The hybrid approaches can yield/obtain better performances than CBF, demographic filtering (DF), and CF approaches, these results are demonstrated in several studies. Whereas, the existing hybrid models are typically heuristic in nature, combining the different types of input data in a specific manner. Such as, the simplest hybrid approach can be considered as the combination of the recommendation results from different approaches. The multiple recommendation lists can be combined by a specific technique, such as, top-N recommendations from different approaches can be combined. Another hybrid algorithm that based on agents are designed to act like a regular user in a CF system to give ratings to items depend on the content information [66]. Also several arbitrary system design and arbitrary parameter settings are embeded, while hybrid approach combines the CBF and CF approaches for recommendation in principle. Since, many previous hybrid recommendation approaches are in fact heuristic-based algorithms. The only exception is probably the algorithms that adopt the generative model approach, in which different types of input data are modeled systematically. Whereas, the performance of these algorithms have not been widely implemented and compared with other approaches. Also, data representation of this algorithm has some limitations to

implement in different application domains. Likewise these models involves the sparse transaction data problem and are also not clear. A unified recommendation framework with the expressiveness for representing multiple types of input data is proposed to improve the recommendation performances and a generic computing mechanism to integrate different recommendation approaches is necessary to utilize the available rich information, [23]. The graph-based recommendation structure is generated to represent different types of input data and to integrate multiple recommendation approaches as a hybrid algorithm is discussed in Chapters 5 and 6 of this dissertation.

Hence, there are also the lack of a meta-level framework for model evaluation and selection and the lack of a unified integration framework in recommender systems. Recommendation algorithms are generally evaluated with a hold-out testing method using transaction or ratings in the datasets. The algorithms are evaluated depend on its recommendation performance with particular training and testing datasets. These evaluation studies also have the lack of a general "double hypothesis" limitation [74]. Since a recommendation algorithm have two components: (1) the major assumption regarding the interaction data generation process; (2) the parameter settings and algorithmic applications that work with a major model to generate recommendations. The major assumption of the user based CF assumes that users replicate the behavior of other users with past similar experiences; the generative model assumes that user-item interactions are generated depend on the hidden types of the users and items. The first component yields much more valuable and potentially generalizable findings, existing evaluation studies can not alternate the effects of these two components. A formal framework is necessary to understand the fundamental mechanisms that manage the interaction data generation process in different application domains. The various fundamental approaches generated by different classes of recommendation algorithms can be verified depend on such a hybrid framework [74]. Such a meta-level recommendation model evaluation and selection with utilizing the recent techniques in graph modeling methodology is presented in Chapter 5 of this dissertation.

Moreover the commonly known testing methodology for evaluating the performance of recommendation algorithms are described in this chapter. An ideal evaluation of a recommendation can involve assessment given by users that has already experienced the recommended products or services, however this type of evaluation is

typically difficult and excessively expensive. In the current literature, a hold-out test procedure is commonly employed to evaluate the performance of recommendation approaches. The aim is to separate a portion of the transaction data as the testing set and utilize the remaining transactions to generate recommendations and these transactions are referred as the training set. Then the generated recommendations are compared with the actual transactions in the testing set to evaluate performance measures. In the previous recommendation algorithms, the portion (e.g. %10) of each user's latest interactions are selected as the testing set and the remaining interactions are indicate as the training set [29]. The approaches that utilize the top-N recommendation task, in which a ranked list of N items is recommended for each user. The recommendation accuracy is evaluated based on recommendations that matched the items in the testing set for each user, it gives the number of hits and their positions in the recommendation list. The literature of the following recommendation quality measure are adopted regarding the coverage, relevance, and ranking quality of the ranked list recommendation for a particular user, [7]. These quality measure has particular importance in real-life systems since users are take only into account a small set of top-N recommendations. Hence, it is important to make sure that this set is as interesting and attractive as possible. Moreover the well known measures that evaluate the performance of recommender systems within the context of accuracy and diversity, serendipity, coverage and novelty are analyzed in [70].

3. PRELIMINARIES

In this chapter, the general background and preliminaries on graph-based recommender systems and the components of graph-based systems are explained. Link prediction techniques for graph-based recommendation systems are described and prediction estimation algorithms for these systems are defined. Afterwards, global and local link prediction techniques are discussed. Finally, the previous graph-based recommendation algorithm, that is used in the experiments is described.

3.1. Graph-Based Recommender Systems

In real-world e-commerce applications, there generally exists more transaction information than explicit rating information [12]. Hence, CBF and CF methods are implemented together to build hybrid schemes for recommendation systems. These schemes are focused on the modeling and prediction of transactions. Modeling items and users in a graph structure is a better way to apply CBF and CF approaches in one framework. In this structure, the transactions T can be represented as a user-object interaction graph $G=(V,E)$. This graph can also be called as a bipartite graph. A bipartite graph is a special graph whose vertices can be divided into two independent sets, U and O such that every transaction/edge (u,o) either connects a vertex from U to O or a vertex from O to U. In other words, for every transaction/edge (u,o), either u belongs to U and o to O. Furthermore, it can be observed that there is no edge that connects vertices of same set.

Two node types exist in a user-object interaction graph as object and users, when vertices V and edges E are represented as in Eq. 3.1;

$$V=U\cup O,\ E=\{(u,o):u\in U,o\in O,u\rightarrow o\in T\} \tag{3.1}$$

where U is denoted as the set of users and O is denoted as the set of objects/items. That corresponds to a representation of the bipartite graph since user nodes can connect only to item nodes and vice versa. An example of a bipartite graph is illustrated in Figure 3.1. The objects are illustrated as movies and the user-object interactions are illustrated as ratings in Figure 3.1. The ratings (range from 0 to 5-star) of users are illustrated in Figure

3.1 (a), and the graph model of interactions are drawn in Figure 3.1 (b). The illustration of Figure 3.1. is inspired from the figure which is drawn in [53].

The recommendation in a bipartite graph may be moderated as a sub-problem of link prediction with this representation. The graph representation shows subtle relations between indirectly connected users and items, which is a primary issue attempting to predict the probability of occurrence of a connection between two nodes depending on the discovered features and other connections between nodes [26, 27]. The bipartite user–item interaction graph can also be projected onto a unipartite user/item graph to simplify the graph model [11,12]. Additionally, some graph models are based on user/item similarities, such artificial graphs may help to solve the data sparsity problem [8].

Figure 3.1. *An illustration of a bipartite graph, a) objects and interactions, b) the bipartite graph model.*

Several CF heuristic algorithms have examined the structure of user-item interaction graphs to enhance recommendation performance [10-12]. For example, two-layer graph model in the context of book recommendation is described in [32], where the authors propose a graph-based recommendation approach to integrate the CBF approach along with CF approach in the context of digital libraries by representing books and users as nodes. Moreover, learning-based algorithms utilize graphs in building effective personalized recommendation models. The learning-based recommendation algorithms usually rely on explicit feature extraction, which is difficult to apply to graph-structured data due to the requirements of computational capacity and to design features [33]. With another approachment, a generic kernel-based machine learning approach of link prediction in bipartite graphs is applied to improve the performance of recommender systems [12]. They propose a novel graph kernel to capture the structure and user/item

features in the context of user-item pairs and this model utilizes the one-class support vector machine algorithm to construct the predictions.

The classification of nodes as the subject (user) and the object (item) can be disregarded by utilizing symmetrical nodes for predicting links. Since user-item interaction graphs can also be defined as an adjacency matrix with nodes of users and items which can be represented as a bipartite graph. These graphs have two types of nodes (users and items) and three categories of links (item-item, user-item, and user-user) depending on varying endpoint combinations. Presently, the type of links between users and items is labelled as like or dislike and the type of user-user or item-item links is labelled to be similar or dissimilar, [28, 29]. Following such adjustment, it is much more appealing to project links of like or dislike since only items are suggested to users.

3.2. Link Prediction

Links can be described as relationships between subjects and objects in various applications such as social networks, recommender systems, and web analysis [34-36]. Hence, along with enhancements in the Web and social networks, social network analysis and link prediction issues are getting more popular in recent years. Link prediction is defined as a task to reveal the presence, absence or strength of a link between two entities based on properties of the objects and other observed links [34]. Link prediction algorithms, that is used in web mining applications are implemented with graph models to predict the links. Hence, graph models are getting more attention to solve real-world problems in recent works [66]. Social network analysis methods are applied to these models, and the inferences of these methods are influenced from properties of the actors corresponding to the node in the graph and the situation of inter-actor relations [34]. The recommendation approach which is modeled as a social graph corresponds to predicting a link/relationship between a subject and an object [28, 37].

3.2.1. Link Prediction Techniques for Recommendation Systems

The recommendation generation problem can transform to find future links for each user node, thus this problem can be defined as a link prediction problem [29]. Recommendation algorithms can be modeled with a user-item interaction/bipartite graph, with specific nodes (users and items) and links (similar relations among users/items, and

interactions between users and items). Therefore, recommendation generation in a user-item interaction graph can be considered as a sub-problem of link prediction. Link prediction methods operate to predict the probability of a connection between a node and another depending on the observed features and connections of nodes [26, 27]. Hence, the symmetrical nodes in the user-item interaction graph can be used for predicting links. Since the symmetrical and square adjacency matrices with an equal number of user and item nodes are used to represent user-item interaction graphs, otherwise, the biadjacency matrices are used to denote bipartite graphs. The graph models in recommender systems have two types of nodes, that are named by items and users. The links in this model can be categorized as user-user, user-item, and item-item with different endpoint combinations. Presently, the links between users and items give information about users' preferences, and the user-user or item-item links give information about user-user or item-item relationships [28, 29]. Since, the recommendation generation is based on the predicting links, that have information about users' preferences, in the graph-based models.

3.2.2. Link Prediction Methods

A good link prediction function should give a higher score when there are more paths connecting two nodes and give a higher score when paths are shorter. A useful technique to solve the problem of link prediction is to describe a network in the form of a matrix where link prediction values are calculated by processing such a matrix. Algebraic graph theory utilize the adjacency matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{ij} = 1$ when (i, j) is an edge and $\boldsymbol{A}_{ij} = 0$ otherwise. For undirected networks, generally, the adjacency matrix $\boldsymbol{A}$ is symmetrical, and its eigenvalue decomposition may be considered as $\boldsymbol{A} = \boldsymbol{U}\Lambda\boldsymbol{U}^T$, where $\boldsymbol{U}$ is an orthogonal matrix and Λ is a diagonal matrix. The logic behind usually considering the adjacency matrix's eigenvalue decomposition is that it is possible to calculate the power of the matrix as;

$$\boldsymbol{A}^k = \boldsymbol{U}\Lambda^k\boldsymbol{U}^T, \tag{3.2}$$

which may be used for expressing link prediction methods like the Neumann kernel, the matrix exponential, triangle closing, and rank reduction. The theory of Eq. (3.1) is given in the Appendix as theorem 1, [31]. The other decomposition methods like probabilistic

latent semantic analysis or non-negative matrix factorization do not have the useful characteristics/features [28].

The most popular and classical link prediction methods are polynomial functions, rank reduction methods, Neumann kernels and hyperbolic sine function e.g.

Polynomials; Any polynomial with only odd powers and nonnegative weights can be used as a recommendation generation algorithm:

$$P(\boldsymbol{A}) = a \cdot \boldsymbol{A} + b \cdot \boldsymbol{A}^3 + c \cdot \boldsymbol{A}^5 + d \cdot \boldsymbol{A}^7 + \ldots \tag{3.3}$$

Rank Reduction; Rank reduction consists of finding a matrix with maximal $rank - r$ which is nearest to the given adjacency matrix. In the resulting matrix, entries of zero which denote unconnected node pairs are assigned nonzero values, which can be used as a recommendation score. The reduction to $rank - r$ of the matrix $\boldsymbol{A}$ can be computed from the singular value decomposition of $\boldsymbol{A}$ by keeping the r largest singular values and changing all other singular values to zero. Since the singular value decomposition $\boldsymbol{A}$ is related to the eigenvalue decomposition of $\boldsymbol{A}_b$, it follows that the best $rank - r$ approximation to $\boldsymbol{A}_b$ is given by the truncation of $\sum$.

Hyperbolic sine; As shown as an Eq. (2.3) above, the odd component of the matrix exponential gives the matrix hyperbolic sine:

$$sinh(\boldsymbol{A}) = \boldsymbol{A} + (1/6) \cdot \boldsymbol{A}^3 + (1/120) \cdot \boldsymbol{A}^5 + \ldots \tag{3.4}$$

Newman kernel; The Newman kernel is given by the geometric (or Newman) series; $(\boldsymbol{I} - \alpha \boldsymbol{A})^{-1} = \boldsymbol{I} + \alpha \boldsymbol{A} + \alpha^2 \cdot \boldsymbol{A}^2 + \alpha^3 \cdot \boldsymbol{A}^3 + \ldots$ in which the constant α must be smaller than the inverse of the largest singular value of $\boldsymbol{A}$. The restriction of the Newman kernel to only odd powers results in the odd Newman kernel;

$$\alpha \boldsymbol{A}(\boldsymbol{I} - \alpha^2 \boldsymbol{A}^2)^{-1} = \boldsymbol{I} + \alpha \boldsymbol{A} + \alpha^3 \cdot \boldsymbol{A}^3 + \alpha^5 \cdot \boldsymbol{A}^5 + \ldots . \tag{3.5}$$

Due to the equivalence with the bipartite case, these odd kernels were previously used for link prediction in bipartite networks/graphs [28].

3.2.3. Graph Notation

In the typical link prediction approach based recommendation scenario, the input data are modeled as a directed graph $G=(V,E,\omega)$, comprises of vertices connected by edges. Vertices, V, in a directed network are defined as the nodes being items and users, while edges, E, represent links between the nodes, i.e., ratings and ω contains all links' weights. Let U is the set of users, and I is the set of items, respectively. Then, V is the union of all users and items, $(V=U\cup I)$, and E is the link set of nodes $(E=U\times U\cup U\times I\cup I\times I)$. Moreover, the notation of any path is represented as $(a_1, a_2,\ldots,a_{k+1})$, and the path length is denoted by k, whereas two endpoints are represented as a_1 and a_{k+1} connected by the inner nodes of $a_i\,(i=2,3,\ldots,k)$. Additionally, k links are observed along this path of $(a_i,a_{i+1})\in E$, where $i=1,2,\ldots,k$. When the path of length corresponds to one, where $k=1$, it means that there is a link to any inner nodes. Furthermore, we describe $N_u(i)$ as the set of items that user u rated and $N_i(u)$ as the set of users rated item I, where $N_u(i)=\{i\mid(u,i)\in E,i\in I\}$ and $\mathrm{N_i(u)}=\{\mathrm{u}\mid(\mathrm{u,i})\in\mathrm{E},\mathrm{u}\in\mathrm{U}\}$. If there is a connection between two nodes, there are always two links that connect this node-pair, one in each direction. Then, it is possible to reduce the recommendation effort to predict whether there will be a link in the graph between a user and a specific item. A prediction which shows the extent of the relevance of any item to a particular user is calculated by using an algorithm of link prediction in graph-based recommender systems [28].

3.2.4. Triangle Closing Model

Nodes in a user-item bipartite graph may have two types of relationships. First of all, for both user-user and item-item links, there is a similarity factor, $\omega_{similar}$, between two entities. Then, including user-item links and item-user links, there is the preference, ω_{like} and $-\omega_{like}$, of the user on an item, due to the necessity of recognizing the asymmetry between the user and the item. Accordingly, in the case of a link from the user u to item i with the weight ω_{like}, there is always a reverse link from item i to the user u with a

weight of $-\omega_{like}$. In this model, ω_{like} and $\omega_{similar}$ are normalized values just for the weights. The triangle closing rule in this model may be described as shown in Figure 3.2.

This rule has two parts: users who have denoted the same interest in shared items may be similar (see Figure 3.2 (a)), similar users will be similarly interested in the same item (see Figure 3.2 (b)), and user similarity is transitive among users (see Figure 3.2 (c)). Likewise, items liked by associated users may be similar (see Figure 3.2 (d)), users are prone to interest in similar items (see Figure 3.2 (e)), and besides that item similarity is transitive among items (see Figure 3.2 (f)). These are the main ideas of CF from a different viewpoint, [28]. Thus, these principles may be mathematically stated as follows:

$$\begin{cases} \omega_{similar} = -\omega^2_{like}, \\ \omega_{like} = \omega_{similar} \times \omega_{like}, \\ \omega_{similar} = \omega^2_{similar} \end{cases} . \tag{3.6}$$

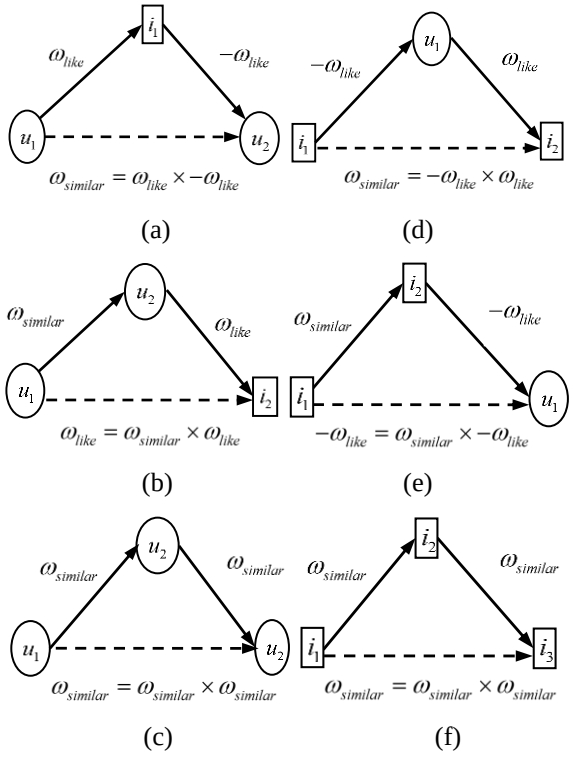


Figure 3.2. *The triangle closing principle is illustrated as the multiplication rule between similar and like relationships, [29].*

Therefore, to solve this system of equations (Eq. 3.6), we need to find two different and nonzero constants, which are $\omega_{similar}$ and ω_{like}. Complex numbers offer an

easy way to solve this system of equations, if we set ω_{like} and $\omega_{similar} = 1$, where j is the imaginary unit. The requirements may be mathematically stated as follows, $1 = -j^2$; $j = 1 \times j$ and $1 = 1^2$.

The corresponding multiplication rules for dislike and dissimilar may then obtained by multiplying both sides by -1. In another situation, where a user dislikes $(-j)$ an item that is dissimilar (-1) to the one that they are interested in (j) may be expressed as the following equation: $-j = j \times (-1)$. In this symbolization, a link has endpoints of the same type, two items or two users, must be weighted with a real number: the higher such value, the more similar the endpoints.

On the contrary, a link with an imaginary weight must be an item-user or user-item link based on the sign and interest. For instance, if a user u dislikes item i, then the link is weighted with $-j$ from u to i, and the other link is weighted with j from i to u. As opposed to similar links, we may distinguish the dislike and like only when the sign of link's weight and the direction of the link are known at the same time. On the other hand, the value of the weight might define the degree of dislike or like.

3.2.5. Adjacency Matrix

The adjacency matrix is described as $\boldsymbol{A} \in R^{|V| \times |V|}$ given by $\boldsymbol{A}(u,i) = \begin{cases} 1 \; if \; (u,i) \in E \\ 0 \; if \; (u,i) \notin E \end{cases}$ when $G = (V, E)$ is denoted as an undirected and unweighted network. The adjacency matrix $\boldsymbol{A}$ is symmetric and square. Therefore, it is possible to derive the number of paths connecting two nodes by calculating the powers of the matrices in unweighted networks. Additionally, it is possible to formulate the number of common neighbors between two nodes u and i $(u, i \in V)$ by taking the square of the adjacency matrix:

$$N(u,i) = \boldsymbol{A}^2(u,i), \tag{3.7}$$

which applies basic triangle closing and may be explained as the number of paths with a length of two among them. This formulization has a significant characteristic: as big as the entry of the square of the adjacency matrix is, these two nodes will be closer. At the same time, the number of paths of any length k from node u to node i can be expressed by the components of $\boldsymbol{A}^k(u,i)$. Therefore, the closeness of the two nodes may be

calculated by the weighted sum of powers of the adjacency matrix $\boldsymbol{A}$. Such an example of a link prediction method to unite these results is the matrix exponential:

$$\exp(\boldsymbol{A}) = \boldsymbol{I} + \boldsymbol{A} + 1/2 \cdot \boldsymbol{A}^2 + \ldots \tag{3.8}$$

This function has two main contributions: it considers that all powers of $\boldsymbol{A}$ involve all paths between two nodes. Also, short paths are prioritized over long paths due to the decreasing weights of the powers. Then the real numbers are used to represent the user-user and item-item relationships, and the complex numbers are used to express the user-item interactions. The adjacency matrix $\boldsymbol{A}$ of the user-item graph G is defined as follows:

$$\boldsymbol{A}(u,i) = \begin{cases} 1 \text{ if } u \text{ similar } i \\ -1 \text{ if } u \text{ dissimilar } i \\ j \text{ if } u \text{ likes } i \text{ or } i \text{ dislikes } u \\ -j \text{ if } u \text{ dislikes } i \text{ or } i \text{ likes } u \\ 0 \text{ if } (u,i) \notin E \end{cases} \tag{3.9}$$

where $\boldsymbol{A}(u,i)$ is the value of row u and column i of the matrix $\boldsymbol{A}$. The matrix $\boldsymbol{A}$ may be conveniently represented as:

$$\begin{bmatrix} \boldsymbol{A}_{UU} & \boldsymbol{A}_{UI} \\ \boldsymbol{A}_{IU} & \boldsymbol{A}_{II} \end{bmatrix}, \tag{3.10}$$

where $\boldsymbol{A}_{II}$ and $\boldsymbol{A}_{UU}$ are the item-item and user-user similarity matrices, $\boldsymbol{A}_{UI}$ and $\boldsymbol{A}_{IU}$ are the user-item preference matrices, also the conjugate transpose of $\boldsymbol{A}_{IU}$ can be described as, $\boldsymbol{A}_{IU} = -\boldsymbol{A}^T{}_{UI}$. The preference matrices are complex matrices, while the similarity matrices are real matrices. In the Complex Representation based Link Prediction method (CORLP)[28], the authors ignore the relationships between users/items, they represent the bipartite graph as G and the adjacency matrix as $\boldsymbol{A}$, corresponding as;

$$\boldsymbol{A} = \begin{bmatrix} 0 & \boldsymbol{A}_{UI} \\ -\boldsymbol{A}^T{}_{UI} & 0 \end{bmatrix}. \tag{3.11}$$

Complying with the representation of the adjacency matrix $\boldsymbol{A}$, each entry in the preference matrix $\boldsymbol{A}_{UI}$ only has three different values: j, $-j$ and 0. Furthermore, $\boldsymbol{B}$, the

biadjaceny matrix of the bipartite graph corresponding to $\boldsymbol{A}$, is a real matrix. Then, $\boldsymbol{A}$ can be expressed as;

$$\begin{bmatrix} 0 & j\boldsymbol{B} \\ -j\boldsymbol{B}^T & 0 \end{bmatrix}. \tag{3.12}$$

Based on the path counting process in the unweighted and undirected networks, the weighted path counting process for paths of length k may be similarly derived by $\boldsymbol{A}^k$. If we only consider the relationships between users and items, the k^{th} power of the adjacency matrix may be further formulated mathematically as:

$$\boldsymbol{A}(u,i) = \begin{cases} \begin{bmatrix} \left(\boldsymbol{B}\boldsymbol{B}^T\right)^n & 0 \\ 0 & \left(\boldsymbol{B}^T\boldsymbol{B}\right)^n \end{bmatrix} where\ \ k = 2n \\ j \times \begin{bmatrix} 0 & \left(\boldsymbol{B}\boldsymbol{B}^T\right)^n \boldsymbol{B} \\ -\left(\boldsymbol{B}^T\boldsymbol{B}\right)^n \boldsymbol{B}^T & 0 \end{bmatrix} where\ \ k = 2n+1 \end{cases} \tag{3.13}$$

Thus, any sum of the powers of the adjacency matrix $\boldsymbol{A}$ may be divided into components that are even and odd, but only the odd components are effective for final recommendation. Hence, the predictions may be generally applied to $\boldsymbol{A}$ giving:

$$P(\boldsymbol{A}) = \lambda \cdot \boldsymbol{A} + \lambda_3 \cdot \boldsymbol{A}^3 + \lambda_5 \cdot \boldsymbol{A}^5 + \lambda_7 \cdot \boldsymbol{A}^7 + \lambda_9 \cdot \boldsymbol{A}^9 + \ldots . \tag{3.14}$$

To guarantee that shorter paths yield more to the predictions $\{\lambda_1, \lambda_2, \lambda_3, ...\}$ is a decreasingly weighted sequence. For instance, we can take the matrix exponential of $\boldsymbol{A}$, that is yielding;

$$\begin{aligned} e^{\boldsymbol{A}} &= \boldsymbol{I} + \boldsymbol{A} + \frac{1}{2}\boldsymbol{A}^2 + \frac{1}{6}\boldsymbol{A}^3 + ... \\ &= (\boldsymbol{I} + \frac{1}{2}\boldsymbol{A}^2 + ...) + (\boldsymbol{A} + \frac{1}{6}\boldsymbol{A}^3 + ...) \\ &= (\boldsymbol{I} + \frac{1}{2}\begin{bmatrix} \boldsymbol{B}\boldsymbol{B}^T & 0 \\ 0 & \boldsymbol{B}^T\boldsymbol{B} \end{bmatrix} + ...) + j \cdot \left(\begin{bmatrix} 0 & \boldsymbol{B} \\ -\boldsymbol{B}^T & 0 \end{bmatrix} + \frac{1}{6}\begin{bmatrix} 0 & \boldsymbol{B}\boldsymbol{B}^T\boldsymbol{B} \\ -\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{B}^T & 0 \end{bmatrix} + ... \right) \end{aligned} \tag{3.15}$$

It can be seen that the even part of the power sum can be used to measure the similarities among users or items, while the odd part can be used to find items of interest

to users. Therefore, only paths of odd length are suitable for a recommendation, [29]. The power sum can be to the odd part sum with using the matrix hyperbolic sine of $\boldsymbol{A}$:

$$sinh(\boldsymbol{A}) = \boldsymbol{A} + (1/6)\cdot \boldsymbol{A}^3 + (1/120)\cdot \boldsymbol{A}^5 + ... \tag{3.16}$$

4. A SIMILARITY INCLUSIVE LINK PREDICTION BASED RECOMMENDER SYSTEM APPROACH

4.1. Introduction

Recommender systems are generally categorized according to their approach to the prediction of ratings. In general, there exist two primary recommendation methods, i.e., CBF and CF methods. CBF recommendation algorithms are usually dependent on the similarity of items to the objects that are previously preferred/purchased by the user before [4]. However, CF techniques are based on the ratings provided by users with similar tastes and choices [6]. Predictions of CF recommender systems depend on items formerly rated by other users. Therefore, the performance of a system of CF recommendation is dependent on the degree of accessible rating information. In any case, these methods are exhibited particular deficiencies. Generally, the user-item preference matrix is highly sparse, which accordingly might lead to inaccurate recommendations [3]. Many different recommendation algorithms have been proposed to deal with these drawbacks. Hence, CBF and CF methods are applied together to generate a better/accurate recommendation systems. Modeling items and users as nodes in a graph structure is a natural way to utilize these two approaches in one framework [12]. Such as, a graph-based recommendation algorithm is proposed to integrate the CBF approach along with the CF approach in the context of digital libraries by representing books and users as nodes, [32]. This approach is described as a two-layer graph model in the context of book recommendation algorithm. Moreover, the models based on the user-item interaction graphs are aimed to improve recommendation accuracy [10-12]. Two node types exist in a user-item interaction graph as items and users.

