Trust-Based Rating Prediction and Malicious Profile Detection in Online Social Recommender Systems

2018

Anahita Davoudi
University of Central Florida

TRUST-BASED RATING PREDICTION AND MALICIOUS PROFILE DETECTION IN ONLINE SOCIAL RECOMMENDER SYSTEMS

by

ANAHITA DAVOUDI
B.Sc., AmirKabir University of Technology, Iran, 2008
M.Sc., University of Texas Arlington, USA, 2012

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2018

Major Professor: Mainak Chatterjee

ABSTRACT

Online social networks and recommender systems have become an effective channel for influencing millions of users by facilitating exchange and spread of information. This dissertation addresses multiple challenges that are faced by online social recommender systems such as: i) finding the extent of information spread; ii) predicting the rating of a product; and iii) detecting malicious profiles. Most of the research in this area do not capture the social interactions and rely on empirical or statistical approaches without considering the temporal aspects. We capture the temporal spread of information using a probabilistic model and use non-linear differential equations to model the diffusion process. To predict the rating of a product, we propose a social trust model and use the matrix factorization method to estimate user's taste by incorporating user-item rating matrix. The effect of tastes of friends of a user is captured using a trust model which is based on similarities between users and their centralities. Similarity is modeled using Vector Space Similarity and Pearson Correlation Coefficient algorithms, whereas degree, eigen-vector, Katz, and PageRank are used to model centrality. As rating of a product has tremendous influence on its saleability, social recommender systems are vulnerable to profile injection attacks that affect user's opinion towards favorable or unfavorable recommendations for a product. We propose a classification approach for detecting attackers based on attributes that provide the likelihood of a user profile of that of an attacker. To evaluate the performance, we inject push and nuke attacks, and use precision and recall to identify the attackers. All proposed models have been validated using

datasets from Facebook, Epinions, and Digg. Results exhibit that the proposed models are able to better predict the information spread, rating of a product, and identify malicious user profiles with high accuracy and low false positives.

To my mom Roya and my sister Anis

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Mainak Chatterjee for his mentorship over the years of my PhD studies. His tireless support during these years have definitely impacted me on professional as well as personal levels. I would like to extend my sincere appreciation to my committee members Dr. Haiyan Hu, Dr. Nazanin Rahnavard, and Dr. Cliff Zou. Their helpful suggestions have helped improve this work and my PhD experience. I would like to also thank my lab mates and friends whose understanding and friendship made my PhD years so much easier.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The Internet was primarily designed for networking networks of computers. However, over the decades it has been used for much more than that– it has brought people, groups, and societies together through their online presence and interactions. On-line social networks such as Facebook, Google+, LinkedIn, etc, have transformed not only the way we communicate with each other but also how we share information. Today, there are many social networking platforms that are used to share multimedia contents (e.g., Flickr, YouTube, and Google Video) and there are many which are primarily used for news and blogs (e.g., Twitter, LiveJournal, BlogSpot, and Digg).

These online social networks not only allow us to remain connected with our friends and relatives but also facilitate information propagation– be it advertising of a certain product or dissemination of a political agenda. Realizing the potential of these online platforms to reach millions of users, a lot of research has been initiated that try to find the most effective strategies for information diffusion in these kinds of networks. Alongside, the availability of *online social data* has made it possible to not only validate the new models that are being developed but also to allow us to predict the future behavior of users. However, to accurately model how a phenomenon would spread across a network is a challenging problem due to the complexity of social interactions between users.

Furthermore, to exploit the power of social networks and realizing that people have the ability to positively or negatively bias ones' opinions, businesses have started using recommender systems that help customers with item selection and purchasing decisions based on individual's tastes and preferences. Recommender systems help users narrow down the set to choose from; for example, selecting an item (i.e., which movie to watch) based on user's preference or helping with online purchasing decisions based on how other users have rated the product to be bought. Recent studies have shown that social recommendations play a significant role in our daily lives [94, 95, 112, 127]. We tend to value recommendations from people we know and trust rather than getting opinions from recommender systems. It is intuitive that two users with similar tastes are more probable to show similar behavior with regard to product or a news item.