Graph-based recommendation algorithms are formed in two fundamental steps; firstly, a graph model is constructed/build to represent the input data and then recommendations are generated by analyzing the graph. These recommendation algorithms have employed with various types of graphs. The main component of all these graphs is the relations between users and those items that have been rated/purchased by them. Besides that, the most common approach is generating a bipartite graph where the connections are from one part of the graph as users, to the other part as items [12,53]. Firstly, the bipartite graph is generated to representing the input data. In the second step,

many common approaches can be used to rank the items with utilizing the knowledge about the neighbors of the target user. Since the approachments, that are utilized the common neighbors, Katz similarity, diffusion scores, and personalized PageRank have been used in this domain [54-56]. However, these approaches are generally designed to utilize rating/binary feedback data and have important insufficiencies for the ranking-oriented class of neighbor-based CF algorithms. To overcome these deficiencies, a framework GRank is introduced in [53]. This framework determines the preference of users using a new tripartite preference graph model that demonstrates the relations between users, items, and pairwise preferences. Besides that, an extended version of personalized PageRank algorithm, that utilize top-N recommendation task to evaluation process, is proposed in [53].

Graph-based models are represented with the relations between users and items nodes as a user-item interaction graph in which there is a weighted or unweighted link between a user and each item that has rated before. Since, the recommendation generation in a user-item interaction graph can be considered as a link prediction problem, [26, 27]. For this purpose, the preference data is modelled tha as a bipartite graph structure and then the different kinds of relations existing in a ranking preference dataset (e.g. users' similarities, items' similarities, etc.) are explored. Also the links between users and items are weighted by utilizing complex numbers in the proposed structure. The type of user-user or item-item links are weighted as a real part of complex numbers, and the type of links between users and items are weighted as the imaginary part of complex numbers [28, 29]. Then, user-user or item-item links are labelled as similar or dissimilar, and the user- item links are labelled as like or dislike [28, 29]. After this labelling process, the proposed link prediction algorithm is implemented in the user-item interaction graph to predict which links are labelled as like or dislike since only items are recommended to users.

4.2. A Similarity-Inclusive Link Prediction Approach

In this dissertation, the proposed model is formulated to depend on the representation of complex numbers with real and imaginary parts in the complex form. In previous studies, similar or dissimilar links were weighted by real numbers, whereas like or dislike links were weighted by complex numbers [29]. Since a complex number

provides a natural algebraic link between real and imaginary values, the problem of recommendation could be considered as a problem of link prediction. With the utilization of the proposed method, other available algorithms of predicting links can still be used by no means of change. The proposed representation's validity and efficiency are assessed by evaluating the performance of the proposed recommendation approach in two real-world datasets.

User\Item	Item1	Item2	Item3	Item4	Item5	...
User1	3	5	3	1	4	
User2	5		5	2	1	3
User3		4			5	
User4	4	3		2	4	
...		2		3		5

(a)

User\Item	Item1	Item2	Item3	Item4	Item5	...
User1	j	j	j	-j	j	
User2	j		j	-j	-j	j
User3		j			j	
User4	j	j		-j	j	
...		-j		j		j

(b)

User\User	User1	User2	User3	User4	...
User1	1	0.4	0.6	0.8	
User2	0.4	1		0.2	0.2
User3	0.6		1	0.6	
User4	0.8	0.2	0.6	1	
...		0.2			1

(c)

Item\Item	Item1	Item2	Item3	Item4	Item5	...
Item1	1	0.4	0.2		0.4	
Item2	0.4	1	0.2		0.6	
Item3	0.2	0.2	1		0.2	
Item4				1	0.2	0.2
Item5	0.4	0.6	0.2	0.2	1	
...				0.2		1

(d)

	User1	User2	User3	User4	...	Item1	Item2	Item3	Item4	Item5	...
User1	1	0.4	0.6	0.8		j	j	j	-j	j	
User2	0.4	1		0.2	0.2	j		j	-j	-j	j
User3	0.6		1	0.6			j			j	
User4	0.8	0.2	0.6	1		j	j		-j	j	
...		0.2			1		-j		j		j
Item1	j	j		j		1	0.4	0.2		0.4	
Item2	j		j	j	j	0.4	1	0.2		0.6	
Item3	j	j				0.2	0.2	1		0.2	
Item4	-j	-j		-j	j				1	0.2	0.2
Item5	j	-j	j	j		0.4	0.6	0.2	0.2	1	
...		j			j				0.2		1

(e)

Figure 4.1. *(a) User-item rating matrix (b) rating conversion of rating matrix* $\boldsymbol{A}_{UI}$ *, (c) user-user similarity matrix* $\boldsymbol{A}_{UU}$ *, (d) item-item similarity matrix* $\boldsymbol{A}_{II}$ *, (e) the adjacency matrix* $\boldsymbol{A}$ *.*

4.2.1. Adjacency Matrix

The generation of adjacency matrix $\boldsymbol{A}$ in the proposed graph model slightly differs from the adjacency matrix represented as in (Eq. 3.10). The item-item and user-user

similarity matrices are represented as in (Eq. 3.10), $\boldsymbol{A}_{II}$, $\boldsymbol{A}_{UU}$, respectively. The user-item preference matrices represented as $\boldsymbol{A}_{UI}$, $\boldsymbol{A}_{IU}$, also the conjugate transpose of $\boldsymbol{A}_{IU}$ is described as, $\boldsymbol{A}_{IU} = -\boldsymbol{A}^{T}_{UI}$, as in (Eq. 3.10). An example of user-item rating matrix is a real matrix and illustrated in Figure 4.1.(a), The user-item preference matrix is a complex matrix after rating conversion process, is illustrated in Figure 4.1. (b). The similarity matrices are real matrices and a sample of user-user similarity matrix $\boldsymbol{A}_{UU}$ is illustrated in Figure 4.1.(c), an example of item-item similarity matrix $\boldsymbol{A}_{II}$ is illustrated in Figure 4.1.(d). The similarity factors are computed as hypothetically, since the only ratio of similar ratings are calculated as the similarity factors between users or items of the matrix which is illustrated in Figure 4.1 (a). Finally, the main adjacency matrix $\boldsymbol{A}$ in the proposed graph model is shown in Figure 4.1.(e).

The proposed algorithm similarity-inclusive link prediction method (SIMLP) differs slightly from the CORLP method [29] in the modeling of the adjacency matrix, and while calculating the powers of the adjacency matrix and yielding the final recommendation are in the same procedure. In the first step, we find user-user and item-item similarity matrices of the preference matrix with utilizing the cosine similarity measurement [39]. In the second step, the rating conversion process is applied to the user-item preference matrix. Following the combination of these matrices, the main adjacency matrix is built as in Eq. (3. 1).

$$\boldsymbol{A} = \begin{pmatrix} u_{11} & \cdots & u_{1n} & r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \cdots & -r_{1n} & i_{11} & \cdots & i_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & i_{m1} & \cdots & i_{mn} \end{pmatrix} , \tag{4.1}$$

where u_{ij} denotes the cosine similarity between the i^{th} and j^{th} users, i_{ij} denotes the cosine similarity between the i^{th} and j^{th} items, r_{ij} expresses the like/dislike relationship between the i^{th} user and j^{th} item, and $-r_{ij}$ expresses the like/dislike relationship between the i^{th} user and j^{th} item in Eq. (4.2). The generation process of the adjacency matrix is illustrated in Figure 3.1. The similarity values are calculated to depend on co-rated item's ratings

which are greater than 3, respectively, (see Figure 4.1 (c), (d)). The combination of user-item preference matrix, user-user similarity matrix, and item-item similarity matrix generates the main adjacency matrix $\boldsymbol{A}$, is illustrated in Figure 4.1 (e).

Furthermore, this adjacency matrix $\boldsymbol{A}$ is a square matrix (Eq. 4.1). Since we can use eigenvalue decomposition on this adjacency matrix in Eq. (4.1). For undirected networks, generally, the adjacency matrix $\boldsymbol{A}$ is symmetrical, and its eigenvalue decomposition may be considered as;

$$\boldsymbol{A} = \boldsymbol{U}\Lambda\boldsymbol{U}^T, \tag{4.2}$$

where $\boldsymbol{U}$ is an orthogonal matrix, and Λ is a diagonal matrix. The logic behind usually considering the adjacency matrix's eigenvalue decomposition is that it is possible to calculate a power of the matrix as $\boldsymbol{A}^k = \boldsymbol{U}\Lambda^k\boldsymbol{U}^T$, which may be used for expressing link prediction methods like the hyperbolic sine of matrix or the matrix exponential. Hence, the exponential of the adjacency matrix can be formulized as;

$$exp(\boldsymbol{A}) = \boldsymbol{U} \cdot exp(\Lambda) \cdot \boldsymbol{U}^T. \tag{4.3}$$

Moreover, the hyperbolic sine of the adjacency matrix can be formulized as;

$$sinh(\boldsymbol{A}) = \boldsymbol{U} \cdot sinh(\Lambda) \cdot \boldsymbol{U}^T. \tag{4.4}$$

Link prediction functions change the eigenvalues (See in Eq. (4.3, 4.4)), but do not change the eigenvectors. Since they are spectral transformations.

The link prediction function (hyperbolic sine) is applied to $\boldsymbol{A}$ as in, [29]. In our proposed method with another approachment, we multiply the link prediction function (hyperbolic sine) with a parameter α, then the predictions applied to $\boldsymbol{A}$ is represented as:

$$sinh(\alpha\boldsymbol{A}) = \alpha\boldsymbol{A} + 1/6 \cdot (\alpha\boldsymbol{A})^3 + 1/120 \cdot (\alpha\boldsymbol{A})^5 + \ldots. \tag{4.5}$$

4.2.2. Recommendation Methodology

Since the closeness values among the nodes are measured by the power sum of the adjacency matrix, the summation of each entry of the top-right and top-left components

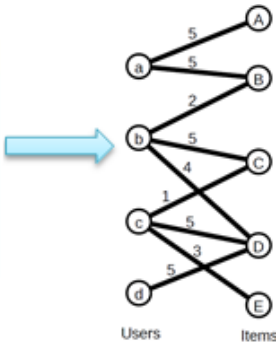
expresses the degree of whichever item is relevant to a specific user. After summation of these components, we obtain prediction scores that denote item recommendation to a particular user. We sort these scores in descending order; the user will like the item if the score is positive or dislike otherwise. Hence, the items with positive and higher values will be recommended to a particular user, if these recommended items are unnoticed by that user. Moreover, top-N recommendation lists are generated for each user by these sorted prediction scores [39].

The testing methodology adopted in this dissertation is the same as in a previous study [29]. The ratings are split by two subsets that are named by training and test sets, for each dataset. The test set includes only 5-star ratings and only items that are relevant to the corresponding users. The detailed procedure used to generate the training set and the test set may be defined as follows: Firstly, we select 10% of items rated by each user randomly to create a temporary test set, while the temporary training set includes other ratings. After the selection, the 5-star ratings in the temporary test set are further filtered out for the final test set, and the remaining ratings in the temporary test set are combined into the temporary training set for the final training set. Then, the training set is utilized to predict ratings or recommendation scores for each item-user pair.

Nevertheless, rating conversion is necessary for the adjacency matrix's generation of our proposed method, as illustrated in Figure 4.2. An example of the user-item rating matrix and the bipartite graph model of this matrix is drawn in Figure 4.2.(a), (b). The ratings in the preference matrix are converted to *like* or *dislike* edges/links based on whether the rating is greater than or equal to 3, (see in Figure 4.2. (c)). Bipartite rating graphs can be drawn as bipartite signed graphs, illustrated in Figure 4.2. (d). The green links are represented as '*like'* edges, red-dash links are represented as *'dislike'* edges in the bipartite signed graph, (see in Figure 4.1.(d)). After the rating conversion process, the ratings in the training set are converted to j or $-j$ based on whether the rating is greater than or equal to 3. Accordingly, in case that the rating is less than 3, it is changed by $-j$, which means that the user states *'dislike'* for the item; equivalently, when the rating is greater than or equal to 3, j is given to defining *'like'*. Furthermore, if the (u,i) pair is not included in the training set, the corresponding component of the adjacency matrix becomes zero. By this partitioning process of the dataset, computing the recommendation

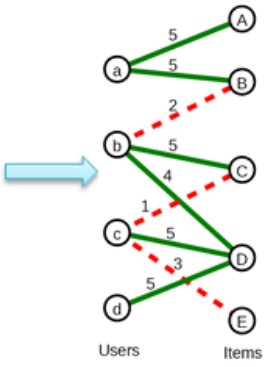
error becomes less meaningful. Hence, we focus on how many relevant items in the test set can be recommended to users. Also, the overall ratio of the items that recommended to all users is calculated.

User \ Item	A	B	C	D	E
a	5	5			
b		2	5	4	
c			1	5	3
d				5	



(a) (b)

User \ Item	A	B	C	D	E
a	like	like			
b		dislike	like	like	
c			dislike	like	like
d				like	



(c) (d)

Figure 4.2. *(a) User-item rating matrix (b) bipartite graph model (c) User-item relationship matrix, (d) bipartite signed graph (green links are represented as like, red-dash links are represented as dislike)*

4.2.3. Evaluation Metrics

Thus, the performance of the comparison methods is measured by using the metrics, hits rate and coverage [9, 15, 29]. In the case of the top-N recommendations, the overall hits rate and coverage are described by averaging all test cases:

$$hits\ rate(N) = \frac{\#hits}{|T|},\ coverage(N) = \frac{|\bigcup recommend(N,u)|}{\#items} \tag{4.6}$$

When the item i is included in the user u's top-N recommendations list for each pair (u,i) in the test set, it will get one hit. The overall hit is symbolized as $\#hits$, and

the number of test pairs is denoted as $|T|$. Hence the hits rate can be accepted as the capability to recommend relevant items to users — the recommendation set to the user u is denoted as;

$$recommend(N,u) \tag{4.7}$$

Thus, coverage is equal to the percentage of items that the system can recommend. Generally, coverage is utilized to determine models which recommend a limited number of items but have high accuracy. The higher coverage value is not only desirable but also useful to trust the accuracy of the metric results better [40]. The algorithm performs better when the values of these two metrics are higher.

Furthermore, the novelty measurement can be described as the complement of the item's popularity in the dataset:

$$1-p(i), \tag{4.8}$$

where $p(i)=\frac{|\{u\in U, r_{ui}\neq\varnothing\}|}{|U|}$ is the fraction of users who rated item i.

In order to measure recommendation techniques regarding with novelty metric, the novelty of individual recommendations is aggregated into a single score for a list of recommendations R, [70];

$$Novelty(R)=\frac{\sum_{i\in R}-log_2\ p(i)}{|R|} \tag{4.9}$$

4.2.4. Similarity Metrics

A similarity metric is a function denoting to a similarity between two observations. The most common similarity metrics used in recommender systems are *cosine*, *Jaccard*, *Pearson*, *adjusted-cosine*, *Adamic-Adar* and *Spearman*. There are various similarity measures to evaluate the similarity between two items or two users. As shown in the formulate below, each formula comprises terms given an m-by-n user-item interaction matrix, in which users are represented as m row vectors and items are represented as n column vectors.

- *Cosine similarity;* two items i and j are represented as $\vec{i}$ and $\vec{j}$ two vectors in the m dimensional user-space in this formulation. The similarity between them is defined by computing the cosine of the angle between these two vectors.

$$sim(\vec{i},\vec{j}) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{\left\|\vec{i}\right\|_2 * \left\|\vec{j}\right\|_2} \tag{4.10}$$

- *Jaccard similarity index;* also known as the Jaccard similarity coefficient, it compares ratings for two users' ratings vectors to find out which ratings are common and which are not. Ranging between 0% and 100%, it is a measure of similarity for two users' vectors, indicating that users with higher similarity have more common ratings. The basic idea is that users are more similar if they have more common ratings. The higher the percentage of *Jaccard index* means that the more number of common ratings between two users.

$$Jaccard(I_u, I_v) = \left|I_u\right| \cap \left|I_v\right| / \left|I_u\right| \cup \left|I_v\right| \tag{4.11}$$

  where $\left|I_u\right|$ and $\left|I_u\right|$ are the cardinality of items rated by users u and v .

- *Pearson correlation coefficient;* the similarity between two items, i and j , is measured by calculating the Pearson correlation similarity measure. First, we need to single out the co-rated cases (i.e., cases where the users rated both items i and j), to correctly evaluate the correlation. The correlation similarity is mathematically stated as "where the set of users is denoted by U who both rated items i and j ". Also, $R_{u,i}$ is denoted as the rating of the user u on an item i , $\overline{R_i}$ is the average rating of the i^{th} item, [14, 16].

$$sim(\vec{i},\vec{j}) = pearson(\vec{i},\vec{j}) = \frac{\sum_{u\in U}(R_{u,i} - \overline{R_i})(R_{u,j} - \overline{R_j})}{\sqrt{\sum_{u\in U}(R_{u,i} - \overline{R_i})^2}\sqrt{\sum_{u\in U}(R_{u,j} - \overline{R_j})^2}} \tag{4.12}$$

- *Adjusted cosine similarity;* This similarity measurement is a modified form of cosine similarity where the system considers that different users have different

rating behavior; in other words, some users may rate items highly in general, and others may give items lower ratings as a preference. The average ratings for each user u, $\overline{R_u}$ is subtracted from each user's ratings $R_{u,i}, R_{u,j}$ for the pair of items i and j as in Eq. (4.13):

$$sim(\vec{i},\vec{j}) = acos(\vec{i},\vec{j}) = \frac{\sum_{u\in U}(R_{u,i}-\overline{R_u})(R_{u,j}-\overline{R_u})}{\sqrt{\sum_{u\in U}(R_{u,i}-\overline{R_u})^2}\sqrt{\sum_{u\in U}(R_{u,j}-\overline{R_u})^2}} \tag{4.13}$$

- *Spearman similarity coefficient*; is obtained by applying the Pearson coefficient to rank-transformed data. The correlation coefficient between item sequences $\boldsymbol{X} = \{x_i : i = 1,...,n\}$, $\boldsymbol{Y} = \{y_i : i = 1,...,n\}$ is defined by Eq. (4.14);

$$spearman(\boldsymbol{X},\boldsymbol{Y}) = \rho = \frac{6\cdot\sum_{i=1}^{n}[R(x_i)-R(y_i)]^2}{n\cdot(n^2-1)} \tag{4.14}$$

where $R(x_i)$ and $R(y_i)$ represent ranks of x_i and y_i in item sequences of $\boldsymbol{X}$ and $\boldsymbol{Y}$.

- *Adamic-Adar (AA) Coefficient*; favors the common neighbors that have fewer neighbors, then evaluate how strong the relationship between a common neighbor. The coefficient is evaluated between pair of nodes same as in [73]:

$$AA(x,y) = \sum_{z\in|\Gamma(x)\cap\Gamma(y)|}\frac{1}{\log(|\Gamma(z)|)}, \tag{4.15}$$

where $\Gamma(x),\Gamma(y)$ are the set of neighbors of node x and y, and $|\Gamma(z)|$ is the degree of the node z.

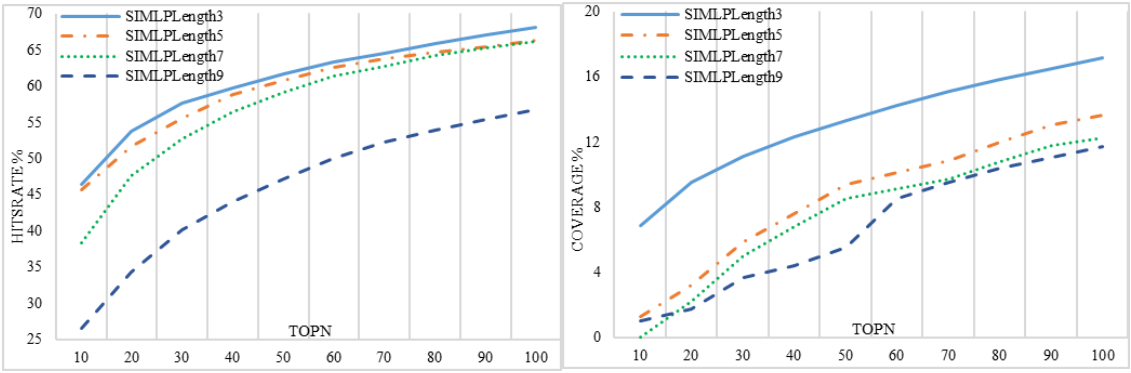
4.3. Experimental Evaluation and Datasets

The proposed algorithm and other comparison methods are implemented in two real-world datasets: MovieLens and MovieLens Hetrec, [41, 42]. These datasets are publicly stored movie rating datasets that were compiled by GroupLens research from the MovieLens and hetrec2011 websites. The former consists of 100,000 ratings ranging from 1-to-5 from 943 users on 1,682 movies. The MovieLens Hetrec dataset consists of

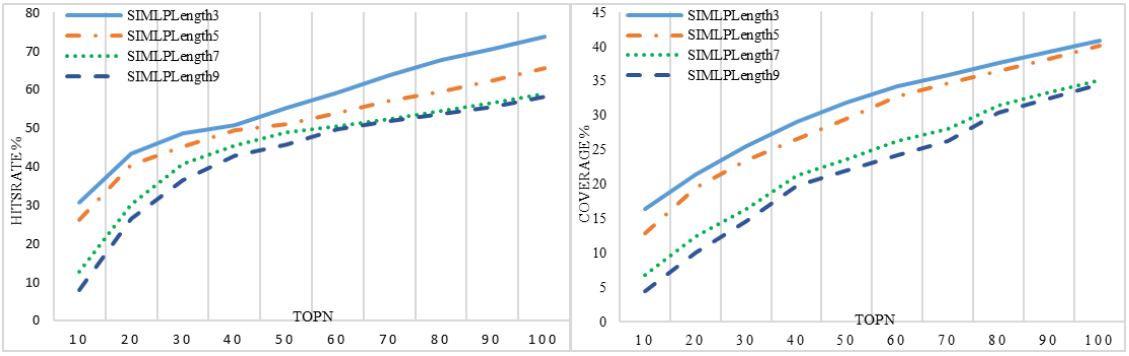
855,598 ratings ranging from 1-to-5 from 2,113 users on 10,197 movies. Firstly, ratings in these datasets are converted into complex numbers, then the complex biadjacency matrices of these datasets are obtained. Secondly, the cosine similarity measure is applied to the user-item rating matrices of these datasets. Lastly, the user-user cosine similarity matrices and item-item cosine similarity matrices of rating matrices of these datasets are computed. After combining all these matrices, the main adjacency matrices are generated as a square matrix for these two datasets as in Eq. (4.1). Therefore, the hyperbolic sine function is applied on the adjacency matrix as a link prediction function, as in [29]. Hyperbolic sine function calculates the sum of the odd powers and gives the shortest path of lengths in bipartite systems. Such function provides a higher score when more paths are connecting two nodes. Therefore, it is needed to have higher powers of the adjacency matrix.

The more paths between two nodes and the shorter these paths are, the most substantial relationship between these two nodes will be in the forecast. Thus, the first experiment was designed to test the performances of the recommendation algorithms based on the link prediction approach with different path lengths for the recommendation generation process. The shortest path of lengths 3, 5, 7 and 9 are found because the sum of the odd powers of bipartite graphs is vital to make a recommendation. For instance, when the path length is chosen as 3, the number of positive value paths with length 3 from user u to item i is more than other path lengths. Hence, if there exist more positive paths from u to i and less negative paths between them, the most probable that i will be recommended to u. Note that the length needs to be odd, and not smaller than 3. As a similar consequence, results of the SIMLP method with top-N recommendations are given. Figure 4.3 shows the results of the SIMLP algorithm with lengths 3, 5, 7 and 9.

Figure 4.3. illustrates that the results of coverage and hits rate decreases as the path length increases in these datasets. Moreover, the proposed algorithm performs much better in the MovieLens dataset than in the Hetrec dataset, since the latter is much sparser and its links between users and items are scarce compared to Movielens. It still shows a higher performance with path length-3 for recommendation generation in the MovieLens and Hetrec datasets.



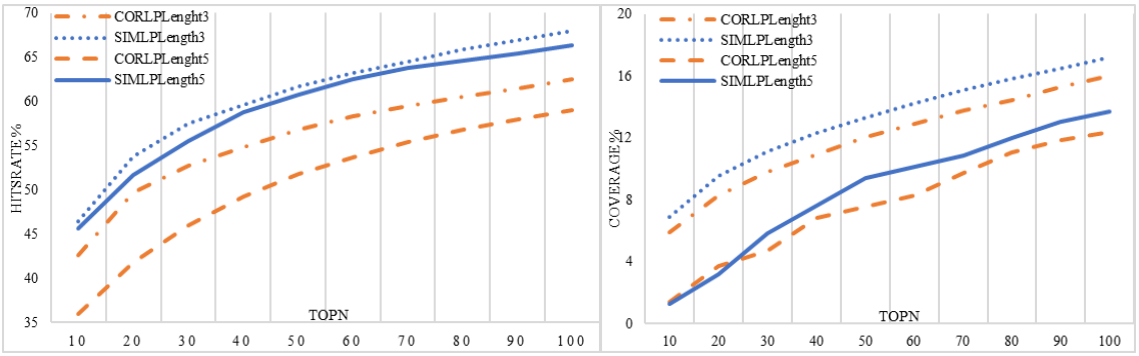
(a)



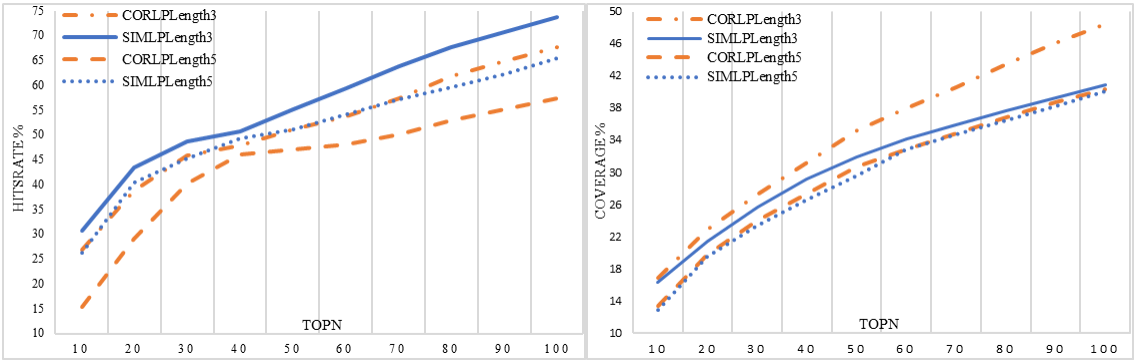
(b)

Figure 4.3*. The hits rate and coverage comparison of SIMLP with different path lengths for top-N recommendation on Hetrec (a) and MovieLens (b) datasets.*

An item-based top-N recommendation algorithm are implemented to measure results. The length of top-N item recommendation lists is increased from 10 to 100. Then, these results are compared with CORLP method based on the fundamental link prediction approach with complex numbers, introduced in [29]. Figure 4.4 illustrates the comparison of results with the CORLP method with path length 3 and length 5 for the different top-N recommendation. The figure shows that the hits rate of the SIMLP method is higher than CORLP method, but the coverage is relatively the same as with the CORLP on the two datasets. Therefore, the link prediction function is modified with multiplying a parameter α, as in Eq. (4.4). Then, the experimental results are obtained with using the modified function to improve the performance of recommendation.



(a)



(b)

Figure 4.4. *The hits rate and coverage comparison of SIMLP and CORLP with path length 3, 5 for topN recommendation on Hetrec (a) and MovieLens (b) datasets.*

The CORLP and SIMLP algorithms are modified with multiplying a scaling parameter α, [43]. The multiplication parameter α is selected as a decimal number in the range of [0, 1], and is decreased from 1 to 0. While SIMLP can obtain higher performance with all path lengths, CORLP method is performed well only with a path of length 3. Thus, only path length 3 and top60/top100 recommendation lists are considered while experiments which compare the proposed method with CORLP method are conducting. Figure 4.5 illustrates the comparison of hits rate and coverage with the recommendation method CORLP [29]. The results show that the SIMLP achieves higher hits rate and this method provides relatively high coverage with decreasing multiplication parameter.

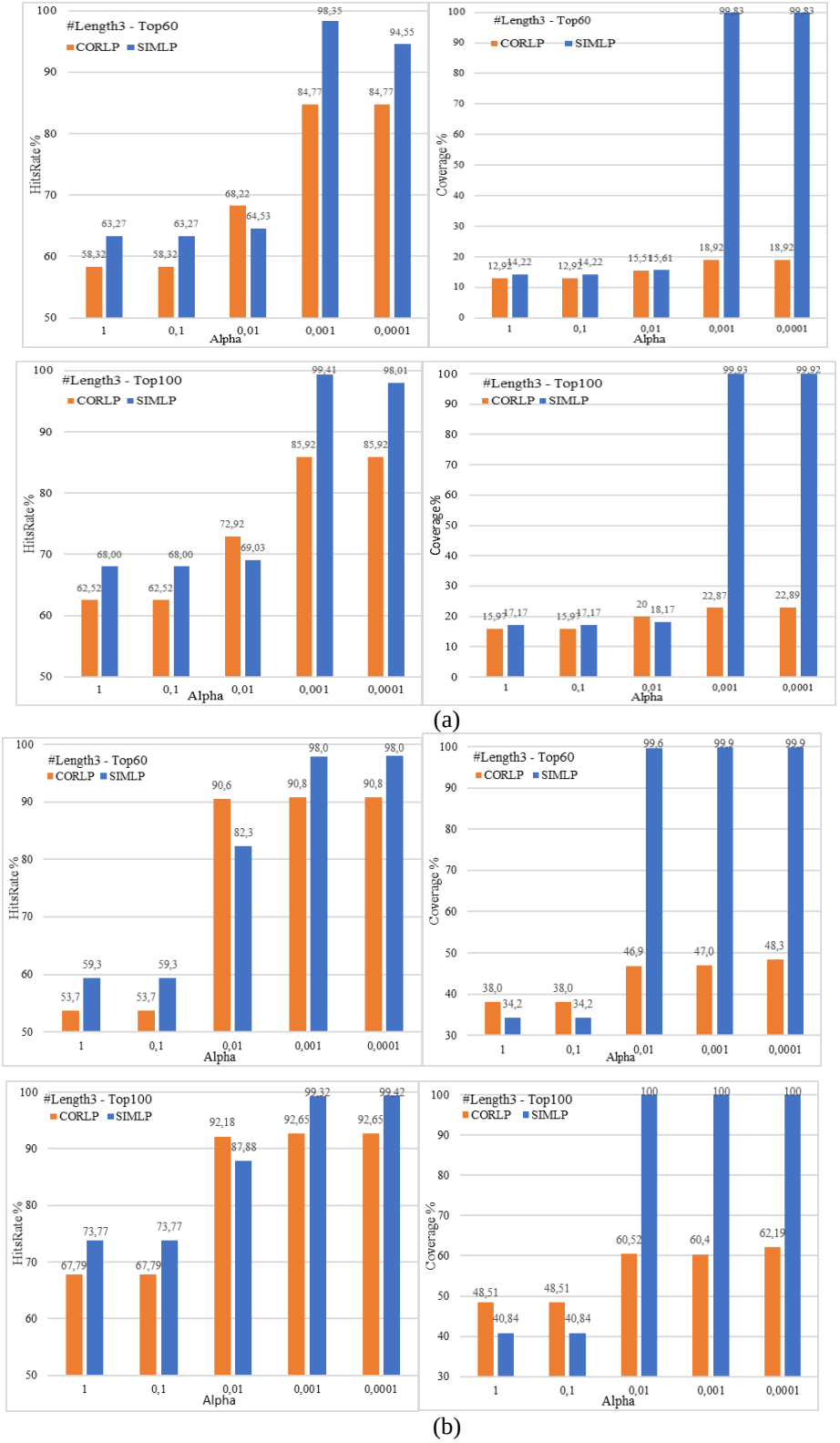
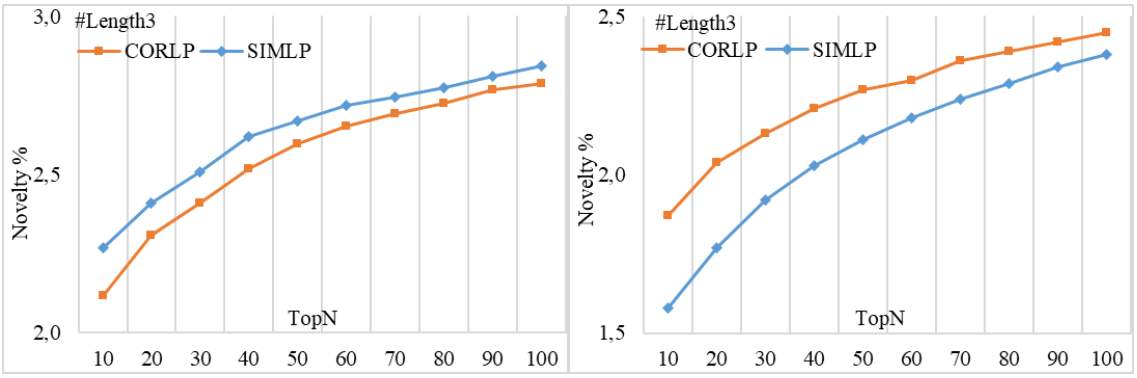
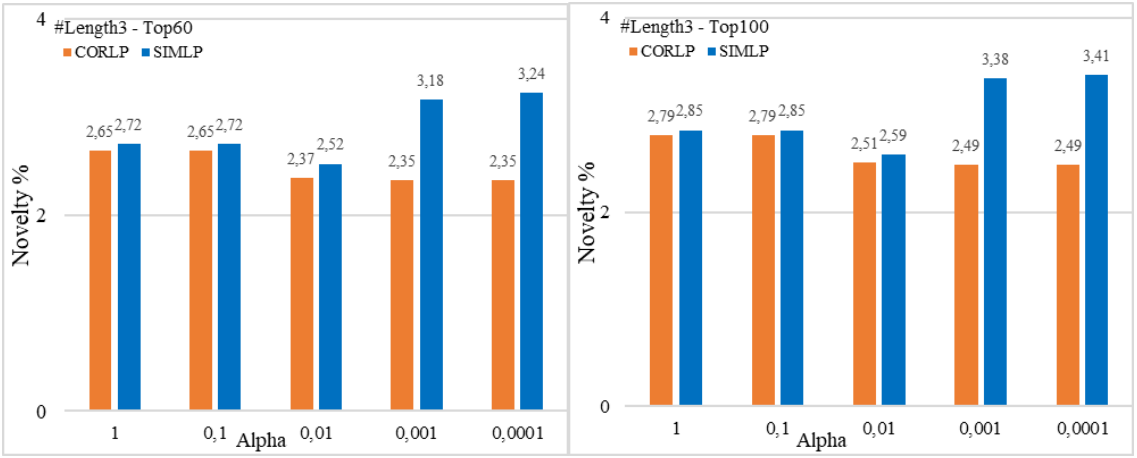


***Figure 4.5**. The coverage and hits rate comparison of SIMLP and CORLP with path length 3 for top60/top100 recommendation on Hetrec (a) and MovieLens (b) datasets.*

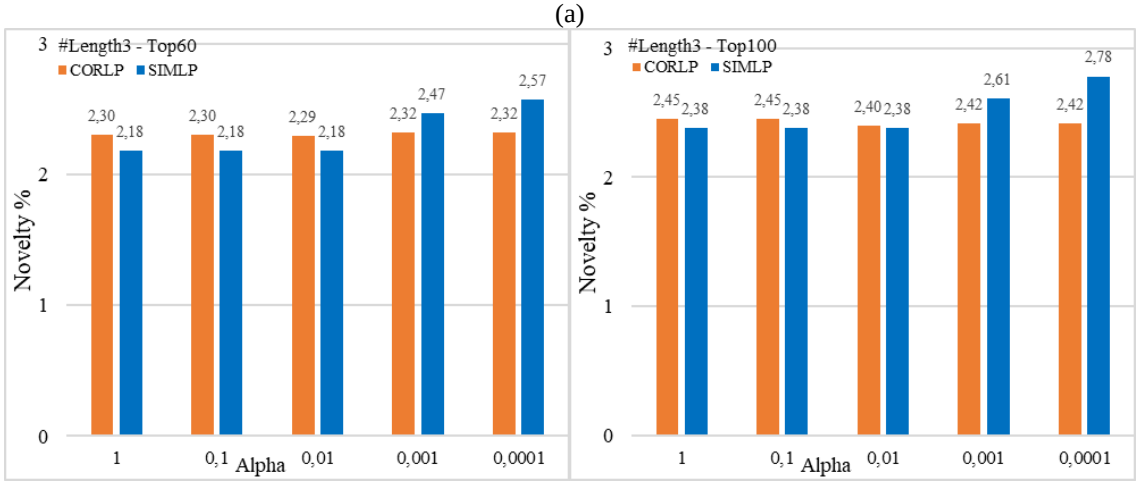


(a) (b)

***Figure 4.6**. The novelty comparison of SIMLP and CORLP with path length 3 for top-N recommendation on MovieLens (a) and Hetrec (b) datasets.*



(a)



(b)

***Figure 4.7**. The novelty comparison of SIMLP and CORLP with path length 3 for top60/top100 recommendation on Hetrec (a) and MovieLens (b) datasets.*

Figure 4.6 illustrates the comparison of novelty with the recommendation method CORLP [29]. The results show that the SIMLP achieves higher novelty than CORLP

method for top-N recommendation on MovieLens dataset Figure 4.6. (a) but not for the Hetrec dataset, Figure 4.6. (b). Hence, we need to modify the link prediction function with multiplying a parameter α, as in Eq. (4.4). Figure 4.7 illustrates the comparison of novelty with the modified CORLP and SIMLP recommendation methods. The results demonstrates that the SIMLP achieves higher novelty result with decreasing multiplication parameter. It can be concluded that SIMLP method generates more diverse and novel recommendations than CORLP method.

Table 4.1. *The p-values of the comparison of the CORLP and the SIMLP methods for each evaluation metrics on MovieLens and Hetrec datasets.*

Evaluation Metrics / Dataset	Hits Rate	Coverage	Novelty
Movielens	0.0351	0.0071	0.0220
Hetrec	0.0014	0.0161	0.0005

In this dissertation, the signifance of the proposed SIMLP approach and the comparison method CORLP are investigated. The signifance is determined with the answer of this question: Does the proposed SIMLP approach that utilizes cosine similarities perform better than CORLP approach for top-N recommendation task? The hits rate, coverage and novelty are utilized as the evaluation metrics to measure the performance of the proposed recommendation algorithm. Two-factor Anova test is conducted to further evaluate performance differences among SIMLP and CORLP approaches, [66]. Thus, the specific hypotheses examined in this dissertation are:

• H1: The SIMLP based recommendation approach achieves higher hits rate than the CORLP based recommendation approach does.

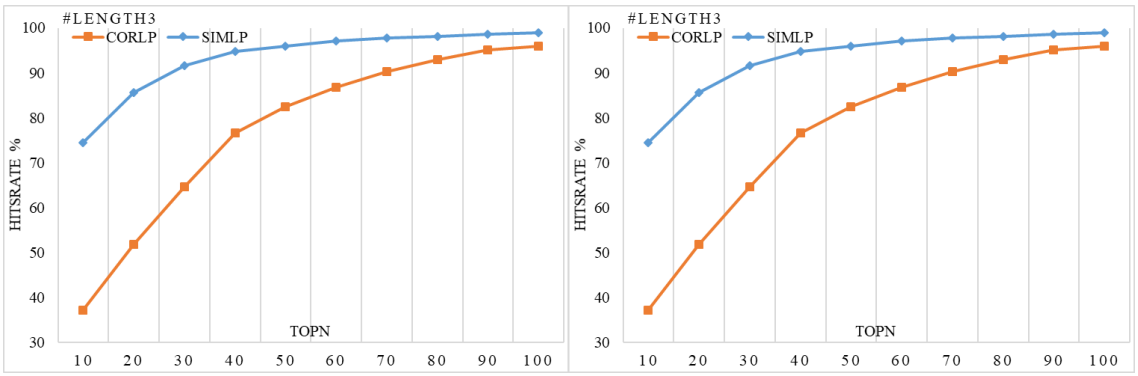
• H2: The SIMLP based recommendation approach achieves higher coverage than the CORLP based recommendation approach does.

• H3: The SIMLP based recommendation approach achieves higher novelty than the CORLP based recommendation approach does.

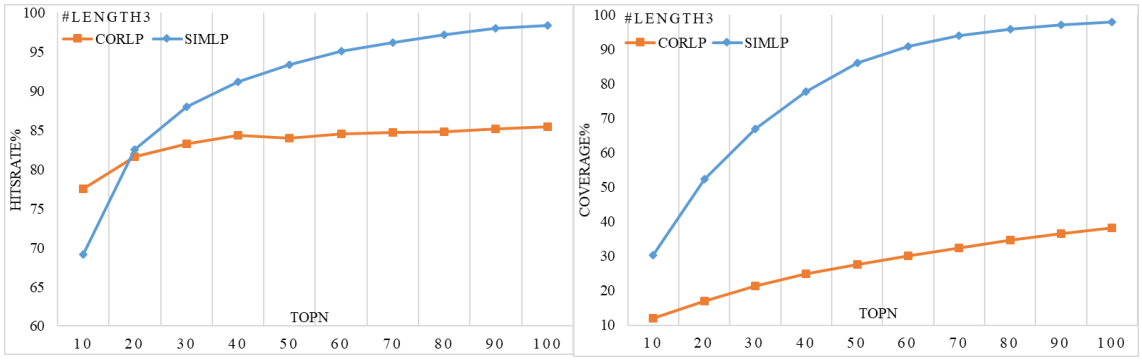
Table 4.1. demonstrates that there are statistically significant differences between CORLP and SIMLP methods with respect to hits rate, coverage and novelty for the experiments on Movielens and hetrec datasets, hence all the p-values are smaller than 0.05. The hypotheses H1, H2 and H3 are supported for each evaluation metrics defined in this chapter by our experimental results.

We propose to modify the CORLP and SIMLP algorithms with using the Neumann Kernel as a link prediction function since to investigate SIMLP and CORLP methods behaviors' after changing link prediction algorithms. Hence, the Neumann Kernel is used as a link prediction function, that formulated as in Eq. 3.5. Firstly, α is set $\alpha = 0.001$ and $\alpha = 0.0001$ in Eq. (3.5) respectively for the experiments on MovieLens and MovieLens Hetrec dataset. Secondly, the similarity values are evaluated with using cosine similarity metrics for the proposed SIMLP method. Figure 4.8 illustrates the comparison of hits rate and coverage with the CORLP and SIMLP recommendation method that are utilized Neumann Kernel as a link prediction function. The results of the SIMLP and CORLP algorithm are measured with path length 3, and with using cosine similarity metrics. Moreover, the results show that the proposed SIMLP method achieves higher hits rate and coverage.

With another approachment, the cross-validation approach is used in the testing methodology to determine the significance of the proposed SIMLP method. The ratings are split by two subsets that are labelled as training and test sets, for each dataset. Since, the 10- fold cross-validation method is used for generating training and test sets, [19]. At each fold, the test set is forced to include only 5-star ratings to assure validity. The test set has only 5-star ratings and only the items relevant to the corresponding users, as in previous works [29, 51]. The detailed procedure used to create the training set and the test set is as follows: First, we detect all 5-star ratings from the rating matrix. Then, in which user-item pairs have 10% (to make experiments as 10-fold) of these ratings are randomly determined to create a temporary test set. Following this selection, the 5-star ratings in the temporary test set are further filtered out for the final test set, and the remaining ratings constitute the training set. Finally, this training set is used to predict ratings or recommendation scores for each item-user pair.



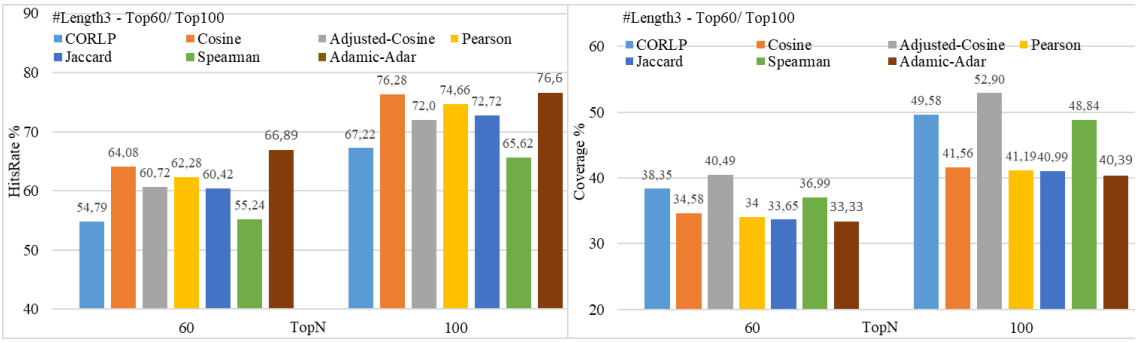
(a)



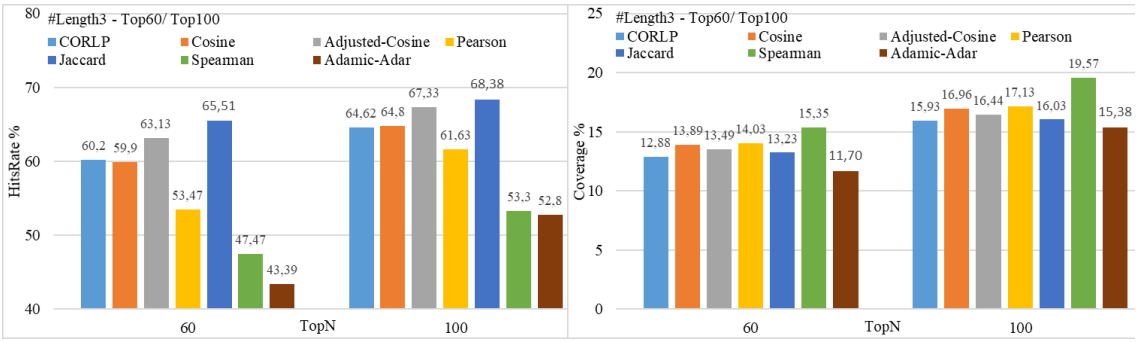
(b)

Figure 4.8. *The coverage and hits rate comparison of SIMLP and CORLP methods that are utilized Neumann link prediction kernel for recommendation on MovieLens (a) and Hetrec (b) datasets.*

The experimental results of the SIMLP and CORLP method with using the 10-fold cross-validation method are given in Figure 4.9. Besides that, the results of the SIMLP algorithm are measured with path length 3 to compare with the CORLP method, Figure 4.9. shows that the proposed SIMLP method achieves higher hits rate and coverage with using Adjusted Cosine similarity metric than using other metrics, for all these two datasets. As can be observed in Figure 4.9, SIMLP method yields comparatively higher results than CORLP when SIMLP is modified with utilizing similarities that are measured by each similarity metrics defined in this dissertation with the exception of Spearman similarity measurement.



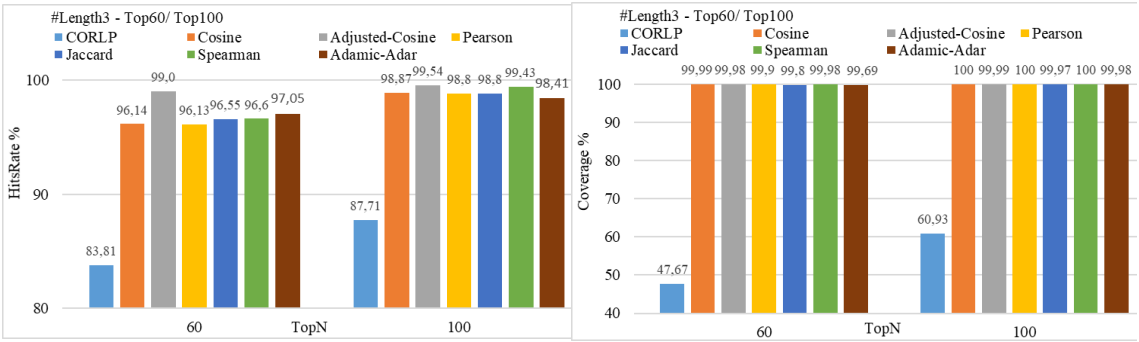
(a)



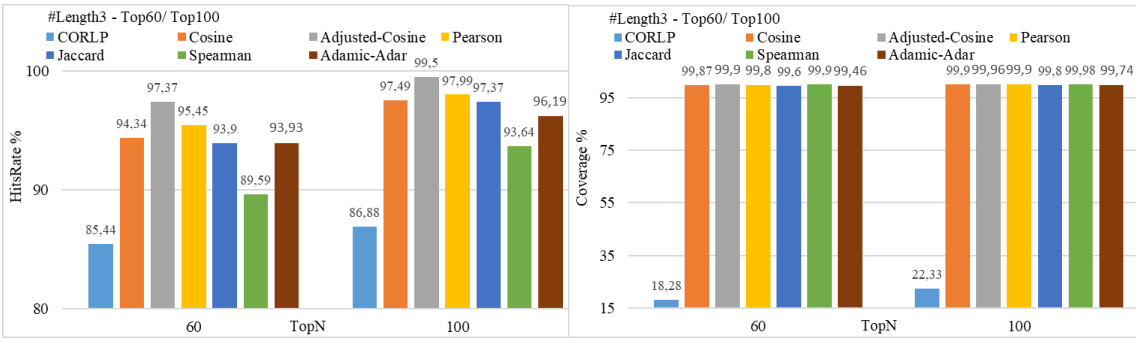
(b)

Figure 4.9. *Comparison of the SIMLP and CORLP methods by coverage and hits rate with using different similarity metrics for top-N recommendation on MovieLens (a) and Hetrec (b) datasets.*

Furthermore, the CORLP and SIMLP algorithms are proposed to modify with using the Neumann Kernel as a link prediction function, in order to investigate SIMLP and CORLP methods' behaviors after changing the testing methodology as the 10-fold cross-validation approach. Hence, the Neumann Kernel formulation in Eq. (3.5) is used as a link prediction function and α is set $\alpha = 0.001$ and $\alpha = 0.0001$ in Eq. (3.5) as the former experiments respectively on MovieLens and MovieLens Hetrec dataset. As the same as, the results of the SIMLP and CORLP algorithm are measured with path length 3. The results of the CORLP and SIMLP method by using various similarity metrics for top-N recommendation tasks are illustrated in Fig. 4.10. The results indicate that the proposed SIMLP method obtains higher hits rate and coverage than CORLP method when SIMLP utilized with similarity metrics defined in this dissertation.