1.1 Open Problems and Challenges

In spite of the advancements made on models that predict information diffusion, there remain challenges that must be overcome to accurately predict how a phenomenon will spread across an online social network given the network structure, connections between users, and the possibility of having malicious users. Next, we discuss some open problems and challenges.

1.1.1 Information Diffusion

Many empirical studies have characterized information diffusion in social networks [67, 121] and multiple mathematical models have been proposed that quantitatively describe the diffusion process [15,41,61,95,131]. The mathematical models extracted from epidemiological processes have influenced social networks' research as well [94]. According to previous studies on methods applied to information diffusion in online social networks, the non-graph based predicative models are of 3 types: i) Epidemiological, ii) Linear Influence Model (LIM), and iii) Partial Differential Equations (PDEs) [45]. The epidemiological models are based on Ordinary Differential Equations (ODE) or probabilistic models [94]. However, they do not necessarily capture the temporal aspects of the information spread i.e., at what rate a piece of information spreads during its lifetime. Though the LIM method [131] predicts the temporal dynamics of the information diffusion by solving the non-negative least squares problems, it does not account for the carrying capacity of the network. Thus, there is a need to develop a model that would capture the feature of dynamic carrying capacity based on the influenced users in the system.

1.1.2 Recommender Systems

Traditional systems assume that users are independent and identically distributed and ignore the varied level of social interactions between users. Thus, the traditional recommender systems fail to capture the importance that we put on our social connections as it bases its recommendations only

on the user-item rating matrix. User's social relationships play an important role in the behavior of users regarding future ratings [30, 119]. Moreover, users' preferences are shaped by their social connections which can be explained by homophily [86]– a phenomenon in which users with similar interests are more likely to be connected.

Social recommender systems focus on easing information and interaction burden by applying different methods that present the most relevant information to the users. However retailing platforms usually do not consider social factors such as relationships and trust among the users and the power of social influence is not exploited. On the other hand, social networking platforms generally do not consider online shopping related factors such as purchase history and product rating. In addition to social connections, trust relationships also influence one's decisions and ought to be considered for recommendations. In a social network, trust relationships and social relationships are two different concepts. Two socially connected users would not necessary trust each other. Also, multiple connections of a user would not have equal impact on user's opinions and decisions. Also social influence [81] suggests that connected users are more likely to have similar interests. Since most of the similarities within a network are caused by the influence and interactions of its users, it is reasonable to develop a social recommender system based on the user connections and interactions. Despite many studies on similar problems, there is still a great potential in exploring the social relationships in furnishing and harnessing the recommender systems.

1.1.3 Anomaly Detection

Typically, recommendations systems base their recommendations on product ratings and reviews that the customers provide. Though such inputs from the users enrich the recommender database, they also make the system vulnerable to numerous types of attacks. Recommender systems are vulnerable to these attacks since their algorithms collect user profiles, which represent the taste of users and make recommendation based on these tastes. One of the popular attack types is the profile-injection attack where malicious users insert fake user profile in order to promote (i.e., push attack) or demote (i.e., nuke attack) a specific product. In a profile injection attack, an attacker would interact with the recommender system to create a number of fake profiles that try to bias the system's output. Though producers of items want their own items to be recommended more often than those of their competitors by injecting fake profiles, they are nevertheless considered malicious or attackers [65]. To counter the above mentioned problems and to make a recommender system robust to attacks, many methods have been proposed that deal with the profile injection attacks [11, 65, 91]. Also many detection methods such as statistical techniques [12, 53], classification [19], unsupervised clustering [17, 100], and Beta-Protection algorithm [28] have been proposed. However, other than simply applying commonly used user-item rating matrix, valuable information can be obtained from social interactions which is represented by the user-connection matrix.

1.2 Contributions of the Dissertation

In order to address the above mentioned challenges, we propose multiple methods to handle these issues. We propose probabilistic and differential equation information diffusion models. We capture the effects of centrality and similarity in user rating prediction. We present a model that analyzes the attributes of social connections in identifying malicious users.

1.2.1 Information Diffusion in Social Networks

The extent of information spread in a social network depends on how users react when a new information is received. We consider two different models for information diffusion: i) probabilistic model, and ii) differential equation model. For both models, we consider that the network is scale-free and obeys power-law degree distribution.