(a)



(b)

Figure 4.10. *Comparison of SIMLP and CORLP methods by coverage and hits rate utilizing Neumann link prediction kernel for top-N recommendation on MovieLens (a) and Hetrec (b) datasets.*

In this dissertation, answers of two questions are investigated. These questions are: Does the proposed SIMLP approach that utilizes various similarities perform better than CORLP approach? Which similarity metrics improve the performance of SIMLP for top-N recommendation task? The hits rate and coverage are used as the primary measures to compute the performance of the proposed recommendation algorithm. Two-factor Anova test is conducted to further evaluate performance differences among SIMLP and CORLP approaches, [66]. Thus, the specific hypotheses examined in this dissertation are:

- H1: The SIMLP based recommendation approach achieves higher hits rate than the CORLP based recommendation approach does.
- H2: The SIMLP based recommendation approach achieves higher coverage than the CORLP based recommendation approach does.

Table 4.2. includes the only one p-value that reflects no significant differences among CORLP and SIMLP (that utilizes Spearman similarities) methods with respect to hits-rate

on MovieLens dataset. Table 4.2. demonstrates that there are statistically significant differences between CORLP and SIMLP methods with respect to hits rate. The hypotheses H1 are supported for each similarity metrics defined in this dissertation with the exception of Spearman similarity measurement by our experimental results. As shown in Table 4.3., there are no significant differences between CORLP and SIMLP (that utilizes Spearman, Jaccard and adjusted-cosine similarities) methods with respect to coverage on MovieLens and Hetrec dataset. The other p-values in Table 4.3. demonstrate that statistically significant differences are observed between CORLP and SIMLP methods with respect to coverage. The hypotheses H2 are supported for each similarity metrics defined in this dissertation with the exception of Spearman and adjusted-cosine similarity measurements by our experimental results.

Table 4.2. *The p-values of the comparison of the CORLP and the SIMLP methods for each similarity metrics with regards to hits rate on MovieLens and Hetrec datasets.*

Similarity Measure / Dataset	Cosine	Adjusted-Cosine	Pearson	Jaccard	Spearman	Adamic-Adar
Movielens	0.0007	0.0061	0.0060	0.0113	0.6744	0.0132
Hetrec	0.0183	0.0176	0.0002	0.0009	0.00001	0.0006

Table 4.3. *The p-values of the comparison of the CORLP and the SIMLP methods for each similarity metrics with regards to coverage on MovieLens and Hetrec datasets.*

Similarity Measure / Dataset	Cosine	Adjusted-Cosine	Pearson	Jaccard	Spearman	Adamic-Adar
Movielens	0.0081	0.1286	0.0045	0.0025	0.7488	0.0077
Hetrec	0.0351	0.1471	0.0235	0.4518	0.0013	0.0088

Table 4.4. *The p-values of the comparison of the SIMLP and the CORLP methods that utilize Neumann link prediction kernel with regards to hits rate on MovieLens and Hetrec datasets.*

Similarity Measure / Dataset	Cosine	Adjusted-Cosine	Pearson	Jaccard	Spearman	Adamic-Adar
Movielens	0.0001	0.0001	0.0001	0.00004	0.0001	0.0006
Hetrec	0.0212	0.0969	0.0012	0.0244	0.7266	0.001

Table 4.5. *The p-values of the comparison of the SIMLP and the CORLP methods that utilize Neumann link prediction kernel with regards to coverage on MovieLens and Hetrec datasets.*

Similarity Measure / Dataset	Cosine	Adjusted-Cosine	Pearson	Jaccard	Spearman	Adamic-Adar
Movielens	6×10^{-11}	5×10^{-11}	6×10^{-11}	9×10^{-11}	5×10^{-11}	1×10^{-10}
Hetrec	5×10^{-14}	4×10^{-14}	1×10^{-14}	1×10^{-14}	1×10^{-14}	1×10^{-14}

The p-values of the comparison between the SIMLP and the CORLP methods that utilize Neumann link prediction kernel with regards to hits rate and coverage on MovieLens and Hetrec datasets are given in Table 4.4 and 4.5. The two p-values, that reflect no significant differences between CORLP and SIMLP (that utilizes Spearman and adjusted-cosine similarities) methods with respect to hits rate on Hetrec dataset, are given in Table 4.4. The other p-values in Table 4.4. indicate that there are statistically significant differences between CORLP and SIMLP methods with respect to hits rate. The hypotheses H1 are supported for each similarity metrics defined in this study with the exception of Spearman and adjusted-cosine similarity measurements by our experimental results. Table 4.5. demonstrates that there are statistically significant differences between CORLP and SIMLP methods with respect to coverage. Then, the hypotheses H2 are supported for each similarity metrics defined in this dissertation.

4.4. Conclusion

The proposed recommendation algorithm is based on the link prediction approach with the weights in the graph represented by complex numbers that can accurately differentiate similarity between two users/items and the like between a user and an item. With the proposed method, available link prediction algorithms may reprocess without any modifications. It is empirically observed that the complex number-based algorithms obtain comparatively higher performance than the state-of-the-art methods, [29]. The improvements of these algorithms are attributed to the inclusion of similarity factors that made algorithms more effective. The experimental results show that the performance of the proposed SIMLP method is better than other complex number-based algorithms with using two quality metrics: coverage and hits rate on the MovieLens Hetrec and MovieLens datasets. The results indicate that the hits rate of the SIMLP method is significantly better than other methods, whereas the coverage is just the same as the

results of other methods on the two datasets. After modifying the link prediction function with a scaling parameter, the results are improved for different top-N recommendations.

Moreover, SIMLP method that modified/utilized by six different similarity measurement to experimentally scrutinize how such similarity measures perform with top-N item recommendation task over the standard Hetrec and MovieLens datasets. The experimental results indicate that hits rate and coverage can be improved by about 7% and 4%, respectively, with Jaccard and Adjusted-Cosine similarity measures being the best performing similarity measures. Significant improvements are observed over the previous CORLP approach. After that, the SIMLP and CORLP method is modified by the link prediction function as Neumann kernel, and these modified algorithms are examined. About %12 better hits rate and about %40 better coverage are obtained with SIMLP as opposed to CORLP, both in MovieLens and Hetrec datasets. The proposed SIMLP algorithm is observed to be significantly superior to CORLP with utilizing two-factor anova test results. The results demonstrate that the proposed SIMLP method overcomes the deficiencies in graph-based recommender systems, making the proposed recommender system a preferable alternative.

Design of a graph image-based recommender system based on semantic relationships among images may be considered as a future follow-up of this study.

5. A SIMILARITY-INCLUSIVE LINK PREDICTION BASED HYBRID RECOMMENDER SYSTEM APPROACH

5.1. Motivation

In recently studies, it is observed that incorporating richer input data beyond numerical ratings can improve recommendation performance of the recommender systems. This observation is led to develop hybrid recommender systems, and these recommender systems are getting more popular in recent years. [17, 14]. At the first sight, hybrid recommendation approaches commonly referred as the integration of both CBF and CF [13, 14] techniques for recommendation generation. Then, the integration can be in different forms such as weighting, switching, mixing, etc., based on the available content profiles and meta-recommendation algorithms. More recently, many e-commerce applications have aggregated a rich source of information, such as text, image, rating, etc., which represent different aspects of user preferences. Hence, the concept of hybrid recommendation is extended with the integration of various information sources. Then integrating heterogeneous information to generate recommendation is become possible by translating the various information sources into a unified representation space [22, 23]. Such as, a visual representation space are generated to map items with utilizing items' visual images based on similar visual features and also utilizing the information about which users preferred them [24]. Since, visual images of products/items are exposed users' visual preference towards different items beyond numerical ratings and textual reviews. Moreover, a visual-based Bayesian personalized ranking approach that utilized top-N recommendation task, is proposed as an image-based recommendation system to extract co-purchase recommendations [24].

Images that widely exist in e-commerce sites, social networks, and many other applications, are one of the most important information resources integrated into recently developed image-based recommender systems. Hence, image-based recommender systems are drawing increasing attention in recent years. Such as, the visually explainable recommendation approach, that utilize deep learning-based models to obtain visual explanations, is introduced in [23]. A deep neural network is developed to generate image regional highlights as explanations for target users in this recommendation approach. In recent studies, researchers have also jointly considered ratings and images to generate

recommendation [24, 25, 30, 44]. Many existing recommendation approaches are still restricted to limited information sources (such as rating plus another input data), or require the pre-existence of domain knowledge to generate recommendation. To overcom these challenges/deficiencies, a new graph based hybrid framework is described for recommendation generation in this chapter. Firstly a simple overview of the framework is provided and then adopt/utilize two different information sources (visual images, and numerical ratings) to describe how the proposed framework can be developed in practice.

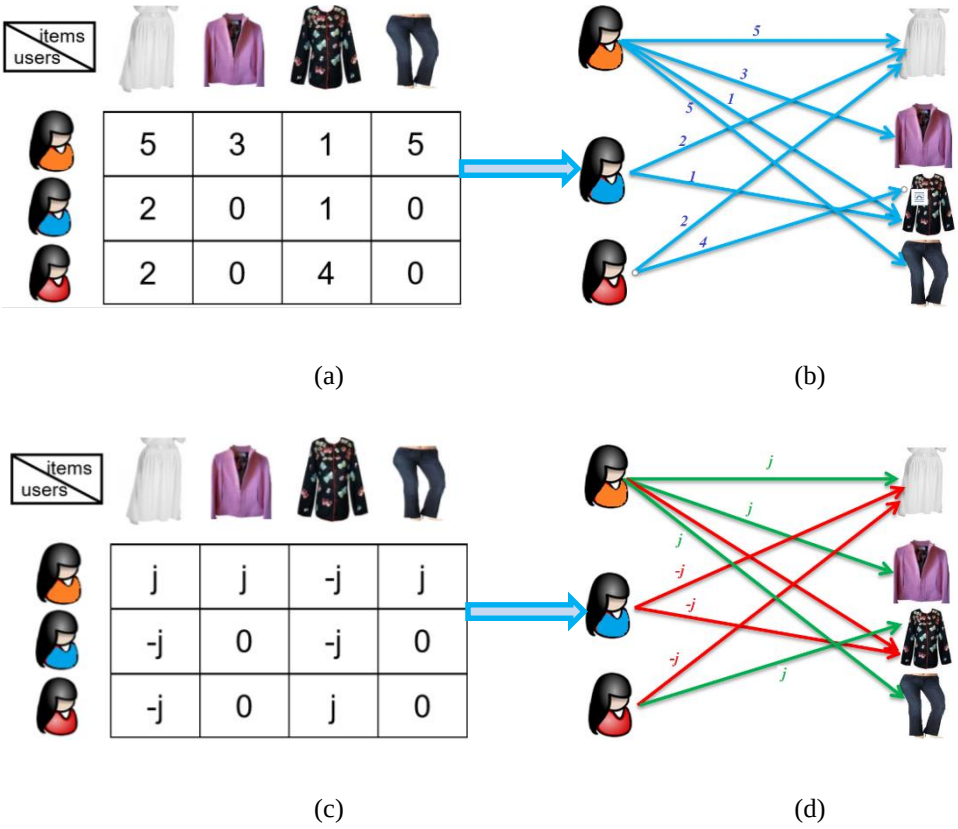


Figure 5.1. *(a) User-item rating matrix, (b) bipartite graph model (c) rating conversion of rating matrix* $\boldsymbol{A}_{UI}$ *, (d) bipartite signed graph.*

5.2. A Link Prediction Based Hybrid Recommender System Approach

The similarity-inclusive link prediction method (SIMLP) for hybrid recommendation system (Hybrid-SIMLP) differs slightly from the previously proposed algorithm in the 4th chapter. Hence, the modeling process of the adjacency matrix for Hybrid-SIMLP algorithm is different from SIMLP algorithm, while the processes of

calculating the powers of the adjacency matrix and providing the final recommendation are in the same procedure, which is based on top-N recommendation task.

In the beginning, a user-item rating/preference matrix are generated from the datasets, that described in [45]. Then, rating conversion is applied to this user-item preference matrix and this generated matrix is denoted as $\boldsymbol{A}_{UI}$. Moreover, the conjugate transpose of $\boldsymbol{A}_{IU}$ can be represented as in the previous chapter $\boldsymbol{A}_{IU} = -\boldsymbol{A}^{T}{}_{UI}$ and these preference matrices $\boldsymbol{A}_{UI}$ and $\boldsymbol{A}_{IU}$ are denoted as complex matrices (see in, Figure 5.1.(c)). An example of the user-item rating matrix and the bipartite graph model of this matrix is illustrated in Figure 5.1 (a) and (b). Rating conversion of user-item preference matrix is given in Figure 5.1. (c). User-item interaction (bipartite signed) graph after the rating conversion of user-item preference matrix is drawn in Figure 5.1. (d). Green links are represented as '*like'* edges and denoted as j , red links are represented as *'dislike'* edges and denoted as $-j$ in this bipartite signed graph, (see in Figure 5.1. (d)).

Secondly, the item-feature matrix is generated with using all items of datasets, which is introduced in [45]. Item features are extracted as the same feature extraction procedure, that described in [22]. These features are AlexNet features of the items. The users' visual preference towards different items' images are revealed to generate users' visual features. In the first step, all items that users' rated or bought before are found. In the second step, AlexNet feature vectors of these items are computed. Then, these feature vectors, that belongs to items rated from a particular user before, are summed up. Finally, the mean of this summation is calculated with the number of items that users' rated before. Hence, each user can be represented/denoted as a 4096-dimensional visual feature vector. An example of this procedure is illustrated in Figure 5.2. An illustration of an item feature matrix is given in Figure 5.2 (a). The process of generating a user feature vector is illustrated in Figure 5.2. (b), and the generated users' feature matrix is given in Figure 5.2. (c).

The process of generating the main adjacency matrix for the proposed hybrid recommender system is varied somewhat points from the generation of the adjacency matrix, that described in chapter 4. These variations are depending on the generating user-user and item-item similarity matrices. The user-user similarity matrix is computed from

the generated user-feature matrix, and item-item similarity matrix is computed from the generated item-feature matrix with utilizing the cosine similarity measurement. Then, the main adjacency matrix is generated with the combination of user-item preference matrix $\boldsymbol{A}_{UI}$, the conjugate transpose of this matrix $\boldsymbol{A}_{UI}$, and the item-item and user-user similarity matrices $\boldsymbol{A}_{II}$, $\boldsymbol{A}_{UU}$ as in Eq. (3. 10). The main adjacency matrix, which is given in Eq. (5.1), is reformulated for the proposed hybrid recommendation algorithm, this reformulation can be represented as:

$$\boldsymbol{A}=\begin{pmatrix} u_{11} & \cdots & u_{1n} & r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mn} & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \cdots & -r_{1n} & k_{11} & \cdots & k_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & k_{m1} & \cdots & k_{mn} \end{pmatrix}, \tag{5. 1}$$

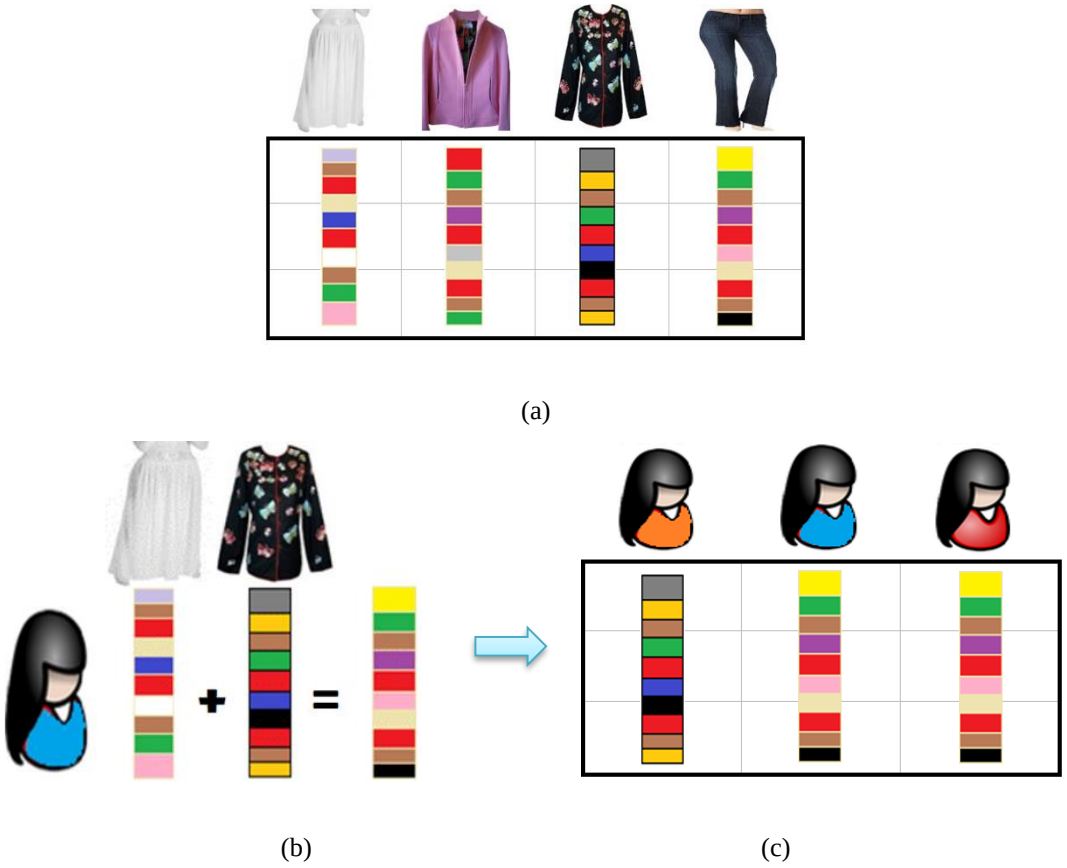
where u_{ij} denotes the cosine similarity value between the i^{th} and j^{th} users' visual feature vectors, k_{ij} denotes the cosine similarity value between the i^{th} and j^{th} items visual feature vectors, r_{ij} expresses the like/dislike relationship between the i^{th} user and j^{th} item, and $-r_{ij}$ expresses the like/dislike relationship between the i^{th} user and j^{th} item in Eq. (5.1).

Furthermore, the hyperbolic sine of this adjacency matrix is implemented as a link prediction function for this proposed hybrid system, as in chapter 4. The hyperbolic sine of the adjacency matrix can be formulized as in Eq. (4.4). Since the hyperbolic sine of the biggest eigenvalue can be not a number. Then, we need to normalize $\boldsymbol{\Lambda}$ with dividing the biggest eigenvalue. After the normalization of $\boldsymbol{\Lambda}$, the Eq. (4.4) is reformulated as;

$$sinh(\boldsymbol{A}) = \boldsymbol{U} \cdot sinh(\frac{\boldsymbol{\Lambda}}{the\ biggest\ \boldsymbol{\Lambda}}) \cdot \boldsymbol{U}^T . \tag{5.2}$$

In another approachment, the eigenvalue vector $\boldsymbol{\Lambda}$ can be normalized as in the range of [0,1]. However, the eigenvalue vector $\boldsymbol{\Lambda}$ includes negative values, and when the $\boldsymbol{\Lambda}$ normalizes as in the range of [0,1], the negative information will be dismissed. Hence,

the Λ have been normalized with dividing the biggest eigenvalue to do not lose any information, (see in Eq. (5.2)).



(a)

(b) (c)

Figure 5.2. *(a) Item-feature matrix, (b) the generation of a user feature vector (c) users feature matrix.*

5.3. Evaluation Metrics

The following three representative top-N recommendation measures are used for evaluation. These are hit-ratio, precision and recall measurements, same as in [22, 23].

5.3.1. Hit-Ratio

The hit-ratio is measured by looking at the number of hits; i.e., the number of items in the test set that are also presented in the top-N recommendation item list returned for each customer/user. Then, the hit-ratio of the recommender system can be represented as:

$$Hit-Ratio(N)=\frac{\#hits}{n} \tag{5.3}$$

where n is represented the total number of users, $\#hits$ is symbolized as the overall hit of the recommendation system.

5.3.2. Recall

The percentage of items in the test set that are also presented in a user's the top-N recommendation list. The recall of the recommender system can be represented as:

$$Recall(N) = \frac{\#hits}{|T|}\ , \tag{5.4}$$

where $|T|$ is the number of test ratings.

5.3.3. Precision

The percentage of correctly recommended items in the test set that are also presented in a user's the top-N recommendation list. The precision of the recommended system can be indicated as:

$$Precision(N) = \frac{\#hits}{|T| \cdot N}\ , \tag{5.5}$$

where $|T|$ is the number of test ratings, N is the length of the recommendation list.

5.4. Experimental Evaluation and Datasets

The proposed hybrid algorithm and other comparison methods are implemented in a real-world Amazon review dataset. This dataset consists user interactions (review, rating, helpfulness votes, etc.) on items as well as the item metadata descriptions, price, brand, image features, etc. on 24 product categories spanning May 1996 – July 2014. Also, each product category covers a sub-dataset. Three product categories that have different sizes and density levels, are adopted/utilized for evaluation and these categories are: Beauty, Cell Phone, and Clothing. The standard five-core datasets (subsets) of these three categories are utilized for the experiment as in [22]. The 5-core data is a subset of the data in which all users and items have at least 5 reviews and ratings. Some statistics of these three datasets are shown in Table 5.1.

The number of interactions in Table 5.1 refers to the total number of ratings for each dataset. Each product/item is accompanied by an image, which has already been processed into a 4096-dimensional AlexNet feature vector in these datasets. Then, these vectors are utilized to generate the item representation, and also user representation, as explained in the generation process of item feature and user feature vectors. These visual feature vectors are extracted by using Alex Net, as in [45].

The testing methodology for the proposed hybrid recommendation algorithm is the same as in a previous study [22, 23]. The ratings are split by two subsets that are named by training and test sets, for each dataset. The test set includes only 5-star ratings and only items that are relevant to the corresponding users. The generation procedure of the training set and the test set can be defined as follows: In the first step, 30% number of items rated by each user are randomly selected to generate a temporary test set, while the temporary training set includes other ratings. In the second step, the 5-star ratings in the temporary test set are selected to generate the final test set, and the remaining ratings in the temporary test set are combined into the temporary training set to construct the final training set. Then, the training set is utilized to predict ratings or relational dualities (like or dislike) for each item-user pair. Futhermore, the 5-core datasets, that are utilized for the experiments, have at least 5 interactions for each user. Thus at least 3 interactions are utilized to per user for training, and at least 2 interactions are utilized to per user for testing. Moreover, all test items are selected as the same as in the previous study [22].

Table 5.1. *Statistics of the 5-core datasets.*

Datasets	#Users	#Items	#Interactions	Density
Clothing	39387	23033	278677	0.0307%
Cell Phones	27879	10429	194493	0.0669%
Beauty	22363	12101	198502	0.0734%

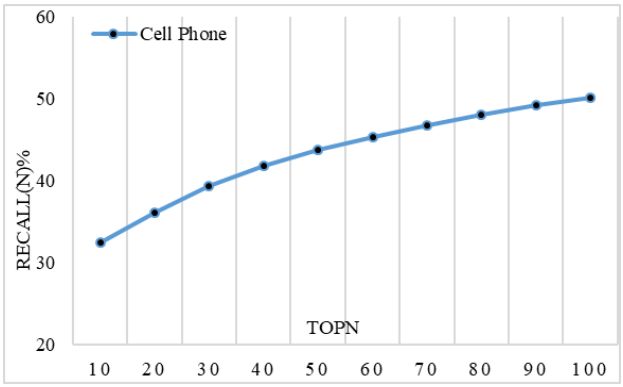
The performance comparison of the proposed hybrid (Hybrid-SIMLP) recommendation algorithm and previous hybrid recommendation methods Collaborative Filtering with Knowledge Graph (CFKG) and Joint Representation Learning (JRL) framework for top-10 recommendation is given in Table 5.2. The results of CFKG and

JRL method are taken from [22, 23]. Since the implementation of these methods is available in [57]. Moreover, the performance comparison of the Hybrid-SIMLP algorithm and rating-based CORLP algorithm is demonstrated in in Table 5.2. Since, the advantages of graph-based hybrid recommendation (Hybrid-SIMLP) algoritm is investigated. The results indicate that the performance of the proposed hybrid recommendation method is better than CFKG and JRL with using three quality metrics: hits ratio, recall, and precision on the Beauty, Clothing and Cell Phone datasets. Furthermore, the experimental results of CORLP method is the worst. Hence, it can be observed that hybrid recommendation algorithms that use the integration of multiple types of input data performs better than previous recommendation algorithms which only utilize user-item interaction matrix.

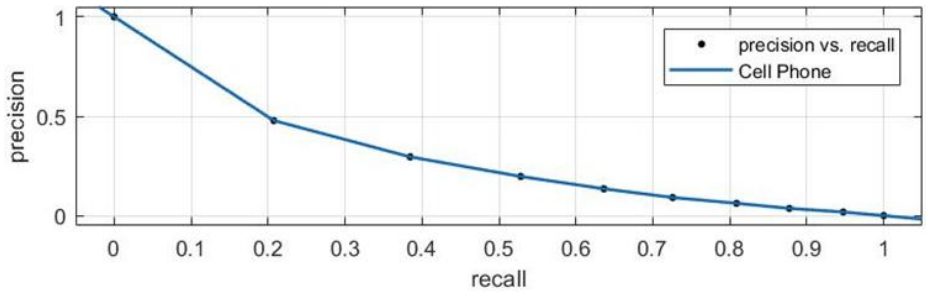
Table 5.2. *The performance comparison of the proposed algorithm and previous methods for top-10 recommendation.*

Datasets	Beauty			Clothing			Cell Phones		
Measures (%)	Recall	Hit Ratio	Precision	Recall	Hit Ratio	Precision	Recall	Hit Ratio	Precision
CFKG [23]	10,34	17,13	1,96	5,47	7,97	0,76	9,50	13,46	1,33
JRL [22]	6,95	12,78	1,55	2,99	4,64	0,44	7,51	10,94	1,1
CORLP [29]	4,76	4,23	0,43	5,35	6,03	0,54	2,75	4,03	0,28
Hybrid-SIMLP	**29,1**	**49,15**	**2,91**	**28,59**	**25,63**	**2,56**	**32,45**	**46,52**	**3,25**

The recall(N) results of the proposed hybrid recommendation algorithm on Cell Phone, Beauty, and Clothing datasets are drawn respectively in Figure 5.3 (a), Figure 5.4 (a) and Figure 5.5 (a). The recall(N) and precision(N) results of the proposed hybrid recommendation algorithm are indicated by utilizing the top-N recommendation task, where N is ranged from 10 to 100. Moreover, precision-versus-recall comparison of the results is drawn in Figure 5.3 (b), Figure 5.4 (b) and Figure 5.5 (b).

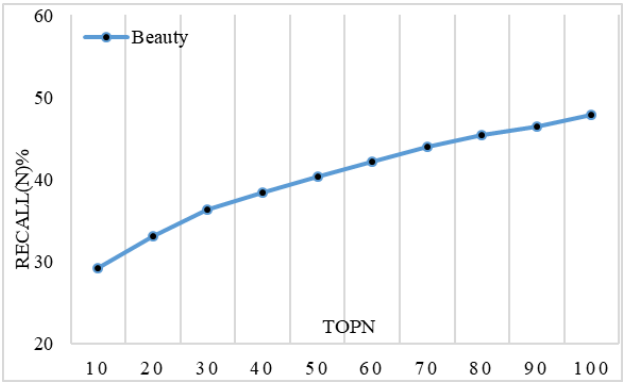


(a)

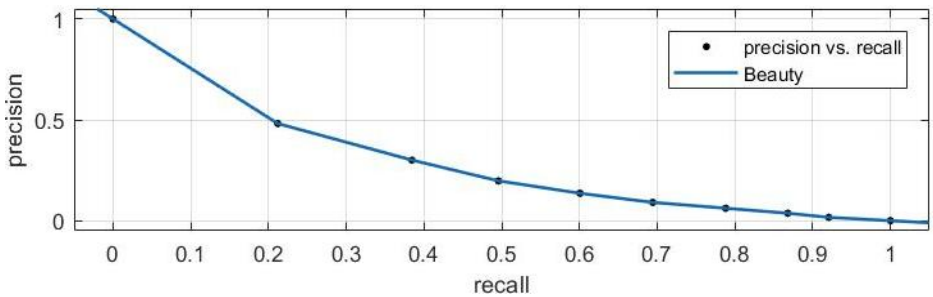


(b)

Figure 5.3. *Cell Phone: (a) recall-at-N and (b) precision-versus-recall on all items.*

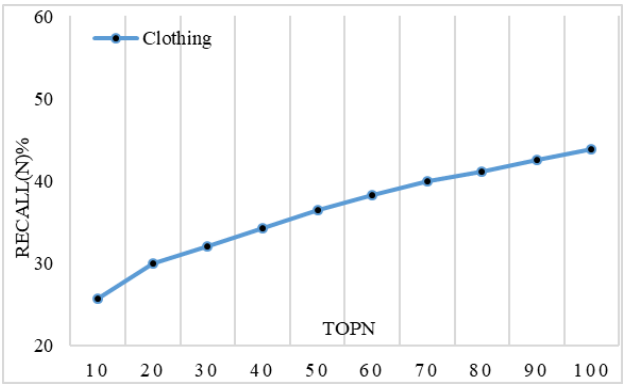


(a)

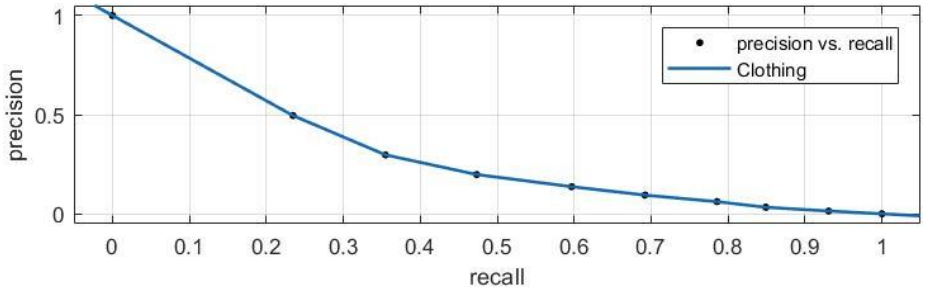


(b)

Figure 5.4. *Beauty: (a) recall-at-N and (b) precision-versus-recall on all items.*



(a)



(b)

Figure 5.5. *Clothing: (a) recall-at-N and (b) precision-versus-recall on all items.*

5.5. Conclusion

In this chapter, a graph-based hybrid recommendation algorithm that is utilized different information sources (images and ratings) are proposed for personalized recommendation. Also, a user-item interaction graph is build as incorporating both user behaviors for each user-item pair and the visual item representations. Furthermore the users' visual preference are exposed towards different items' visual images to generate each users' visual features. The proposed hybrid recommendation method is based on a link prediction approach with the weights in the bipartite signed graph represented by complex numbers. Fisrtly, similarity factors between users or items are evaluated from the user-visual feature or item- visual feature matrices, the inclusion of similarity factors yields the proposed algorithm more powerful. The proposed system reveals the significant potential of performance improvement in top-N recommendation tasks. Experimental results on real-world datasets demonstrated the superior performance of our approach, as well as its flexibility to incorporate different information sources. The experimental results show that the performance of the proposed method is better than CORLP [29],

CFKG [22], and JRL [23] with using three quality metrics: hit-ratio, recall, and precision on the three subsets of Amazon dataset. Finally, it is concluded that hybrid recommendation algorithms that use the integration of multiple types of input data performs better than previous recommendation algorithms which only utilize one type of input data.

Finally, it is concluded that the proposed method deals well with the deficiencies in hybrid recommender systems. Also, it can be observed that the proposed recommender system is more feasible than other methods, on real-world dataset. As for directions for future work, the usage of other number systems for the proposed link prediction method can be investigated, such as the quaternions.

6. A SIMILARITY-INCLUSIVE LINK PREDICTION APPROACH FOR ITEM RECOMMENDATION WITH QUATERNIONS

6.1. Quaternions

The quaternions is first discovered by William Rowan Hamilton and they are members of a noncommutative division algebra. The idea for quaternions occurred his mind when he was walking along the Royal Canal on his way to a meeting of the Irish Academy, and Hamilton was so pleased with his exploration that he scratched the fundamental formula of quaternion algebra into the stone of the Brougham bridge [46]. The formula of quaternion algebra can be mathematically stated as:

$$i^2 = j^2 = k^2 = ijk = -1, \tag{6.1}$$

The quaternions are a single example of a more general class of hyper complex numbers invented by Hamilton and the set of quaternions is represented as H, H or Q_8. The quaternions are associative, while they are not commutative. Furthermore, the quaternions can be formed as a group, that is known as the quaternion group. The graphical representation of quaternion units as 90°-rotations in the planes is given in Figure 6.1. This figure is drawn in [47], firstly.

By analogy with the complex form, complex numbers can be represented as a sum of real and imaginary parts, $a + i \cdot b$, hence a quaternion number can also be denoted as a linear combination of real and imaginary parts;

$$\mathrm{H} = a + b \cdot i + c \cdot j + d \cdot k \tag{6.2}$$

Moreover, H can be represented as: $\mathrm{H} = (w, v) = w + x \cdot i + y \cdot j + z \cdot k$, when w is real (scalar), v is imaginary (vector) part.

$$\mathrm{H} = \begin{bmatrix} w \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} w \\ v \end{bmatrix} \begin{matrix} scalar \ \ part \\ vector \ \ part \end{matrix} \tag{6.3}$$

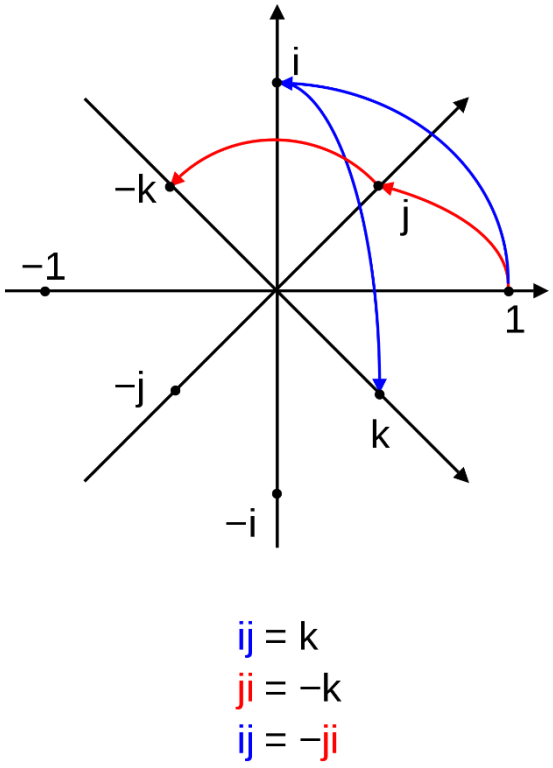


Figure 6.1. *The graphical representation of quaternion units as 90°-rotations in the planes [47].*

The skew-symmetric matrix formula of complex quaternion Q is obtained by using 4×4 matrix representations of real quaternion basis elements and this representation is introduced in [71,72]. The complex quaternionic matrix formalism provides applicable and elegant representations, and this representation has some advantages with simpler and more expressive properties [71].

The quaternion based multi-valued architecture is introduced in some research fields and demonstrated that it has potential with numerical examples of multi-channel prediction and classification [48,49]. A variety of real-valued learning structures have been presented in prior literature, hence multi-valued architecture is shed light on due to deal with their drawbacks. A better way to represent multidimensional data is utilizing quaternions [48]. The purpose of this representation is because of a four-dimensional associative normed division algebra over the real numbers allows the multiplication and division of points in three-dimensional space [48].

Furthermore, an adaptive method for tag-rating based recommender system is introduced in [50]. A term-association matrix is represented to describe the relation between tags and items properties in this approach. Quaternions are used for the definition

of the term-association matrix, and the components of this matrix are users, items, tags, and ratings as each part of a quaternion number. A privileged matrix factorization method for CF by utilizing the quaternions is introduced in [51]. Besides that, this method is utilized by review texts that are in companion with rating values to assist the learning of user and item factors/representation. This recommendation algorithm is also considered as a rating prediction problem based on the quaternions. A user representation, an item representation, a rating, and a review are denoted as each part of a quaternion number in this algorithm [51].

In the previous chapter 4, rating conversion is implemented to generate an adjacency matrix based on the representation of complex numbers with real and imaginary parts in the proposed SIMLP algorithm. In this algorithm, similar or dissimilar links were weighted by real numbers, whereas like or dislike links were weighted by complex numbers [52]. The problem of recommendation generation is considered as a link prediction problem, due to the complex numbers yield a natural algebraic link between real and imaginary parts. Moreover, the available link prediction algorithms may reprocess with the proposed SIMLP method without any modifications. In this chapter, the proposed SIMLP algorithm is reformulated to depend on the representation of quaternion numbers with a scalar and imaginary vector part in the quaternion form. In this proposed method, the similar valued links are denoted as a scalar part, and the dissimilar, like and dislike valued links are denoted as imaginary vector part of the quaternion. As a quaternion number provides a link between real and imaginary vector parts in the bipartite graph model, the problem of recommendation generation still can be considered as a link prediction problem. Besides that, the available link prediction algorithms can operate with the proposed quaternion based recommendation method as in the SIMLP method.

6.2. Quaternion-Based Triangle Closing Model

In this chapter, a quaternion based triangle closing model is proposed depending on the graph models, which are introduced in [37, 58, 59]. Moreover, the new design of triangle closing model based on the social graph models is presented in [37]. The extended version of this model is recommended based on the usage of other number systems to identify each edges/links, such as the quaternions or the complex number systems. The

only possible relationship in a social graph depends on the friendship [37]. Then, the social recommendation problem can be considered as recommending new friends based on existing friendships. The fundamental model utilized for this purpose can be considered as the major principle of triangle closing model: people who have (maybe many) common friends might be friends themselves. Figure 6.2. (a) illustrates this principle of triangle closing model. Two adjacent friend edges let us predict a new friend edge, hence "*The friend of my friend is my friend*", this rule is given in Figure 6.2 (a). Another triangle closing principle in a social graph with friend and foe relationships is illustrated in Figure 6.2 (b). In such a social graph, new edges can be predicted using the principle that can be stated as "*The enemy of my enemy is my friend*", [37]. Moreover, these two principles can be converted for the user-item interaction graph, by utilizing the user-user and item-item similar and dissimilar relationships. Hence, two adjacent similar links let us predict a new similar link. Furthermore, two adjacent dissimilar links let us predict a new similar link, illustrated in Figure 6.3 and 6.4.

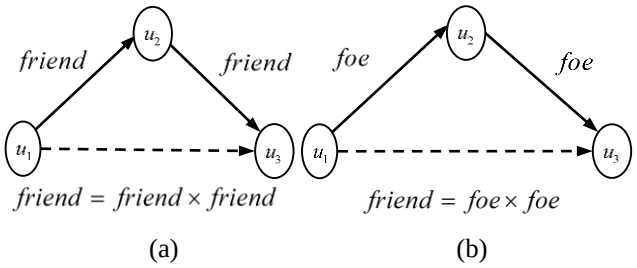


Figure 6.2. *(a) Triangle closing model with only the friend relationship, (b) triangle closing model with friend and foe relationship.*

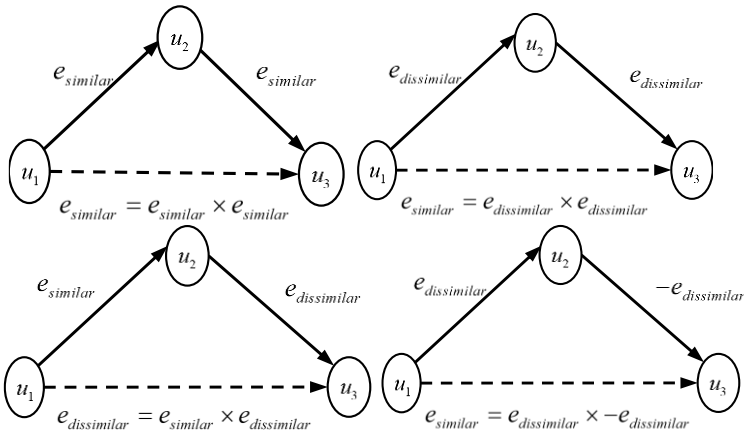


Figure 6.3. *The triangle closing principle is illustrated as the multiplication rule between similar/dissimilar relationships, for only three user nodes*

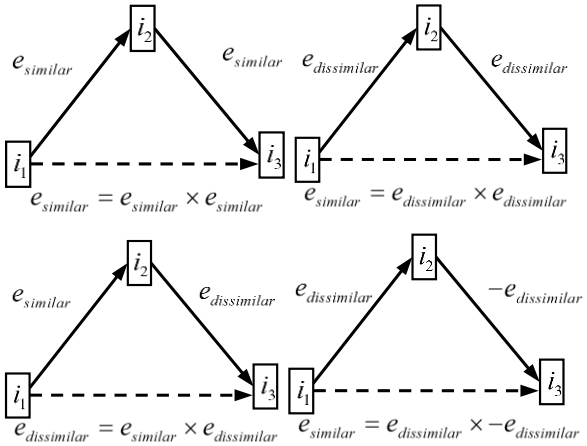


Figure 6.4. *The triangle closing principle is illustrated as the multiplication rule between similar/dissimilar relationships, for only three item nodes*

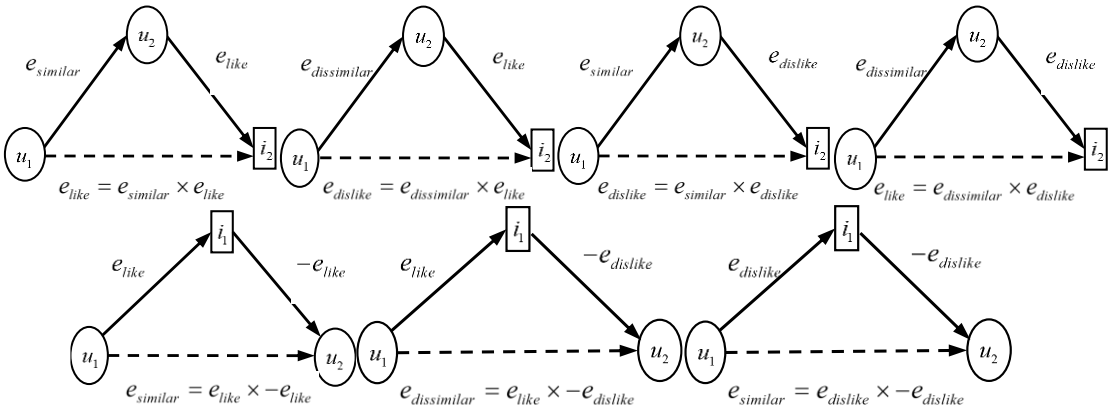


Figure 6.5. *The triangle closing principle is illustrated as the multiplication rule between similar/dissimilar and like/dislike relationships, for two users and an item nodes*

The triangle closing model can be generated with four different combinations. First of all, the vertices of the triangle model may only be constructed with users nodes, it means that the triangle model is generated with three user nodes. This triangle model has two types of relationships. For the user-user links, there is a similarity factor, $e_{similar}$ or $e_{dissimilar}$ between two entities. This triangle model is illustrated in Figure 6.3. Similarly, the vertices of the triangle model may be generated with only item nodes, it means that the triangle model is generated with three item nodes. In a similar manner, this triangle model has two different relationships. There is a similarity factor $e_{similar}$ or $e_{dissimilar}$ in this triangle model for the item-item links. This triangle model is illustrated in Figure 6.4.

Secondly, the vertices of the triangle model may be generated with two users and an item nodes. For the user-item links, there is a similarity factor, e_{like} or $e_{dislike}$, between a user and an item nodes. Due to the necessity of recognizing the asymmetry between the user and the item, the triangle model includes item-user links, then there is a similarity factor, $-e_{like}$ or $-e_{dislike}$, between item and user nodes. Accordingly, in the case of a link from the user u to item i with the weight e_{like} or $e_{dislike}$, there is always a reverse link from item i to the user u with a weight of $-e_{like}$ or $-e_{dislike}$. Moreover, there is a similarity factor, $e_{similar}$ or $e_{dissimilar}$, between two users nodes for the user-user links. This triangle model is illustrated in Figure 6.5.

Lastly, the vertices of the triangle model may be generated with a user and two item nodes. Similarly, there is a similarity factor, e_{like} or $e_{dislike}$, for the user-item links and $-e_{like}$ or $-e_{dislike}$ for the item-user links among user nodes to item nodes. Furthermore, there is a similarity factor, $e_{similar}$ or $e_{dissimilar}$, between two items nodes for the item-item links. This triangle model is illustrated in Figure 6.6.

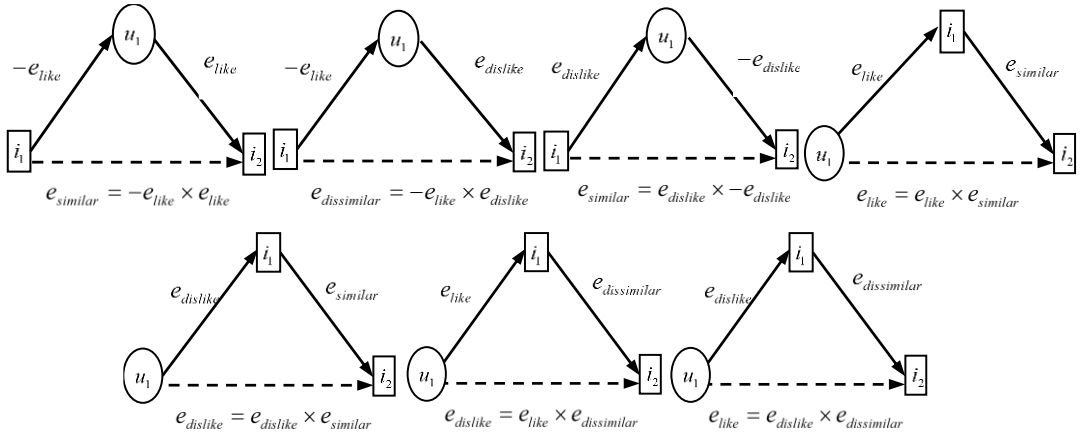


Figure 6.6. *The triangle closing principle is illustrated as the multiplication rule between similar/dissimilar and like/dislike relationships, for two items and a user node*

In the quaternion-based triangle model, e_{like}, $e_{dislike}$ and $e_{similar}$, $e_{dissimilar}$ are normalized values just for the weights. The entire triangle closing rule in this model may be described as shown in Figure 6.7. This rule has four parts: the triangle model consists of three user nodes and their relations (see Figure 6.3), the triangle model consists of three item nodes and their relations (see Figure 6.4), the triangle model is consisted of two users and an

item nodes and their relations (see Figure 6.5), and also the triangle model consists of a user and two items nodes and their relations (see Figure 6.6). These are the main ideas of CF from a different viewpoint, [29, 52]. Thus, these multiplication principles of this model may be mathematically stated as follows:

$$e_{similar} = -e^2{}_{like} = -e^2{}_{dislike} = -e^2{}_{dissimilar},\ e_{like} = e_{similar} \cdot e_{like} = e_{dissimilar} \cdot e_{dislike},$$

$$e_{dislike} = e_{dissimilar} \cdot e_{like} = e_{similar} \cdot e_{dislike},\ e_{dissimilar} = e_{like} \cdot -e_{dislike} = -e_{like} \cdot e_{dislike}. \qquad (6.4)$$

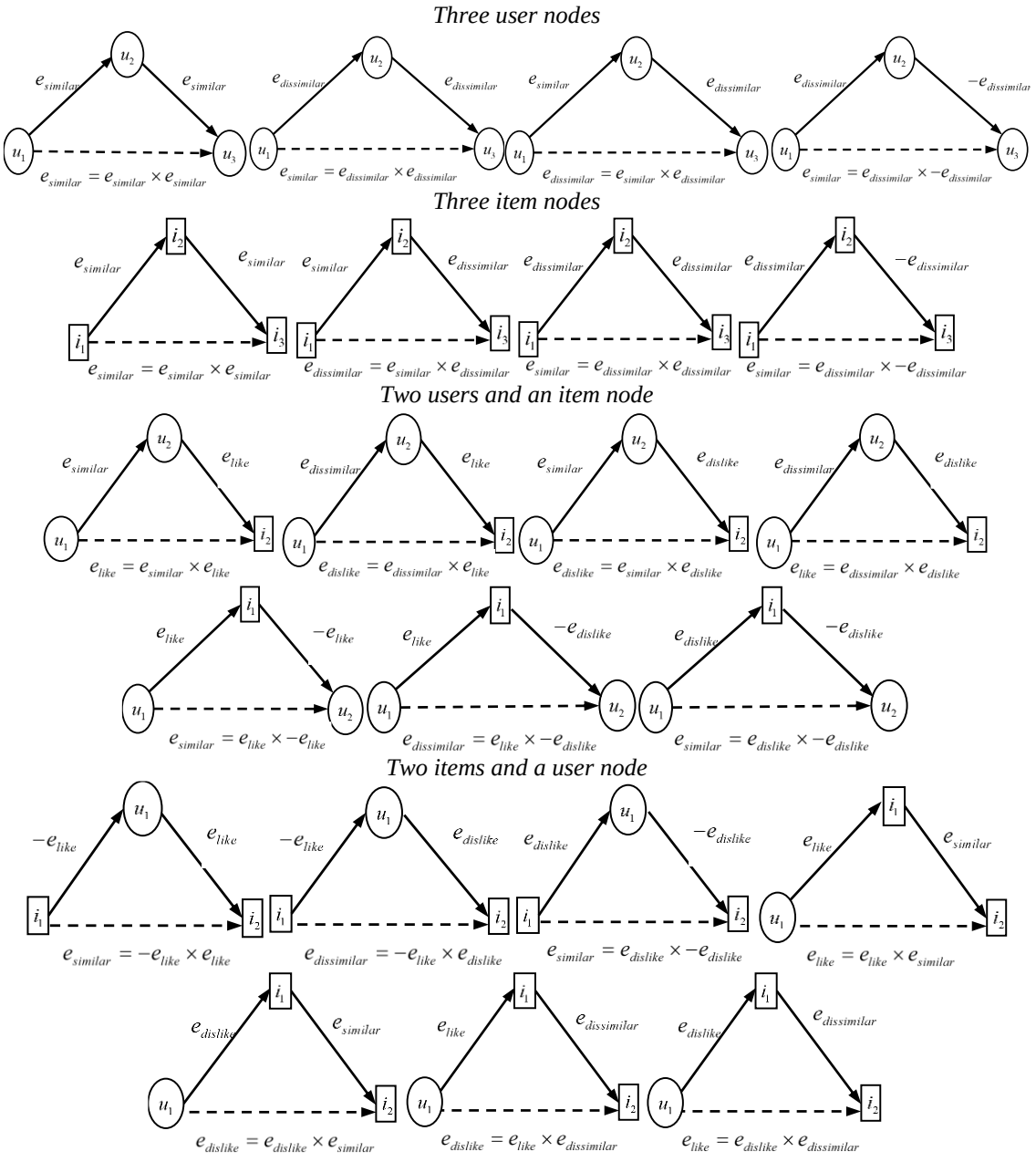


Figure 6.7. *The triangle closing principle is illustrated as the multiplication rule between similar/dissimilar and like/dislike relationships*