Probabilistic Model

In this model, a node provides recommendations to its neighbors in a probabilistic manner. A node that is the origin for the recommendation starts by recommending a product to its directly connected neighbors. The neighbors in turn, recommend to their neighbors in a probabilistic manner. Obviously, the distance of a node from the origin (i.e., hop-count) plays a crucial role as the recommendation of the product propagates through the network. In order to find what fraction of the nodes get the recommendation, we start by computing the probability with which a node gets the recommendation. To do so, we divide the problem into three components: i) when a node gets

recommendation from nodes that are closer to the origin (i.e., one hop-count less), ii) when the node gets recommendation from nodes further from the origin (i.e., one hop-count more), and iii) from nodes that have the same hop-count. We use the in-degree/out-degree distribution functions and the clustering coefficients to compute these three probabilities. Using a dataset from Facebook available at SNAP [85], we show the impact of how the location of a node from the origin affects the probability of being recommended. Also, we find what happens when the origin node has a certain connectivity and the impact of its distance from the hub.

Differential Equation Model

We use partial differential equations (PDEs) to study the temporal patterns of information diffusion process considering the social carrying capacity to be dynamic. Typically, when a user posts a piece of information like a news story, it draws the attention of followers of that user. If the followers like/vote the story then their followers would be able to get that story. This process might continue or die out depending on the level of interest of the story, user connections, and their interactions. These factors determine the carrying capacity of the network at any point of time. Our model is able to predict the influenced users at any time. The predicted values are found by minimizing the Mean Absolute Error (MAE) between the observed and predicted values. Genetic Algorithm (GA) with random initial guess was used for minimizing the error. In order to validate our proposed model, we use real dataset collected from Digg [52] which is a popular news aggregation website. The dataset consists of millions of votes on news stories during June of 2009. The news aggregation that does not emerge from the structure of social networks behaves mostly randomly which would be similar to random walk in case of partial differential equations.

This feature makes the Digg dataset a good source to analyze the information spreading. When the votes are cast, their timestamps are recorded, which allows us to study and predict the diffusion patterns.

1.2.2 Rating Prediction in Social Networks

Social recommender systems play a significant role in our daily lives since we tend to value recommendations from people we know and trust rather than getting opinions from traditional recommender systems. We investigate how different factors affect affecting user rating behavior. We model the user rating prediction based on connections, trust relationships, centrality in the system, and similarity to other users.

Rating prediction model based on centrality and trust

Based on the above observations, we propose to use the social network in conjunction with the user-item rating matrix to accurately predict the rating of a product. We not only consider the user connections but also consider that one values the opinions of all her connections differently. This is because there is non-uniformity in how we trust our connections. Also, trust is non-transitive and asymmetric implying the extent to which A trusts B does not necessarily mean that B would trust A to the same extent.

We predict how a user would rate a product based on not only what the system recommends, but also on how her connections rated the product. We use the time-varying trust relationships to compute how important each connection is and weigh that with the ratings provided by that con-

nection. We update the predicted rating using an exponentially weighted moving average. Using the trust matrix, we model the importance of a connection using two centrality measures: degree and eigen-vector centralities. As trust changes over time, so does the centrality. For degree centrality, we simply use the user adjacency matrix without caring for how trustworthy a connection is. We update the eigen-vector centrality using the current trust matrix and the centrality from the previous time period. To find the overall rating, we find how the connections and non-connections affect the ratings. We use their linear combination using the social factor [131] as the weight for the ratings by ones' connections. Using the mean absolute error, we measure how accurate our predictive model is.

In order to verify the accuracy of our predictive model, we resort to simulation using data from Epinions [122]. The dataset primarily consists of the trust relationship matrix and the user-item rating matrix for 11 time periods. Our method predicts the rating for products for each user based on the ratings a user receives from her connections and from all the other users who rated the same product. These two types of ratings are combined using the social impact factor, $0 \leq \lambda \leq 1$. We predict the overall rating of a product by a user and compare with the real data set for the prediction accuracy given by the mean absolute error. For modeling the importance of the connections, we use both degree centrality and eigen-vector centrality. The results show that our method outperforms the prediction schemes that do not consider centrality measures. Our method can also be applied to larger datasets since it has a linear complexity.