Therefore, to solve this system of equations (Eq. 6.4), we need to find four different and nonzero constants, which are $e_{similar}$, $e_{dissimilar}$ and e_{like}, $e_{dislike}$. Quaternion numbers offer an easy way to solve this system of equations, when we set $e_{like} = i, e_{dislike} = j, e_{dissimilar} = k$ and $e_{similar} = 1$, where i, j, k are the imaginary unit vector. The requirements may be mathematically stated as follows;

$$i^2 = j^2 = k^2 = ijk = -1 \text{ and } 1 = 1^2. \tag{6.5}$$

In this symbolization, a link has endpoints of the same type, two items or two users, may be weighted with a real number if there is a similarity factor $e_{similar}$. It means that the more similar the endpoints have the higher such value. A link has endpoints of the same type, two items or two users, may be weighted with an imaginary weight k if there is a dissimilarity factor $e_{dissimilar}$. It means that the more dissimilar the endpoints have the higher such value. Besides that, a link with an imaginary weight must be an item-user or user-item link based on the sign and interest. For instance, if the user u dislikes item i, then the link is weighted with j from u to i, and the reversed link is weighted with $-j$ from i to u. Equivalently, if the user u likes the item i, then the link is weighted with i from u to i, and the reversed link is weighted with $-i$ from i to u. As opposed to similar links, we may distinguish e_{like}, $e_{dislike}$ and $-e_{like}$ $-e_{dislike}$ only when the sign of link's weight and the direction of the link are known at the same time. Since the sign of similar link's weight is independent from the direction of the link and it can be concluded that the similar links are providing this rule;

$$\begin{aligned} e_{similar} &= -e_{similar}, \\ e_{dissimilar} &= -e_{dissimilar}. \end{aligned} \tag{6.6}$$

6.3. Recommendation Algorithm with Quaternions

In this chapter, the proposed signed graph model is expanded by using the representation of quaternion numbers with real and imaginary parts in the quaternion form. In this expanded model, the similar links are denoted as a scalar part of the quaternion, user-user and item-item similarity matrices and rating conversion of these matrices are illustrated in Figure 6.8. The user-item interaction matrix which is illustrated in Figure 4.1 (a), is utilized to compute user-user and item-item similarity matrices that drawn in Figure 6.8 (a) and (b). The similarity values are computed as hypothetically,

since the only ratio of similar ratings are calculated as the similarity values between users or items as in Chapter 4. Furthermore, the values of the dissimilar links are weighted as *1- similar* link values are illustrated in Figure 6.9 (a). After that, these similar and dissimilar links values are passed through a threshold at 0.5. Then, the dissimilar links are multiplied by k and these links denoted as k in the imaginary part of the quaternions is illustrated in Figure 6.9 (b).

User\User	User1	User2	User3	User4	...
User1	1	0.4	0.6	0.8	
User2	0.4	1		0.2	0.2
User3	0.6		1	0.6	
User4	0.8	0.2	0.6	1	
...		0.2			1

User/User	User1	User2	User3	User4	...
User1	1	0	1	1	
User2	0	1		0	0
User3	1		1	1	
User4	1	0	1	1	
...		0			1

(a)

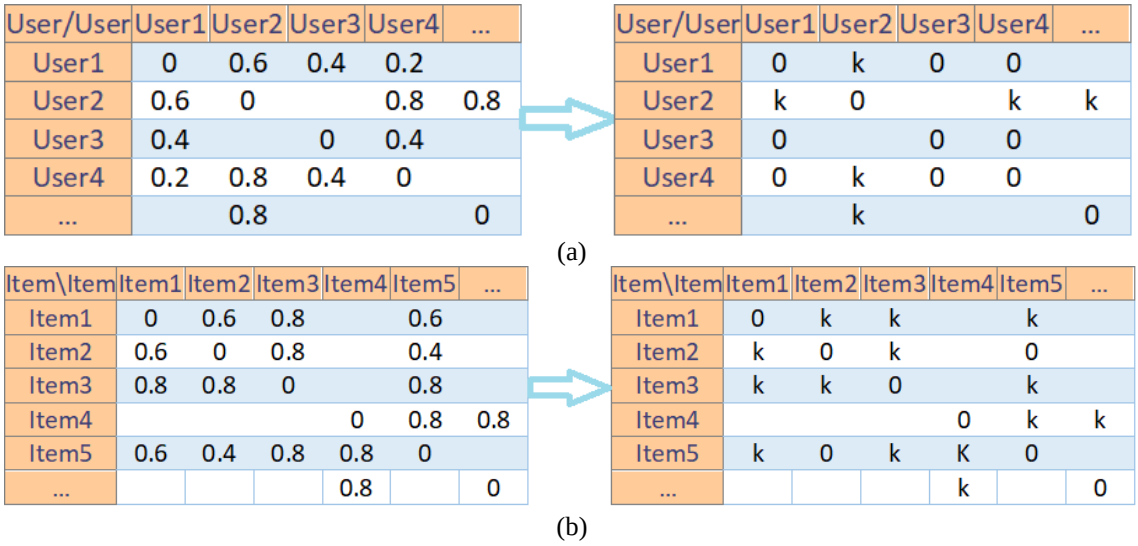
Item\Item	Item1	Item2	Item3	Item4	Item5	...
Item1	1	0.4	0.2		0.4	
Item2	0.4	1	0.2		0.6	
Item3	0.2	0.2	1		0.2	
Item4				1	0.2	0.2
Item5	0.4	0.6	0.2	0.2	1	
...				0.2		1

Item\Item	Item1	Item2	Item3	Item4	Item5	...
Item1	1	0	0		0	
Item2	0	1	0		1	
Item3	0	0	1		0	
Item4				1	0	0
Item5	0	1	0	0	1	
...				0		1

(b)

Figure 6.8. *(a) user-user similarity matrix and rating conversion of this matrix, (b) item-item similarity matrix and rating conversion of this matrix*

Moreover, the like and dislike links are denoted as i and j respectively in the imaginary part of the quaternions. The quaternion based rating conversion of the user-item matrix is illustrated in Figure 6.10. Lastly, an example of the main quaternion-based adjacency matrix after rating conversion process is given in Figure 6.11. After this the quaternion-based rating conversion process, the problem of recommendation generation still can be considered as a link prediction problem. Since the quaternion-based recommendation algorithm is an expansion of the SIMLP method and is named by Quaternion based similarity inclusive link prediction (Q-SIMLP) method. The quaternion based representation's validity and efficiency are assessed by evaluating the performance of the Q-SIMLP recommendation approach in two real-world datasets as same as in Chapter 4.



User/User	User1	User2	User3	User4	...
User1	0	0.6	0.4	0.2	
User2	0.6	0		0.8	0.8
User3	0.4		0	0.4	
User4	0.2	0.8	0.4	0	
...		0.8			0

User/User	User1	User2	User3	User4	...
User1	0	k	0	0	
User2	k	0		k	k
User3	0		0	0	
User4	0	k	0	0	
...		k			0

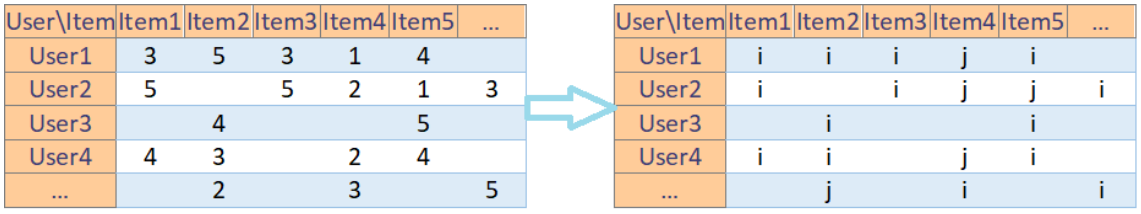
(a)

Item\Item	Item1	Item2	Item3	Item4	Item5	...
Item1	0	0.6	0.8		0.6	
Item2	0.6	0	0.8		0.4	
Item3	0.8	0.8	0		0.8	
Item4				0	0.8	0.8
Item5	0.6	0.4	0.8	0.8	0	
...				0.8		0

Item\Item	Item1	Item2	Item3	Item4	Item5	...
Item1	0	k	k		k	
Item2	k	0	k		0	
Item3	k	k	0		k	
Item4				0	k	k
Item5	k	0	k	K	0	
...				k		0

(b)

Figure 6.9. *(a) user-user dissimilarity matrix and rating conversion of this matrix, (b) item-item dissimilarity matrix and rating conversion of this matrix*



User\Item	Item1	Item2	Item3	Item4	Item5	...
User1	3	5	3	1	4	
User2	5		5	2	1	3
User3		4			5	
User4	4	3		2	4	
...		2		3		5

User\Item	Item1	Item2	Item3	Item4	Item5	...
User1	i	i	i	j	i	
User2	i		i	j	j	i
User3		i			i	
User4	i	i		j	i	
...		j		i		i

(a) (b)

Figure 6.10. *(a) user-item rating matrix, (b) quaternion based rating conversion of this matrix based on like and dislike relationships*

User/User	User1	User2	User3	User4	...	Item1	Item2	Item3	Item4	Item5	...
User1	1	k	1	1		i	i	i	j	i	
User2	k	1		k	k	i		i	j	j	i
User3	1		1	1			i			i	
User4	1	k	1	1		i	i		j	i	
...		k			1		j		i		i
Item1	-i	-i		-i		1	k	k		k	
Item2	-i		-i	-i	-j	k	1	k		1	
Item3	-i	-i				k	k	1		k	
Item4	-j	-j		-j	-i				1	k	k
Item5	-i	-j	-i	-i		k	1	k	k	1	
...		-i			-i				k		1

Figure 6.11. *The main quaternion based adjacency matrix*

6.3.1. Quaternion-Based Adjacency Matrix

The adjacency matrix $\boldsymbol{A}$ is expanded as a quaternion matrix, and this matrix can be mathematically formulized as:

$$\boldsymbol{A} = \boldsymbol{A}_{similar} + \boldsymbol{A}_{like} \cdot i + \boldsymbol{A}_{dislike} \cdot j + \boldsymbol{A}_{dissimilar} \cdot k, \tag{6.7}$$

where the combination of item-item similarity $\boldsymbol{A}_{II}$ and user-user similarity matrices $\boldsymbol{A}_{UU}$ is denoted as $\boldsymbol{A}_{similar}$, the combination item-item dissimilarity $1_{n\times n} - \boldsymbol{A}_{II}$ and user-user dissimilarity matrices $1_{m\times m} - \boldsymbol{A}_{UU}$ are denoted as $\boldsymbol{A}_{dissimilar}$, the user-item preference matrix is denoted as $\boldsymbol{A}_{UI}$, with using the both $\boldsymbol{A}_{like}$, $\boldsymbol{A}_{dislike}$ relationships. Moreover, the conjugate transpose of $\boldsymbol{A}_{UI}$ can be described as same as in Chapter 4, $\boldsymbol{A}_{IU} = -\boldsymbol{A}^{T}_{UI}$. The preference matrices $\boldsymbol{A}_{like}$, $\boldsymbol{A}_{dislike}$ and the dissimilarity matrix $\boldsymbol{A}_{dissimilar}$ are complex matrices, while the similarity matrix $\boldsymbol{A}_{similar}$ is a real matrix. User-item preference matrix $\boldsymbol{A}_{UI}$ is illustrated in Figure 6.10 (b) and user-user similarity matrices $\boldsymbol{A}_{UU}$ are illustrated in Figure 6.8 (a), item-item similarity matrices $\boldsymbol{A}_{II}$ is shown in Figure 6.8 (b). The example of user-user dissimilarity matrix $1_{m\times m} - \boldsymbol{A}_{UU}$ are illustrated in Figure 6.9 (a), item-item dissimilarity matrix $1_{n\times n} - \boldsymbol{A}_{II}$ is shown in Figure 6.9 (b). Finally the combination of all these matrices which gives the main quaternion based adjacency matrix is shown in Figure 6.11.

The proposed Q-SIMLP algorithm differs slightly from the SIMLP based recommendation method in the modeling of the adjacency matrix, and while calculating the powers of the adjacency matrix and yielding the final recommendation are in the same procedure. User-user and item-item similarity and dissimilarity matrices of the user-item preference matrix is computed with utilizing cosine similarity measurement. After that, these similarity and dissimilarity factors are passed through from a threshold at 0.5. Then, the dissimilar links are multiplied by k and these links indicated as k in the imaginary part of the quaternions as stated in Eq. (6.7). Moreover, user-item like relational matrix is generated based on whether rating is greater than 3, as stated in Eq. (6.7). Also, user - item dislike relational matrix is generated based on whether rating is less than 3, as stated in Eq. (6.7). Following the summation of these matrices, the main adjacency matrix is built as in Eq. (6.7).

The components of quaternion based adjacency matrix are mathematically stated as;

$$\boldsymbol{A}_{similar}=\begin{pmatrix} u_{11} & \dots & u_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} & 0 & \cdots & 0 \\ 0 & \dots & 0 & t_{11} & \dots & t_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & t_{n1} & \cdots & t_{nn} \end{pmatrix}, \boldsymbol{A}_{like}=\begin{pmatrix} 0 & \dots & 0 & r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \dots & -r_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & 0 & \cdots & 0 \end{pmatrix},$$

$$\boldsymbol{A}_{dislike}=\begin{pmatrix} 0 & \dots & 0 & r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \dots & -r_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & 0 & \cdots & 0 \end{pmatrix}, \boldsymbol{A}_{dissimilar}=\begin{pmatrix} 1-u_{11} & \dots & 1-u_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1-u_{m1} & \cdots & 1-u_{mn} & 0 & \cdots & 0 \\ 0 & \dots & 0 & 1-t_{11} & \dots & 1-t_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1-t_{n1} & \cdots & 1-t_{nn} \end{pmatrix} \quad (6.8)$$

where u_{ij} denotes the similarity relationship between the i^{th} and j^{th} users, t_{ij} denotes the similarity relationship between the i^{th} and j^{th} items, r_{ij} expresses the like or dislike relationship between the i^{th} user and j^{th} item, and $-r_{ij}$ expresses the conjugate transpose of the like or dislike relationship between the i^{th} user and j^{th} item in Eq. (6. 8). When r_{ij} expresses the like relationship between the i^{th} user and j^{th} item, r_{ij} multiplied by *i*, an imaginary part of quaternions. Equivalently, if r_{ij} expresses the dislike relationship between the i^{th} user and j^{th} item, r_{ij} multiplied by j, an imaginary part of quaternions. Moreover, $1-u_{ij}$ expresses the dissimilarity relationship between the i^{th} user and j^{th} users, and $1-t_{ij}$ expresses the dissimilarity relationship between the i^{th} item

and j^{th} items as in Eq. (6. 8). Equivalently, $1-u_{ij}$ and $1-t_{ij}$ are multiplied by k as an imaginary part of quaternions.

After the summation of these matrices, the main adjacency matrix $\boldsymbol{A}$ is built as in Eq. (6.9).

$$\boldsymbol{A}=\begin{pmatrix} u_{11} & \dots & u_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mm} & 0 & \cdots & 0 \\ 0 & \dots & 0 & t_{11} & \dots & t_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & t_{n1} & \cdots & t_{nn} \end{pmatrix}+\begin{pmatrix} 0 & \dots & 0 & r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \dots & -r_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & 0 & \cdots & 0 \end{pmatrix}\cdot i+$$
$$\begin{pmatrix} 0 & \dots & 0 & r_{11} & \dots & r_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{m1} & \cdots & r_{mn} \\ -r_{11} & \dots & -r_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -r_{m1} & \cdots & -r_{mn} & 0 & \cdots & 0 \end{pmatrix}\cdot j+\begin{pmatrix} 1-u_{11} & \dots & 1-u_{1n} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1-u_{m1} & \cdots & 1-u_{mn} & 0 & \cdots & 0 \\ 0 & \dots & 0 & 1-t_{11} & \dots & 1-t_{1n} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1-t_{n1} & \cdots & 1-t_{nn} \end{pmatrix}\cdot k \tag{6.9}$$

Moreover, this adjacency matrix is a square matrix. Still, eigenvalue decomposition can be used on this adjacency matrix in Eq. (6.9). In the proposed quaternion based method with another approachment, the link prediction function can be multiplied by a parameter α, then the predictions applied to $\boldsymbol{A}$ can be represented as:

$$P(\alpha\boldsymbol{A})=\lambda\cdot\alpha\boldsymbol{A}+\lambda_1\cdot(\alpha\boldsymbol{A})^3+\lambda_2\cdot(\alpha\boldsymbol{A})^5+\lambda_3\cdot(\alpha\boldsymbol{A})^7+\dots \tag{6.10}$$

6.3.2. Quaternion-Based Similarity Inclusive Link Prediction Method

The rating conversion is necessary for the quaternion-based adjacency matrix's generation to the proposed Q-SIMLP method, where the ratings/values in the user-item rating matrix are converted to i or j based on whether the rating is greater than or equal to 3. In this direction, if the rating is less than 3, it is changed by j, which means that the user states 'dislike' for the item; equivalently, imaginary value i is given to defining 'like', while the rating is greater than or equal to 3. Moreover, when the user-item pair

(u,i) is not included in the training set, the corresponding component of the adjacency matrix is equal to zero. Following that, the user-user similarity and item-item similarity matrices of the preference matrix are generated with utilizing cosine similarity measure to calculate similarity values. On the other hand, we find the user-user dissimilarity and item-item dissimilarity matrices with utilizing the user-user similarity and item-item similarity matrices of the preference matrix respectively, as stated in Eq. 6.8. Then, the components of similarity matrices and dissimilarity matrices of the preference matrix are passed through a threshold at 0.5, hence these matrices include only binary values. The similarity matrices are represented as a scalar part of the quaternion-based adjacency matrix, as formulated in Eq. 6.8. The user-user dissimilarity and item-item dissimilarity matrices are multiplied by k, and these matrices are taken as one of the imaginary parts of the adjacency matrix. Following the summation of the like relationships and dislike relationships matrices and the dissimilarity matrices are generated the entire imaginary part of the quaternion-based adjacency matrix.

Evaluation the powers of the quaternion-based adjacency matrix and providing the final recommendation are in the same procedure for the proposed Q-SIMLP algorithm as SIMLP algorithm. Thus, the hyperbolic sine function is considered as link prediction function for the proposed Q-SIMLP algorithm. Hence the closest values among the nodes are measured by the power sum of the adjacency matrix, the summation of each entry of the top-right and top-left components expresses the degree of whichever item is relevant to a specific user. Following the summation of odd powers of the adjacency matrix, the prediction scores, that denote item recommendation to a particular user, are obtained. The prediction scores are denoted as the summation of a scalar/real part and the imaginary part i of the entire scores. Since only the like relationships are taking into consideration for recommendation generation. The prediction scores are sorted in descending order; since the user will like the item if the score is positive or dislike the item when the score is negative. When the items score are positive and higher values, these items will be recommended to a particular user, while these recommended items are unnoticed by that user before. Furthermore, top-N recommendation lists are generated for each user by these sorted prediction scores [39].

6.3.3. Quaternion-Based Hybrid Recommender System

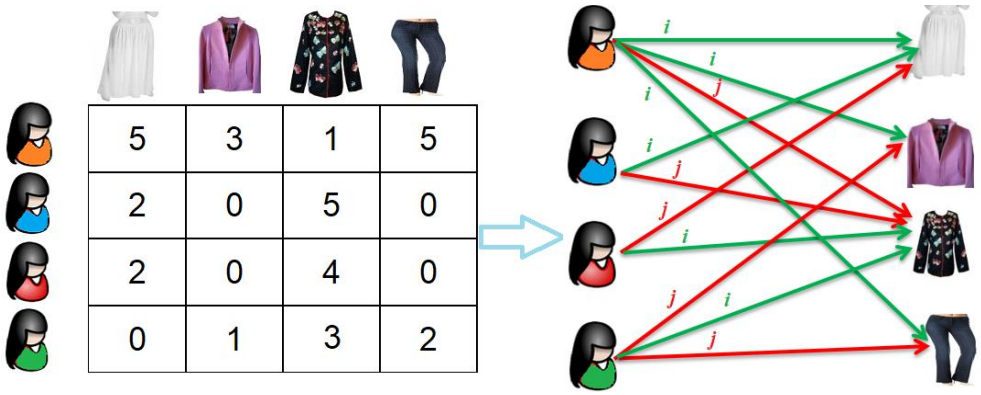
The quaternion based hybrid recommendation algorithm differs slightly from the proposed Q-SIMLP method in the modeling of the adjacency matrix. For the proposed hybrid recommender system, we have the knowledge about user-item ratings and also visual images of the entire items in the datasets. Hence, the proposed hybrid method benefits from items' visualization with utilizing AlexNet features, this feature extraction process is mentioned in the previous chapter 4. On the other hand, each users' visual feature vector is generated from using the users' preference towards different items' visual features. In the beginning, all items that a particular user noticed (rated or bought) before is found. Then, the AlexNet feature vectors of these items are extracted and these item feature vectors are summed up. Lastly, we calculate the mean of this summation with the number of items that users' rated/noticed before. Since each user can be represented as a 4096-dimensional visual feature vector, and a user visual-feature matrix can be generated for each dataset. Following the generation of the user visual-feature and item visual-feature matrices, we can find user-user and item-item similarity matrices with utilizing these feature matrices.

The quaternion based adjacency matrix generation for hybrid recommendation algorithm is modified with somewhat points from the proposed Q-SIMLP algorithm. However, the rating conversion part of the adjacency matrix has the same procedures in the generation of the adjacency matrix for the Q-SIMLP algorithm. Following that, the user-user similarity matrix is generated from the user visual-feature matrix with applying cosine similarity measures to evaluate similarity values. Besides that, the item-item similarity matrix is generated from the item visual-feature matrix with applying cosine similarity measures to compute similarity values. In other aspects, we find the user-user and item-item dissimilarity matrices with utilizing the user-user similarity and item-item similarity matrices respectively, as mentioned in Q-SIMLP method. Then, the components of similarity and dissimilarity matrices of the system are passed through a threshold at 0.5, since these matrices are consist of only binary values. As the same procedure for Q-SIMLP algorithm, we take the similarity matrices as a scalar part of the adjacency matrix, as formulated in Eq. 6.11. Also, the user-user and item-item dissimilarity matrices are multiplied by k, and these matrices are taken as one of the

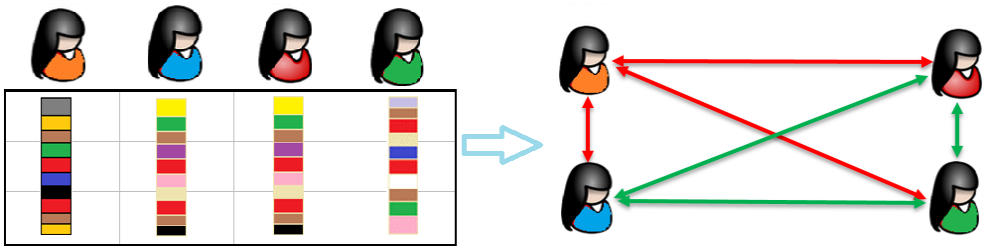
imaginary parts of the adjacency matrix. Following the summation of these matrices, the main adjacency matrix is built as in Eq. (6.11).

$$\boldsymbol{A} = \boldsymbol{A}_{visual\text{-}similar} + \boldsymbol{A}_{like} \times i + \boldsymbol{A}_{dislike} \times j + \boldsymbol{A}_{visual\text{-}dissimilar} \times k \tag{6.11}$$

This quaternion based adjacency matrix is a square matrix. Since we can use the same link prediction (hyperbolic) function on this adjacency matrix in Eq. (6.11), to evaluate the power sum of this matrix. Thus, the recommendation methodology adopted in this quaternion based hybrid recommender system is the same as in Q-SIMLP recommendation algorithm.



(a)



(b)

(c)

Figure 6.12. *(a) User-item rating matrix and bipartite signed graph, (b) user-feature matrix and user-user relationship graph, (c) item-feature matrix and item - item relationship graph*

An example of user-item signed graph generation process for quaternion-based hybrid recommender system is illustrated in Figure 6.12 and Figure 6.13. User-item rating matrix and bipartite signed graph model of this rating matrix are drawn in Figure 6.12 (a). The green links are represented as '*like'* edges and denoted as i, red links are represented as *'dislike'* edges denoted as j in the bipartite signed graph, (see in Figure 6.12 (a)). User-feature matrix and user-user relationship graph are drawn in Figure 6.12 (b). The green links are represented as user-user '*similar'* relationships, red links are represented as user-user *'dissimilar'* relationships in Figure 6.12 (b). Item-feature matrix and item- item relationship graph are drawn in Figure 6.12 (c). Finally, the generated user-item signed graph for quaternion based hybrid recommender system is drawn in Figure 6.13.

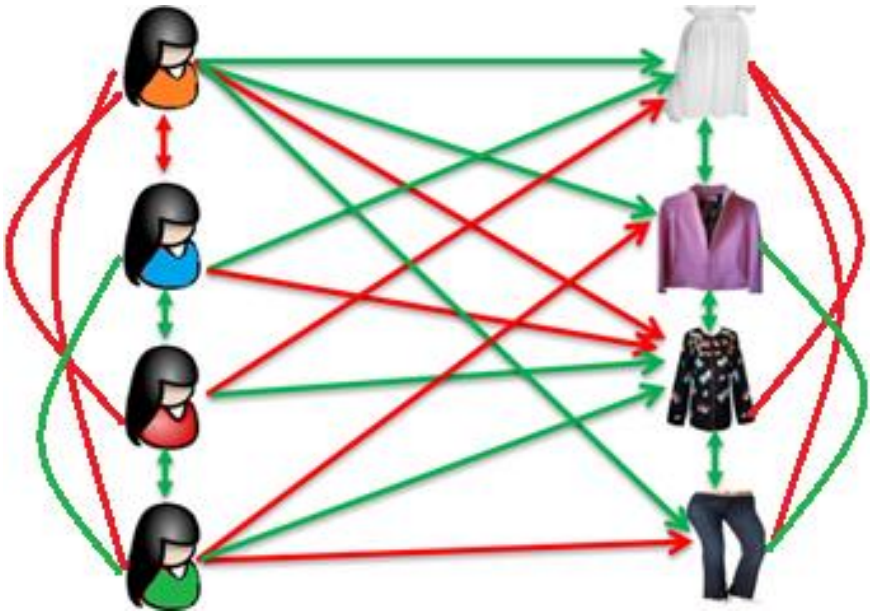


Figure 6.13. *The generated user-item signed graph*

6.4. Experimental Evaluation

The proposed Q-SIMLP algorithm and other comparison methods are implemented on two real-world datasets: MovieLens [41] and MovieLens Hetrec, [42]. These datasets are introduced in chapter 4. First of all, rating conversion is applied to user-item rating matrix in these datasets, they are converted into two imaginary part i and j of the quaternions. Then, the cosine similarity measure is applied to user-item rating matrices of these datasets. Lastly, the user-user and item-item similarity matrices of user-item rating matrices for these datasets are obtained, after the cosine similarity values are passed

through a threshold at 0.5 and 0.7 for Movielens and Hetrec datasets respectively. Likewise, the user-user and item-item dissimilarity matrices of user-item rating matrices for these datasets are obtained after the dissimilarity values are passed through a threshold at 0.5 and 0.3 for Movielens and Hetrec dataset respectively. Also, the threshold of dissimilarity values is represented as 1-threshold of similarity values for these datasets. The threshold of similarity values for Hetrec dataset is indicated as 0.7, since this dataset is sparser than Movielens dataset. Following the combination of all these matrices, the main quaternion based adjacency matrices are constructed as a square matrix for these two datasets as in Eq. (6.8). Hence, we can apply the hyperbolic sine function on the adjacency matrix as a link prediction function, as in [29, 52]. Moreover, we multiply the link prediction function with a parameter α, since the predictions applied to $\boldsymbol{A}$ can be represented as:

$$\alpha \cdot sinh(\boldsymbol{A}) = sinh(\alpha \cdot \boldsymbol{A}) = \boldsymbol{U} \cdot (\alpha \boldsymbol{\Lambda}) \cdot \boldsymbol{U}^{\boldsymbol{T}} \tag{6.12}$$

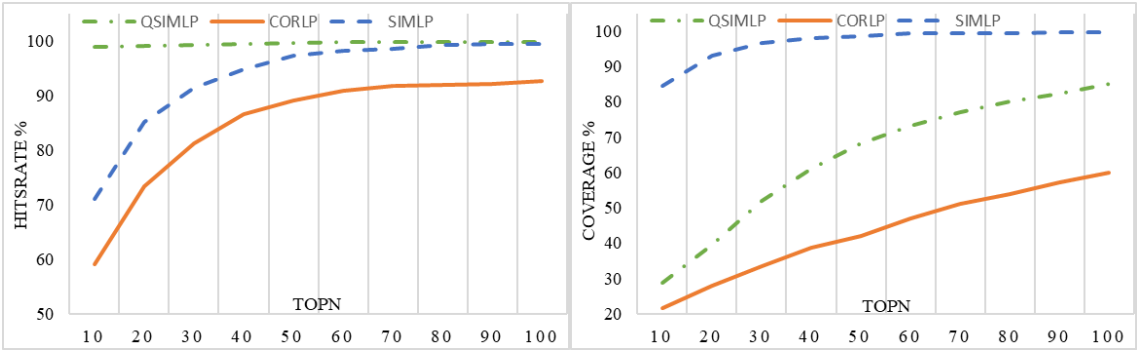
When the adjacency matrix is a square n by n matrix, the sum of the n eigenvalues of $\boldsymbol{A}$ is the same as/equivalent to the trace of $\boldsymbol{A}$;

$$\sum_{\mathbf{i}=1}^{\mathbf{n}} \lambda_{\mathbf{i}} = trace(\boldsymbol{A}) \tag{6.13}$$

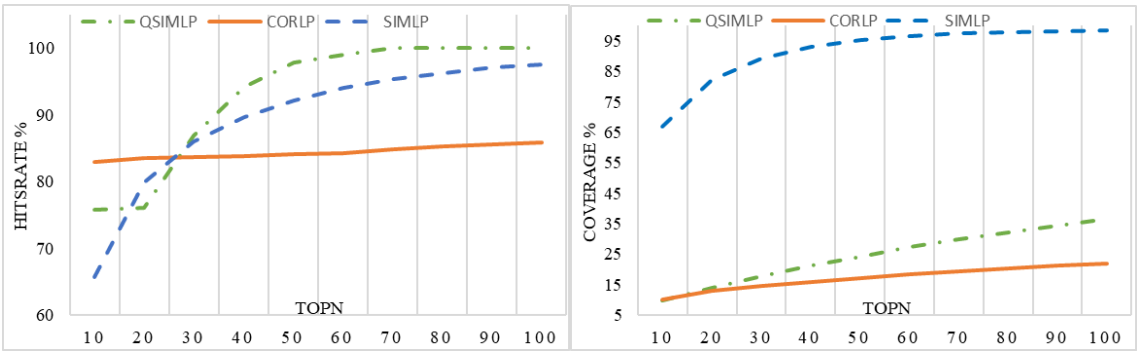
The theory and proof of Eq. (6.13) is given in the Appendix, as theorem 2. Furthermore, we assumed that the trace of the adjacency matrix is equal to the length of the adjacency matrix, since all the components of adjacency matrix values (similar, dissimilar, like and dislike values) are evaluated as binary values before the rating conversion as quaternion numbers. Then, the scaling parameter α is chosen as;

$$\alpha = 1 / length(\boldsymbol{A}), \tag{6.14}$$

since the biggest eigenvalue cannot be bigger than the trace of the adjacency matrix. Hence to normalize the eigenvalues of $\boldsymbol{A}$, we set α as in Eq. (6.14). Moreover, to evaluate the results of CORLP and SIMLP approaches α is set as same as in Q-SIMLP method.



(a)



(b)

Figure 6.14. *Comparison of the Q-SIMLP, SIMLP, and CORLP methods by coverage and hits rate for top-N recommendation on MovieLens (a) and Hetrec (b) datasets*

The testing methodology adopted in the proposed rating based recommendation algorithm is the same as in a previous study [29, 52]. The ratings are split by two subsets that are named by training and test sets, for each dataset as in the previous chapter 4. Also, the rating conversion threshold value is chosen 2.5 for the Hetrec dataset, hence this dataset includes decimal rating numbers. The performance of the proposed Q-SIMLP recommendation method is measured by using the metrics; hits rate and coverage. Figure 6.14 illustrates the comparison of the proposed Q-SIMLP, SIMLP, and CORLP recommendation algorithm with path length 3 for topN recommendation on Movielens (a) and Hetrec (b) dataset. Figure 6.14 shows that the hits rate of the Q-SIMLP method is higher than SIMLP and CORLP method. However, the coverage of the Q-SIMLP method is relatively less than SIMLP method, and still more than the CORLP method on these two datasets. It can be seen in Figure 6.14 that the Q-SIMLP method can give higher results for the top-10 recommendation task. Since the hits rate of the Q-SIMLP method for the top-10 recommendation task is higher than the hits rates of the SIMLP and CORLP methods for the top-100 recommendation. Hence the quaternion-based recommendation algorithm can reach accurate recommendations easily and faster than other approaches.

It is concluded that the Q-SIMLP method provides accurate recommendations by consuming less time.

In this chapter, the signifance of the proposed Q-SIMLP approach and the comparison method CORLP and SIMLP are investigated. Similarly, the signifance is determined with the answer of this question: Does the proposed Q-SIMLP approach that utilizes cosine similarities perform better than CORLP and SIMLP approach for top-N recommendation task? The hits rate and coverage are utilized as the evaluation metrics to measure the performance of the proposed Q-SIMLP recommendation algorithm. One-way Anova test is conducted to further evaluate performance differences between Q-SIMLP and SIMLP and also CORLP approaches respectively. Thus, the specific hypotheses examined in this dissertation are:

• H1: The SIMLP based recommendation approach achieves higher hits rate than the CORLP based recommendation approach does.

• H2: The SIMLP based recommendation approach achieves higher coverage than the CORLP based recommendation approach does.

Table 6.1. *The p-values of the comparison of the Q-SIMLP between CORLP and SIMLP methods with regards to hits rate on MovieLens and Hetrec datasets.*

Methods / Dataset	CORLP	SIMLP
Movielens	0.0005	0.0479
Hetrec	0.0137	0.0538

Table 6.2. *The p-values of the comparison of the Q-SIMLP between CORLP and SIMLP methods with regards to coverage on MovieLens and Hetrec datasets.*

Methods / Dataset	CORLP	SIMLP
Movielens	0.0087	0.00006
Hetrec	0.0249	5×10^{-11}

Table 6.1. includes the only one p-value that reflects no significant differences among Q-SIMLP and SIMLP methods with respect to hits-rate on Hetrec dataset. Since this p-value is so closed to 0.05 (0.0538≅0.05). Then, it can be concluded that there are statistically

significant differences between CORLP, SIMLP and Q-SIMLP methods with respect to hits rate for the experiments on Movielens and Hetrec datasets. Table 6.2. demonstrates that there are statistically significant differences between CORLP, SIMLP and Q-SIMLP methods with respect to coverage for the experiments on Movielens and Hetrec datasets, hence all the p-values are smaller than 0.05. The hypotheses H1 and H2 are supported for each evaluation metrics utilized in this chapter.

The proposed quaternion-based hybrid recommendation method is implemented on three real-world Amazon datasets [45]: Cell phone, Beauty and Clothing. These datasets are introduced in chapter 5. As the same process in Q-SIMLP algorithm, quaternion based rating conversion is applied to user-item rating matrices in these datasets. Then, the cosine similarity measure is applied to user visual-feature and item visual-feature matrices of these datasets. Thus, the user-user and item-item similarity matrices of user-item rating matrices for these datasets are obtained, after the cosine similarity values are passed through a threshold at 0.6, 0.7 and 0.6 for Cell phone, Beauty and Clothing datasets respectively. Similarly, the user-user and item-item dissimilarity matrices of user-item rating matrices for these datasets are obtained after the dissimilarity values are passed through a threshold at 0.4, 0.3 and 0.4 for Cell phone, Beauty and Clothing datasets respectively. At the same time, the threshold of dissimilarity values is determined as 1-threshold of similarity values for these datasets. The threshold of similarity values for Beauty dataset is the highest one, since this dataset is sparser than other datasets.

Following the combination of all these matrices, the main quaternion based adjacency matrices are generated as a square matrix for these three datasets as in Eq. (6.11). Since, the hyperbolic sine function can be applied to the adjacency matrix as a link prediction function, as in chapter 4. Then, we multiply the hyperbolic sine function by a scaling parameter α, as introduced in Q-SIMLP algorithm.

The testing methodology adopted in the quaternion-based hybrid recommendation algorithm slightly alternates from the other proposed hybrid-SIMLP recommendation method, that is introduced in chapter 4. We adopt three product categories of different sizes and density levels, and we generate the standard 10-core datasets from each 5-core datasets for the experiments. Density level of a dataset is calculated as in [22];

$$sparsity = \frac{\#\,zero\ elements}{\#\,total\ elements}, \qquad (6.15)$$
$$density = 1 - sparsity$$

in which $\#\,zero\ elements$ is denoted as the number of zero values in the user-item rating matrix of a dataset and the total number of elements in this matrix is denoted as $\#\,total\ elements$.

Firstly, the user-item rating matrix of 5-core data is filtered out for each user that have at least 10 ratings, to generate a temporary 10-core dataset. Secondly, the temporary 10-core dataset is further filtered out for each items, that have at least 5 ratings. The remaining items in the temporary 10-core data, which have not 5 ratings, are deleted/emitted from the temporary dataset set for the generation of the final 10-core dataset. Since, the 10-core data is a subset of the 5-core data [22], in which all users have at least 10 ratings and items have at least 5 ratings. Statistics of the 10-core datasets are shown in Table 6.3.

Table 6.3. *Statistics of the 10-core datasets.*

Datasets	#Users	#Items	#Interactions	Density
Clothing	5197	4248	37515	0.3%
Cell Phones	3214	2743	34083	0.39%
Beauty	5123	4774	74497	0.17%

Table 6.4. *The performance comparison of Q-Hybrid and Hybrid-SIMLP methods for top-10 recommendation.*

Datasets	Beauty			Clothing			Cell Phones		
Measures (%)	Recall	Hit Ratio	Precision	Recall	Hit Ratio	Precision	Recall	Hit Ratio	Precision
Hybrid-SIMLP	29,21	61,73	2,92	22,25	42,30	2,23	29,75	57,59	2,98
Q-Hybrid	**32,15**	**75,80**	**3,22**	**36,84**	**65,18**	**3,68**	**30,78**	**66,58**	**3,08**

The ratings are split by two subsets that are named by training and test sets, for each dataset. The test set includes only 5-star ratings and only items that are relevant to the corresponding users. The detailed procedure applied to generate the training set and the

test set is the same procedure as mentioned in chapter 5. Also, the performance of the quaternion-based hybrid recommendation algorithm is measured by using the metrics, hit ratio, precision, and recall same as in chapter 5. The results of the proposed quaternion based hybrid recommendation (Q-Hybrid) method utilizing with top-10 recommendation tasks are demonstrated in Table 6.4. The results indicate that the Q-Hybrid recommendation algorithm is obtained higher hit-ratio than other proposed hybrid-SIMLP recommendation algorithm on Beauty, Cell Phone and, Clothing datasets. Likewise, the precision and recall results of the Q-Hybrid method are higher than the results of hybrid-SIMLP recommendation method. It is concluded that quaternion-based representations yields improvements for the performance of hybrid recommendation algorithms.

Moreover, the signifance of the proposed Q-Hybrid approach and the comparison method Hybrid-SIMLP are investigated. Likewise, the answer of the question is investigated to determine signifance: Does the proposed Q-Hybrid approach perform better than Hybrid-SIMLP approach for top-N recommendation task? Furthermore, the range of N is taken from 10 to 100 for experiments on Cellphone, Clothing and Beauty datasets. The hit-ratio, recall and precision are utilized as the evaluation metrics to measure the performance of the proposed Q-Hybrid and Hybrid-SIMLP recommendation algorithm. After that, two-factor Anova test is employed to evaluate performance differences among Q-Hybrid and Hybrid-SIMLP recommendation approaches [66]. Thus, the specific hypotheses analyzed in this chapter are:

- H1: The Q-Hybrid recommendation approach achieves higher hit-ratio than the Hybrid-SIMLP based recommendation approach does.

- H2: The Q-Hybrid recommendation approach achieves higher recall than the Hybrid-SIMLP based recommendation approach does.

- H3: The Q-Hybrid recommendation approach achieves higher precision than the Hybrid-SIMLP based recommendation approach does.

Table 6.5. demonstrates that there are statistically significant differences among Q-Hybrid and Hybrid-SIMLP methods with respect to hit-ratio for the experiments on Cellphone, Beauty and Clothing datasets. Since, the hypothesis H1 are supported by the

experimental results on each dataset defined in this chapter. It can be seen in Table 6.5. that there are statistically significant differences among Q-Hybrid and Hybrid-SIMLP methods with respect to recall for the experiments on Beauty and Clothing datasets, not for the experiments on Cellphone dataset. Moreover, the hypothesis H2 are supported only for the experiments on Beauty and Clothing datasets. Besides that, it can be concluded that there are no statistically significant differences among Q-Hybrid and Hybrid-SIMLP methods with respect to precision for the experiments on each datasets. Finally, the hypothesis H3 are not supported with the experimental results on each datasets.

Table 6.5. *The p-values of the comparison of among Q-SIMLP and Hybrid-SIMLP methods with regards to hit-ratio, recall and precision on CellPhone, Beauty and Clothing datasets.*

Methods / Dataset	Hit-Ratio	Recall	Precision
CellPhone	0.0002	0.5543	0.8507
Beauty	0.00001	0.0160	0.9126
Clothing	0.0008	0.0011	0.0588

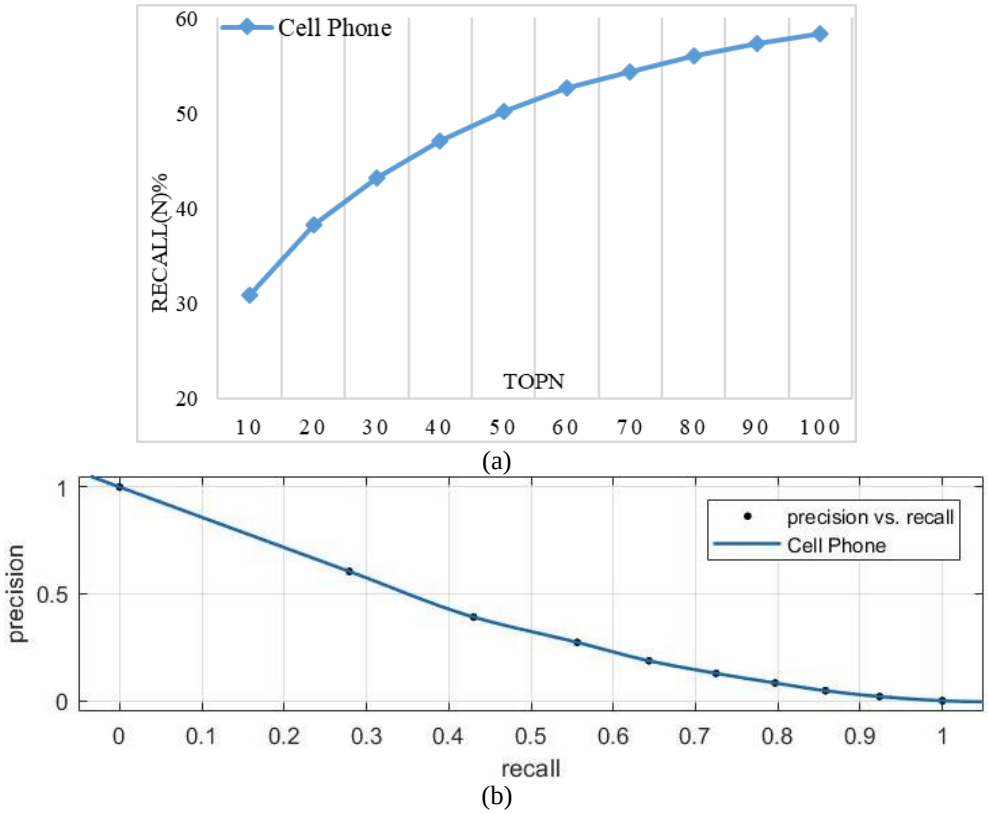
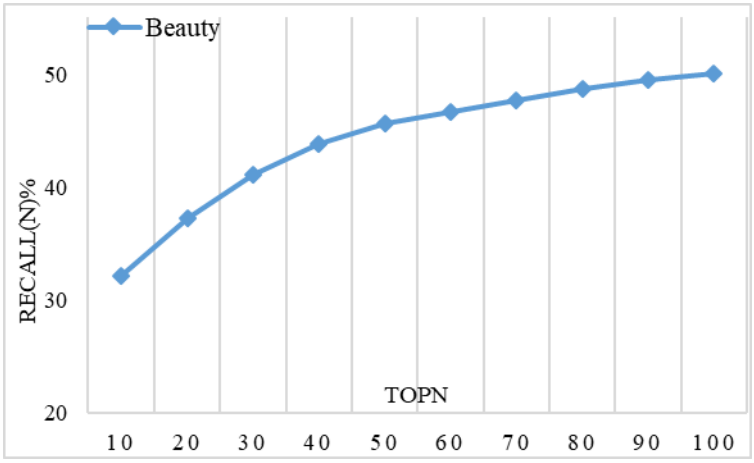
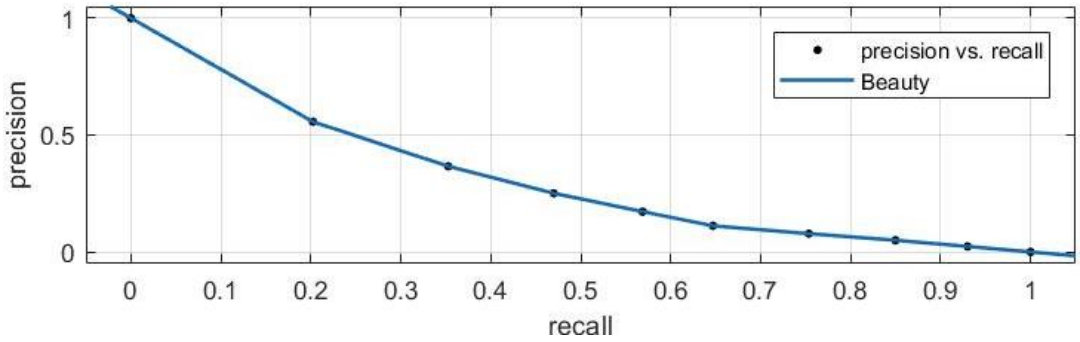


Figure 6.15. *Cellphone: (a) recall-at-N and (b) precision-versus-recall on all items*

The recall(N) and precision(N) results of the proposed Q-Hybrid recommendation algorithm on Cell Phone, Beauty, and Clothing datasets are obtained and drawn respectively in Figure 6.15 (a), Figure 6.16 (a) and Figure 6.17 (a). Furthermore, precision-versus-recall comparison of the results for each dataset are drawn in Figure 6.15 (b), Figure 6.16 (b) and Figure 6.17 (b). It can be seen from these figures that the precision and recall results of the Q-Hybrid method are getting higher than the results of the hybrid-SIMLP method with increasing N for top-N recommendation task.



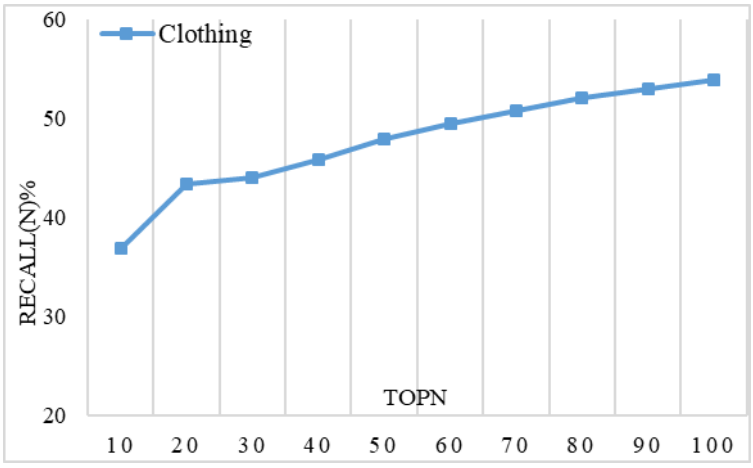
(a)



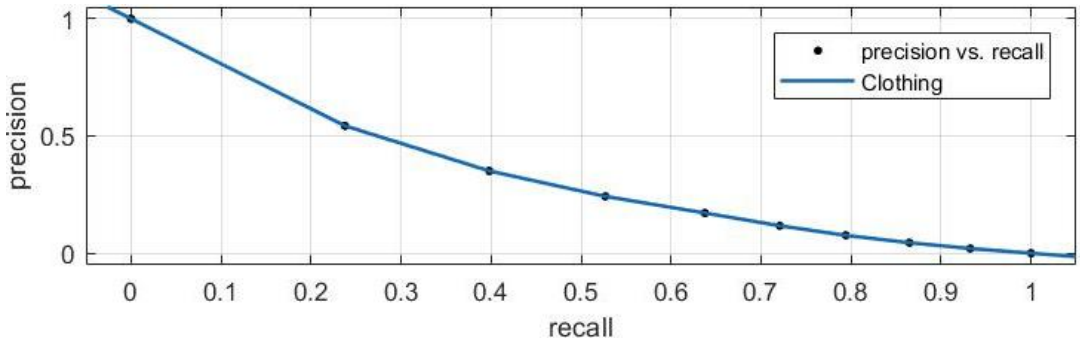
(b)

Figure 6.16. *Beauty: (a) recall-at-N and (b) precision-versus-recall on all items*

Quaternion toolbox [60] in Matlab, that is named by 'qtfm_2.6', is used for the experiments.This toolbox is utilized to generate of quaternion-based adjacency matrix and to evaluate hyperbolic sine of this matrix.