Rating prediction model based on centrality, similarity, and trust

We combine the features of social networks and e-commerce platforms to design a social recommender mechanism to increase the prediction accuracy of product recommendations in e-commerce by considering the factors of similarity, user importance in the network, and social trust relationships. The proposed model could be practically applied to new emerging social commerce platforms. We argue that users are influenced by social interactions, in particular, by the set of trusted friends and their respective importance. To that end, we combine social trust connections and user-item matrix to predict the rating that a user would assign to a product. We use matrix factorization to factor user-item rating matrix into two low-dimensional matrices consisting of user latent matrix and item latent matrix. For the social connections, we consider both user importance and user similarity to build the social trust model between users. We use vector space similarity (VSS) and Pearson Correlation Coefficient (PCC) to obtain the similarity between users. Using degree, eigen-vector, Katz and PageRank centralities, we quantify the importance of users in the network. We use a linear combination of similarity and centrality to model the trust parameter between users. The proposed method captures the balance between user taste and her friends' taste and adjusts the share of centrality and similarity in the trust values using two parameters. The low-dimensional latent user-specific and item-specific matrices are estimated by performing gradient descent on the objective function. As for the objective function we seek to minimize the sum-of-squared-errors between the predicted and actual rating values. We use a dataset from Epinions [122] to validate the proposed model. We estimate the accuracy of the proposed method in terms of the mean absolute error by comparing the predicted and the actual user ratings of

products. Results reveal that there is a high correlation between the predicted and the actual ratings. The proposed method is also compared using binary trust values as well as considering the eigenvector and degree centralities. Our experiment results show that the proposed model could enhance recommendation accuracy.

1.2.3 Anomaly Detection in User Behavior in Social Networks

In order to detect the suspicious users in online social networks, we take a different approach than simply applying the commonly-used user-item rating matrix. We argue that valuable information can be obtained from social interactions which is represented by the user-connection matrix. Also, we observe that injecting fake user profiles would cause meaningless connections with other users. To that end, we propose three detection attributes: i) deviation from predicted rating, ii) similarity between two users, and iii) abnormal rating behavior. These attributes, based on user-item rating matrix and user-connection matrix, provide the likelihood of a user having a profile of that of an attacker. The output of these three attributes are fed to a k-means clustering algorithm that categorizes users into authentic users and attackers. In order to verify the accuracy of our anomaly detection framework, we used Epinions dataset [122] with 922267 ratings on 296277 products by 22166 users having 355754 connections between them. Based on values obtained by precision and recall parameters, the clusters built using detection attributes can detect fake users with high probability– the exact value of which depends on other system parameters. We also observe that

detection of the attacker profiles is not only based on user behavior and attack types, but also are based on the filler size and attack size for each attack type.

1.3 Organization of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 discusses the previous studies in recommender systems and presents the significant related work that are relevant to this dissertation. Probabilistic Information Diffusion model is presented in chapter 3. Differential Equation Diffusion model is presented chapter 4. In chapter 5, the connection-based rating prediction is presented. In Chapter 6, the centrality, similarity and trust metrics are used to predict the user rating. Chapter 7 presents different detection attributes to identify malicious users in social networks. Conclusions of this dissertation are drawn in chapter 8.

CHAPTER 2: BACKGROUND AND RELATED WORK

Several studies have attempted to model how social networks influence users' daily life. With the availability of large data sets of various social networks, there have been various investigations on what additional information those data sets reveal. In this chapter, we discuss how information diffuses in a network and the role of recommender systems. We also discuss how malicious users or fake profiles are identified in social networks.

2.1 Information Diffusion in Social Networks

The study of information spreading in social networks has recently become increasingly popular among research communities [68]. Empirical methods have been applied to different online social networks which showed information diffusion patterns in these networks. In [44] information diffusion in weblogs has been studied. Multiple studies [23, 133] have analyzed spreading of popular photos in Flickr. In [67], news spreading in Digg and Twitter have been studied based on empirical data. Also, epidemic transmission of popular news and user characteristics of Digg have been empirically studied [121, 124].