(a)



(b)

Figure 6.17. *Clothing: (a) recall-at-N and (b) precision-versus-recall on all items.*

6.5. Conclusion

Quaternion-based recommendation algorithms are promising methods to overcome the sparsity problem of recommender systems. The proposed quaternion-based recommendation method, Q-SIMLP, is based on such a link prediction approach with the weights in the graph represented by quaternion numbers that may properly distinguish "*similarity*" and "*dissimilarity*" between two users (or two items) nodes and distinguish the "*like*" and "*dislike*" from a user to an item nodes. The experimental results demonstrate that Q-SIMLP method performs better than remaining complex number-based algorithms, such as CORLP and SIMLP, regarding coverage and hits rate on the MovieLens Hetrec and MovieLens datasets. Obtained improvements of Q-SIMLP are attributed to the inclusion of similarity and dissimilarity factors among users and items, and like and dislike relationships between users and items. The results indicate that the proposed Q-SIMLP algorithm is significantly better than CORLP and SIMLP methods in graph-based recommendation systems.

The quaternion based hybrid recommendation method performs better than the proposed Hybrid-SIMLP algorithm in chapter 5, regarding hits ratio, recall, and precision on the real-world Amazon sub-datasets. The improvements of our proposed method are attributed to the inclusion of similarity and dissimilarity factors among users' feature and items' feature vectors. The experimental results show that our approach is revealed the superior performance on real-world datasets, rather than other algorithms. On the other hand, the proposed algorithm is flexible/adaptable to incorporate different information sources. As a conclusion, the proposed algorithm deals well with the deficiencies in hybrid recommender systems, which suggests that the proposed recommender system offers a better algorithmic option/design.

7. CONCLUSIONS

Recommender systems are promising technologies to cope with the information overload problem of modern times. Due to the challenging problem of predicting inclinations of individuals based on their limited past preference history, researchers are implementing new strategies to estimate original items to recommend. Graph-based recommender systems are one such approach to model relations among users and items in a graph structure and estimate referrals using link prediction algorithms. The proposed recommendation method, SIMLP, is based on such a link prediction approach with the weights in the graph represented by complex numbers that can accurately differentiate *"similarity"* between two users (or two items) nodes and distinguish the *"like"* from a user to an item nodes.

The experimental results indicate that the proposed SIMLP method performs better than previous complex number-based algorithms, such as CORLP, regarding coverage and hits rate on the MovieLens Hetrec and MovieLens datasets. The improvements of SIMLP algorithm are attributed to the inclusion of similarity factors that made the algorithm more remarkable. The results demonstrate that the hits rate of the SIMLP method is significantly (about 7%) better than CORLP methods, whereas the coverage is marginally higher compared to this approach. With the modification of the link prediction function by a scaling parameter, the proposed SIMLP method achieves higher hits rates. The proposed method provides relatively higher coverage at smaller scale parameters, meaning that the cold-start problem of recommender systems can be easily overcome.

Furthermore, SIMLP method that modified/utilized by six different similarity measurement is compared in terms of hits rate and coverage while providing top-N recommendations. The modified SIMLP method experimentally scrutinizes how such similarity measures perform with top-N item recommendation processes over the standard MovieLens Hetrec and MovieLens datasets. The experimental results show that hits rate and coverage can be improved by about 7% and 4%, respectively, with Jaccard and Adjusted-Cosine similarity measures being the best performing similarity measures. Significant differences/improvements are observed over the previous CORLP approach. Besides, the SIMLP and CORLP method is modified by the link prediction function as Neumann kernel, and these modified algorithms are examined. About %12 better hits rate

and about %40 better coverage are obtained with SIMLP as opposed to CORLP, both in MovieLens and Hetrec datasets. From two-factor anova test analysis, the proposed SIMLP algorithm is observed to be significantly superior to CORLP.

Moreover, the proposed hybrid recommendation method performs better than other hybrid recommendation algorithms, such as JRL and CKFG regarding with three quality metrics: hits ratio, recall, and precision on the three subsets of Amazon datasets. Obtained improvements of our proposed hybrid recommendation method are attributed to the inclusion of similarity factors among users' visual-feature and items' visual-feature vectors. The experimental results show that the proposed Hybrid-SIMLP method provides better efficiency for the top-N recommendation task, as well as this method is flexible to integrate various information sources. Finally, it is concluded that the proposed Hybrid-SIMLP method deals well with the deficiencies in hybrid recommender systems, rendering the proposed recommender system a better algorithmic option.

The quaternion based recommendation methods are the expansion of other proposed recommendation algorithms in chapter 4 and 5. The results indicate that the hits rate of the Q-SIMLP method is significantly better than SIMLP and CORLP methods, whereas the coverage is almost the same as the results of other methods on Movielens and Hetrec dataset. From one way anova test analysis, the proposed Q-SIMLP algorithm is observed to be significantly superior to CORLP and SIMLP algorithms. Furthermore, the experimental results show that the performance of the proposed Q-Hybrid recommendation algorithm is better than other proposed Hybrid-SIMLP algorithm with utilizing three evaluation metrics: recall, hit-ratio, and precision on the Beauty, Clothing, and Cell Phone datasets. It is concluded from two-factor anova test analysis that, the proposed Q-Hybrid algorithm is observed to be significantly superior regarding with hit-ratio than Hybrid-SIMLP algorithm. The proposed Q-SIMLP and Q-Hybrid recommendation methods improve the overall performance of the system regarding the proposed recommendation algorithms SIMLP in chapter 4 and Hybrid-SIMLP in chapter 5 respectively. Finally, it is concluded that the proposed quaternion based recommendation methods deal well with the deficiencies in graph-based recommender systems

Recommender systems has commonly focused on accuracy, however diversity of recommendations is important to improve the quality of recommendation lists. It has been observed that such as whether the list of recommendations is diverse and whether it contains novel items, provide improvements of the overall quality of a recommender system [70]. As a future work, the diversity of the proposed recommender systems in this dissertation will be investigated. Additionally, the design of a graph-based recommender system, which is utilized by deep neural networks, is considered to be a follow-up of the proposed algorithms in this dissertation.

REFERENCES

[1] Belkin, N. J., and Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin. In *Communications of the ACM*., 12, 29-38.

[2] Webb, G.I. (2011). *Posterior Probability*. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston: MA.

[3] Pazzani, M. J., and Billsus, D. (2007). Content-Based Recommendation Systems. *The Adaptive Web*, 4321, 325–341.

[4] Adomavicius, G., and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6), 734-749.

[5] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. and Riedl, J. (1994). GroupLens: an open architecture for collaborative filtering of netnews. *Proceedings of Conference on Computer supported cooperative work*, pp. 175-186.

[6] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. (2007). Collaborative filtering recommender systems. *The adaptive web*, pp. 291-324. Springer, Berlin, Heidelberg.

[7] Breese, J. S., Heckerman, D., & Kadie, C. (1998, July). Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the Fourteenth Conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc. Madison, Wisconsin, USA, pp. 43-52.

[8] Al-Shamri, M. Y. H., Bharadwaj, K. K., (2008). Fuzzy-genetic approach to recommender systems based on a novel hybrid user model, *Expert Systems with Applications*, 35 (3), 1386-1399.

[9] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T., (2004). Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems*, 22 (1), 5-53.

[10] Huang, Z., Zeng, D., and Chen, H. (2007). A comparative study of recommendation algorithms in e-commerce applications. *IEEE Intelligent Systems*, *22*(5), 68-78.

[11] Zhou, T., Ren, J., Medo, M., and Zhang, Y. C. (2007). Bipartite network projection and personal recommendation. *Physical Review E*, *76*(4), 046115.

[12] Li, X. And Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, *54*(2), 880-890.

[13] Melville, P., Mooney, R. J. and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. *In: Proceedings of 18th National Conference on Artificial Intelligence, Edmonton*, Alberta, Canada, pp. 187-192.

[14] Burke, R., (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12 (4), 331-370.

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

[16] Deshpande, M. and Karypis, G. (2004). Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1), 143-177.

[17] Qiming Diao, Wu, C. Y., Diao, Q., Qiu, M., Jiang, J., and Wang, C. Jointly modeling aspects, ratings and sentiments for movie recommendation. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD* , 14, pp. 193-202.

[18] Sarwar, B.M., Karypis, G., Konstan, J.A., and J. Reidl. (2001) Item-based collaborative filtering recommendation algorithms. *In Proceedings of the 10th International World Wide Web Conference,* pp. 285-295.

[19] James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*. New York: Springer.

[20] Ekstrand, M. D., Riedl, J. T. and Konstan, J. A. (2011). Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81-173.

[21] Gunawardana, A. and Meek, C. (2009). A unified approach to building hybrid recommender systems. *RecSys*, *9*, 117-124.

[22] Zhang, Y., Ai, Q., Chen, X. and Croft, W. B. (2017). Joint representation learning for top-n recommendation with heterogeneous information sources. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management,* ACM., pp. 1449-1458.

[23] Zhang, Y., Ai, Q., Chen, X., and Wang, P. (2018). Learning over knowledge-base embeddings for recommendation. *arXiv preprint arXiv:1803.06540*.

[24] He, R., and McAuley, J. (2016). VBPR: visual bayesian personalized ranking from implicit feedback. *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, July 09-15, 2016, New York, USA, pp.3740-3746

[25] McAuley, J., Targett, C., Shi, Q., and Van Den Hengel, A. (2015, August). Image-based recommendations on styles and substitutes. *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM., pp. 43-52.

[26] Getoor, L., and Diehl, C. P. (2005). Link mining: a survey. *Acm Sigkdd Explorations Newsletter*, *7*(2), 3-12.

[27] Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology*, *58*(7), 1019-1031.

[28] Kunegis, J., De Luca, E. W. and Albayrak, S. (2010). The link prediction problem in bipartite networks. *Proceedings International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*, Springer, Berlin, Heidelberg, pp. 380-389.

[29] Xie, F., Chen, Z., Shang, J., Feng, X. and Li, J. (2015). A link prediction approach for item recommendation with complex number. *Knowledge-Based Systems*, *81*, 148-158.

[30] Kurt, Z. and Özkan, K. (2017). An image-based recommender system based on feature extraction techniques. *Proceedings International Conference on Computer Science and Engineering*, IEEE., pp. 769-774.

[31] Saff, E. B. and Snider, A. D. (2015). *Fundamentals of Matrix Analysis with Applications*. John Wiley & Sons.

[32] Huang, Z., Chung, W., and Chen, H. (2004). A graph model for E-commerce recommender systems. *J. Am. Soc. Inf. Sci. Technol*, *55*(3), 259-274.

[33] Huang, Z., Zeng, D. D. and Chen, H. (2005). A unified recommendation framework based on probabilistic relational models, *Proceedings in 14th Annual Workshop on Information Technologies and Systems*, 8-13.

[34] Pujari, M. (2015). *Link Prediction in Large-scale Complex Networks (Application to bibliographical Networks).* Doctoral dissertation, Université Sorbonne Paris Cité.

[35] Stan, J., Muhlenbach, F. and Largeron, C. (2014). Recommender systems using social network analysis: Challenges and future trends. *Encyclopedia of Social Network Analysis and Mining*, 1522-1532.

[36] Ermiş, B. (2010). *Probabilistic Tensor Factorization For Link Prediction*, Master dissertation, Ankara: Bilkent University, Computer Engineering.

[37] Kunegis, J., Gröner, G. and Gottron, T. (2012). Online dating recommender systems: The split-complex number approach. *Proceedings of the 4th ACM RecSys workshop on Recommender systems and the social web*, ACM., pp. 37-44.

[38] Zhang, Y. and Chen, X. (2018). Explainable recommendation: A survey and new perspectives. *arXiv preprint arXiv:1804.11192*.

[39] Bedi, P., Gautam, A., Bansal, S. and Bhatia, D. (2017). Weighted Bipartite Graph Model for Recommender System Using Entropy Based Similarity Measure. *The International Symposium on Intelligent Systems Technologies and Applications*, Springer, Cham, pp. 163-173.

[40] Cacheda, F., Carneiro, V., Fernández, D. and Formoso, V. (2011). Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. *ACM Transactions on the Web*, *5*(1), 2, 1-33.

[41] http://grouplens.org/datasets/movielens/100k/.

[42] http://ir.ii.uam.es/hetrec2011/datasets.html.

[43] Shimbo, M. and Ito, T. (2006). Kernels as link analysis measures. *Mining Graph Data*, John Wiley & Sons, ch. 12., pp. 283-310.

[44] He, R., and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Proceedings of the 25th international conference on world wide web*, pp. 507-517.

[45] http://jmcauley.ucsd.edu/data/amazon/.

[46] Mishchenko, A. and Solovyov, Y. (2000). *Quaternions*. Quantum 11, 4-7 and 18.

[47] Arvind, D. K. and Valtazanos, A. (2009). Speckled tango dancers: Real-time motion capture of two-body interactions using on-body wireless sensor networks. *Proceedings of the Sixth International Workshop on Wearable and Implantable Body Sensor Networks*, IEEE. pp. 312-317.

[48] Greenblatt, A. B. and Agaian, S. S. (2018). Introducing quaternion multi-valued neural networks with numerical examples. *Information Sciences*, *423*, 326-342.

[49] Saoud, L. S., Ghorbani, R., and Rahmoune, F. (2017). Cognitive quaternion valued neural network and some applications. *Neurocomputing*, *221*, 85-93.

[50] Yuan, X. and Huang, J. J. (2012). An adaptive method for the tag-rating-based recommender system. *Proceeding International Conference on Active Media Technology*, Springer, Berlin, Heidelberg, pp. 206-214.

[51] Du, Y., Xu, C. and Tao, D. (2017). Privileged matrix factorization for collaborative filtering. *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, Melbourne, Australia — August 19 - 25, 2017, pp 1610-1616.

[52] Kurt, Z., Özkan, K., Bilge, A. and Gerek, Ö. N. (2019) A Similarity-Inclusive Link Prediction Based Recommender System Approach. *will be published in Elektronika IR Elektrotechnika*, vol. Xx, no. X, 2019.

[53] Shams, B. and Haratizadeh, S. (2017). Graph-based collaborative ranking. *Expert Systems with Applications, 67*, 59-70.

[54] Huang, Z., Li, X. and Chen, H. (2005). Link Prediction Approach to Collaborative Filtering. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pp. 141–142.

[55] Zhang, Z.-K., Zhou, T., and Zhang, Y.-C. (2010). Personalized recommendation via integrated diffusion on user–item–tag tripartite graphs. *Physica A: Statistical Mechanics and Its Applications*, 389(1), 179–186.

[56] Ting, Y., Yan, C. and Xiang-wei, M. (2013). Personalized Recommendation System Based on Web Log Mining and Weighted Bipartite Graph. *Proceedings International Conference on Computational and Information Sciences*, pp. 587–590.

[57] https://www.dropbox.com/s/th672ttebwxhsfx/CIKM2017.zip?dl=0.

[58] Harary, F. (1955). On the notion of balance of a signed graph, *Michigan Mathematical Journal*, 2, 143–146.

[59] Harary, F. and Palmer, E.M. (1967). On the number of balanced signed graphs. *Bulletin of Mathematical Biophysics*, 29(4), 759-765.

[60] Quaternion toolbox for Matlab, http://qtfm.sourceforge.net/.

[61] Pazzani, M. and D. Billsus. (1997). Learning and revising user profiles: the identification of interesting Web sites. *Machine Learning*, 27(3), 313-331.

[62] Basu, C., H. Hirsh and W. Cohen. (1998). Recommendation as Classification: Using Social and Content-based Information in Recommendation. *Proceedings of 15th National Conference on Artificial Intelligence*, pp. 714-720.

[63] Ansari, A., S. Essegaier and R. Kohli. (2000). Internet Recommendations Systems. *Journal of Marketing Research*, 363-375.

[64] Lieberman, H. (1995). Letizia: An agent that assists Web browsing, *Proceedings of International Joint Conference on Artificial Intelligence*, Montreal, pp. 924-929.

[65] Goldberg, D., D. Nichols, B. Oki and D. Terry. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12), 61-70.

[66] Huang, Z., Chung, W., Ong, T. H. and Chen, H. (2002). A graph-based recommender system for digital library. *Proceedings of the 2nd ACM/IEEE-CS joint Conf. on Digital libraries*, ACM, Oregon, USA, pp. 65-73.

[67] Sarwar, B., J. Konstan, A. Borchers, J. Herlocker, B. Miller and J. Riedl. (1998). Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system, *Proceedings of ACM Conference on Computer Supported Cooperative Work*, pp. 345-354.

[68] Aggarwal, C. C., Wolf, J. L., Wu, K. L., Yu, P. S. (1999). Horting hatches an egg: A new graph-theoretic approach to collaborative filtering. *Proceedings of the 50th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM. pp. 201-212.

[69] Schein, A. I., A. Popescul, L. H. Unger and D. M. Pennock. (2002). Methods and metrics for ColdStart recommendations. *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* ACM, Tampere, Finland, pp. 253-260.

[70] Kaminskas, M., Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems*, 7(1), 2.

[71] Demir, S., Tanışlı, M. (2007). Complex quaternionic reformulation of the relativistic elastic collision problem. *Physica Scripta*, *75*(5), 630.

[72] Tanişli, M., Kansu, M. E., Demir, S. (2013). Supersymmetric quantum mechanics and euclidean–dirac operator with complexified quaternions. *Modern Physics Letters A*, 28(08), 1350026.

[73] De SaÂ, H. R., Prudêncio R. B. (2011). Supervised link prediction in weighted networks. *Proceedings of the IEEE International Joint Conference on Neural Networks*., IEEE, USA pp. 2281-2288.

[74] Huang, Z. (2005). Graph-based analysis for e-commerce recommendation. Phd dissertation, Arizona: The University of Arizona, Faculty of the Business Administration.

APPENDIX

Definition 1. The polynomial $f_{\boldsymbol{A}}(\lambda)=det(\boldsymbol{A}-\lambda \boldsymbol{I}_n)$ is called the characteristic polynomial of $\boldsymbol{A}$, [31].

Proposition 1. The eigenvalues of $\boldsymbol{A}$ are the roots of the characteristic polynomial.

Proof. If $\boldsymbol{A}v=\lambda v$, then v is in the kernel of $\boldsymbol{A}-\lambda \boldsymbol{I}_n$. Moreover, we know that $det(\boldsymbol{A})\neq 0$ if and only if $\boldsymbol{A}$ is invertible. Consequently, $\boldsymbol{A}-\lambda \boldsymbol{I}_n$ is not invertible, and then $det(\boldsymbol{A}-\lambda \boldsymbol{I}_n)=0$. Since it gives that the eigenvalues λ are the roots of the characteristic polynomial $f_{\boldsymbol{A}}(\lambda)=det(\boldsymbol{A}-\lambda \boldsymbol{I}_n)$.

Theorem 1. Let $\boldsymbol{A}$ be a square n by n matrix and let $\lambda=(\lambda_1,\lambda_2,...,\lambda_n)$ be all the eigenvalues of $\boldsymbol{A}$. For each positive integer k, prove that $\lambda^k=(\lambda_1^k,\lambda_2^k,...,\lambda_n^k)$ are all the eigenvalues of $\boldsymbol{A}^k$, [31].

Proof 1. By the triangularization (or Jordan canonical form), there exists a nonsingular matrix $\boldsymbol{S}$ such that;

$$\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S}=\begin{bmatrix}\lambda_1 & * & * & * & * \\ 0 & \lambda_2 & * & * & * \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \lambda_{n-1} & * \\ 0 & 0 & 0 & 0 & \lambda_n\end{bmatrix}.$$

Here the right matrix is an upper triangular matrix whose diagonal entries are eigenvalues of $\boldsymbol{A}$.

Then we take the k power of the Jordan canonical form and have

$$\boldsymbol{S}^{-1}\boldsymbol{A}^k\boldsymbol{S}=(\boldsymbol{S}^{-1}\boldsymbol{A}\boldsymbol{S})^k=\begin{bmatrix}\lambda_1^k & * & * & * & * \\ 0 & \lambda_2^k & * & * & * \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \lambda_{n-1}^k & * \\ 0 & 0 & 0 & 0 & \lambda_n^k\end{bmatrix}.$$

The characteristic polynomial of the matrix $\boldsymbol{A}^k$ is given by;

$$
\begin{aligned}
& f(\lambda) = det(\boldsymbol{A}^k - \lambda \boldsymbol{I}) \\
& = det(\boldsymbol{S}^{-1}) det(\boldsymbol{A}^k - \lambda \boldsymbol{I}) det(\boldsymbol{S}) \\
& = det(\boldsymbol{S}^{-1} \cdot (\boldsymbol{A}^k - \lambda \boldsymbol{I}) \cdot \boldsymbol{S}) \\
& = det(\boldsymbol{S}^{-1} \cdot \boldsymbol{A}^k \cdot \boldsymbol{S} - \lambda \boldsymbol{I}) \\
& = \begin{bmatrix} \lambda_1^k - \lambda & * & * & * & * \\ 0 & \lambda_2^k - \lambda & * & * & * \\ \vdots & \cdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \lambda_{n-1}^k - \lambda & * \\ 0 & 0 & 0 & 0 & \lambda_n^k - \lambda \end{bmatrix} \\
& = \prod_{i=1}^{n} (\lambda_i^k - \lambda)
\end{aligned}
\tag{A.1}
$$

Since the roots of the characteristic polynomial are all the eigenvalues as mentioned in Proposition 1, we can see from the Eq. (A.1) that $\lambda^k = (\lambda_1^k, \lambda_2^k, ..., \lambda_n^k)$ are all the eigenvalues of $\boldsymbol{A}^k$.

Hence, the power k of $\boldsymbol{A}$, when the eigenvalue matrix of $\boldsymbol{A}$ is denoted as $\boldsymbol{\Lambda}$, and the eigenvector matrix of $\boldsymbol{A}$ is denoted as $\boldsymbol{U}$, then $\boldsymbol{A}^k$ can be written as;

$$\boldsymbol{A}^k = \boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^T$$

Theorem 2. If $\boldsymbol{A}$ is a square matrix with eigenvalues $\lambda_i, i = 1,2,...,n$,

$$\text{(a) } \det(\boldsymbol{A}) = \prod_{\mathbf{i}=1}^{\mathbf{n}} \lambda_{\mathbf{i}} \text{, (b) } trace(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i .$$

Proof. [Method 1] Recall that eigenvalues are roots of the characteristic polynomial $f_A(\lambda) = det(\boldsymbol{A} - \lambda \boldsymbol{I}_n)$.

It follows that we have

$$
\begin{aligned}
det(\boldsymbol{A} - \lambda \boldsymbol{I}_n) &= \begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1,n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{m2} & \cdots & a_{nn} - \lambda \end{vmatrix} \\
&= \prod_{i=1}^{n} (\lambda_i - \lambda).
\end{aligned}
\tag{A.2}
$$

If we set $\lambda = 0$, we see that $\det(\boldsymbol{A}) = \prod_{\mathbf{i}=1}^{\mathbf{n}} \lambda_{\mathbf{i}}$ and this completes the proof of part (a).

Compare the coefficients of λ^{n-1} of the both sides of Eq. (A.2). The coefficient of λ^{n-1} of the determinant on the left side of Eq. (A.2). is

$$(-1)^{n-1}(a_{11} + a_{22} + \ldots + a_{nn}) = (-1)^{n-1} trace(A).$$

The coefficient of λ^{n-1} of the determinant on the right side of Eq. (A.2) is

$$(-1)^{n-1} \sum_{i=1}^{n} \lambda_i .$$

Thus we have $trace(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i$.

Proof. [Method 2] Expressing the characteristic polynomial $f(\lambda)$ as a product $\prod_i (\lambda - \lambda_i)$, we get $0 = f(\lambda) = \prod_{i=1}^{n} (\lambda - \lambda_i) = \lambda^n - \lambda^{n-1} \sum_{i=1}^{n} \lambda_i + \ldots + (-1)^n \prod_{i=1}^{n} \lambda_i$.
With multiplying a column by -1 inverts the sign of the determinant. It follows that $det(\boldsymbol{A}) = (-1)^n det(-\boldsymbol{A})$, and since, $f(0) = det(-A) = (-1)^n \prod_i \lambda_i$ we have $\det(\boldsymbol{A}) = \prod_{\mathbf{i}=1}^{\mathbf{n}} \lambda_{\mathbf{i}}$.

Substituting the definition of the determinant in $det(\lambda \boldsymbol{I}_n - \boldsymbol{A})$, we see that the only terms of power λ^{n-1} result from a multiplication of the diagonal terms $\prod_{i=1}^{n} (\lambda - \boldsymbol{A}_{ii})$. More specifically, there are n terms containing a power λ^{n-1} in the determinant expansion: $-\boldsymbol{A}_{11}\lambda^{n-1}, \ldots, -\boldsymbol{A}_{11}\lambda^{n-1}$. Collecting these terms, we get that the coefficient associated with λ^{n-1} in the characteristic polynomial is $-trace(\boldsymbol{A}) = \sum_{i=1}^{n} \boldsymbol{A}_{ii}$. Comparing this to the coefficient of λ^{n-1} in the equation above we get that $trace(\boldsymbol{A}) = \sum_{i=1}^{n} \lambda_i$.