Multiple studies on mathematical models for the diffusion process have taken a more global perspective– a survey of which can be found in [5]. Continuous time Markov chain has been used

in [120] to model the information diffusion that is based on interpersonal discussion rate. The Susceptible-Infectious-Susceptible (SIS) epidemic model has been used in [113] to characterize information diffusion in social networks. Also [41, 131] have proposed different mathematical models to capture the information diffusion in social networks. Another study [128] proposed a PDE model for information diffusion validated by Digg dataset.

There are other models that have analyzed the system locally due to the importance of interpersonal interactions in predicting the information diffusion. A model has been proposed to predict the negative/positive impact of a user on her neighbors [68]. Linear Threshold and Independent Cascade Models [61] were used to search the most influential users. Many studies have been done in other areas such as biology, sociology, economics, and physics to model information diffusion [15, 22, 39, 42, 60] which use dynamic mathematical models including ordinary differential equations and partial differential equations. The methods mentioned so far did not consider the dynamic carrying capacity of the system-which is one of the contributions of this dissertation.

Businesses are making use of correlated data from social networks for product recommendations and advertising. Assuming that a user's chances of buying a product would be impacted by the opinion of her trusted friends, a study to maximize the marketing was done using the Epinion trust network in [112]. Efficiency of several algorithms for maximizing the influence in marketing through a social network has been studied in [61]. It has been found that there are different patterns of spreding in the network since a node may receive recommendation from multiple sources which might even be contradicting [69]. Moreover, there could be cascading effects based on the connec-

tivity of the network. Diffusion of information via word-of-mouth and viral marketing effects for new products has been investigated in [14].

2.2 Rating prediction in social networks based on Similarity, Centrality and Trust

The rapid expansion of the online world and e-commerce has led to serious problem of information overload, where the users find it difficult to quickly locate the right product. Users are overloaded with many choices when making on-line purchasing decisions, and recommender systems have become handy and alleviate the problem by providing customized recommendations. These systems offer a personalized experience based on social interactions or user preferences which are considered as fantastic opportunities for retailers in e-commerce businesses.

2.2.1 Recommender Systems

Users have many choices while purchasing products online and recommender systems are becoming more popular as they provide the needed information both for consumers and retailers. Many recommendation techniques have been studied [64, 116] and have been well adapted to commercial websites which offer a vast number of products for users with different tastes. Good examples of such systems are Epinions [122], IMDb, and Amazon where there are sets of products which have been rated by other users. Some systems like Netflix provide recommendations based on

users' taste and preferences. Users can read reviews about a variety of products which aid their purchasing decision. Also users have the option to submit their own reviews.

When someone does not have sufficient information on a product that she wants to buy, she would probably seek advice from friends and family. Such recommendations from our social connections are often instrumental in forming an opinion about a product [62]. Recommender systems are being used to address this need as well [2]. Recommender systems help customers by providing useful information and recommendations on products they are interested in [116]. In recommender systems, a node passes recommendations to its neighbors to spread the information through the network [29]. E-commerce companies selling the products, know this fact and make use of the social networks for advertising and reaching out to a target customer base.

In recent years, different types of recommender systems have been developed, most of which use content-based filtering, collaborative filtering, or a mix of both [8]. Content-based systems use items' characteristics and the ratings that users have given to generate recommendations. Collaborative systems identify similar users and analyze their preferences to generate recommendations. Hybrid methods, such as the content-based collaborative filtering algorithm [72, 89], combine these two techniques, hoping to avoid the limitations of either approach and improve the recommendation performance.

2.2.2 Collaborative Filtering

Collaborative filtering methods have proved to be useful and take advantage of the collaborative world especially when combined with hybrid methods [8]. Collaborative filtering methods are further divided into three categories: memory-based, model-based, and hybrid of both. An example of an algorithm which is a hybrid between memory-based and model-based methods is personality diagnosis [105].

Memory-based Methods: Memory-based methods utilize users' past behavior and recommend products that other users with similar interests have selected in the past [116]. They have been widely used in commercial recommender systems [109]. Memory-based algorithms are either user-based [13,50] or item-based [71,116]. User-based algorithms predict rating given by a user to an item based on the ratings by similar users, whereas, item-based algorithms estimate the rating based on the ratings of similar items previously chosen by the user.

Model-based Methods: Model-based methods utilize available data to train a predefined model for rating prediction. Some of the commonly used methods are: clustering [63] and Matrix Factorization [77]. Model-based approaches can handle problems with limited data using hierarchical clustering to enhance the accuracy of the prediction [63]. Matrix factorization factorizes the user-item rating matrix using low-rank representation.

2.2.3 Factors Governing Rate Prediction

Let us now discuss some related research on trust, similarity, preference, and social influence which we argue are the most important factors that govern the design of efficient recommender systems.

Trust: Since in online environments users do not have enough information about other users or items being offered, online interactions involve taking some risks as doing business with people we never met before requires a great deal of trust [56]. Trust has a significant impact on users' online purchasing behavior. Therefore, trust plays a critical role in e-commerce experience. The importance of a user must be taken into consideration for finding the true rating of a product. Thus, it is crucial to model the importance of a user using a trust parameter so that the ratings by malicious users can be purged. In online communities, it is essential to trust the data we receive. Trust helps users to assign a value to other users based on their willingness to interact with them [7]. Trust between users can be of two types: implicit [96] and explicit [83, 107]. Implicit trust is usually obtained from user-item interactions (i.e., ratings), and explicit trust is extracted from the user relationships (who they trust and upto what extent).

Similarity: Users with similar preferences or behavior tend to be interested in the same products, even though they may not know each other [36]. The preference similarity of two customers can be estimated according to their product purchases or rating records. The similarity measures (i.e., Vector Space Similarity (VSS) and Pearson Correlation Coefficient (PCC)) have been incorporated in social recommender systems [13, 74]. Trust relations are typically bidirectional and equal in both directions. However, this is not true in real world relationships where trust relationships are

non-transitive [77]. Also in order to provide meaningful recommendation, trust must reflect user similarity to some extent; recommendations only make sense when obtained from like-minded people exhibiting similar taste [1, 57].

Preference: For personalized recommendations, there are two ways to capture users' preferences [48]: implicit and explicit. In implicit feedback [26], the system infers users' preferences by monitoring different actions of users such as purchasing history, browsing history, clicks, email contents, etc. Thus, this type of feedback reduces the burden from user. In explicit feedback [118], recommender systems prompt users to provide ratings for items in order to reconstruct and improve its model. The drawback with this method is that it requires some efforts from the users. However, it seems that explicit feedback still provides more reliable data, since it does not involve extracting preferences from actions [4, 16]. However, an implicit feedback system lacks these characteristics, at it observes the user's actions and makes inferences about the users interests based on these actions. Matrix factorization models can use both implicit and explicit feedbacks from the system [64]. In [36], a framework has been developed to recommend similar users and resources based on social network analysis. The work in [136] uses a social network to develop a recommender system for peer-to-peer knowledge sharing.

Social Influence: Users with closer social ties to others are much worth to be believed and are more powerful in influencing others [66]. In [75], user's opinion is modeled based on her own and her friends' opinions which reflect real life social interactions. Also, social influence might create shopping intention for people to consume a product [62] and is thus one of the important factors for predicting the potential purchasing intention of a customer [66].

2.2.4 Rating Prediction

Predicting the rating of a product that a user would have given is challenging. User-based algorithms predict rating given by a user to an item based on the ratings by similar users, whereas, item-based algorithms estimate the rating based on the ratings of similar items previously chosen by the user. These methods find similar users [13, 50] or similar items [37, 71, 116] for providing accurate predictions. Methods used in traditional recommender systems are mostly based on user-item rating matrix. These algorithms usually fail to find similar users since density of ratings in user-item rating matrix is often less than 1 percent [71]. These methods assume that there are at least two users who have rated some common items, which might not be possible for a sparse user-item rating matrix. Moreover, almost none of the memory-based and model-based algorithms can handle users who never rated any item [51].

2.2.5 Trust Models

Several models have incorporated trust into e-commerce decisions [82] which use trust as a tool to identify and distinguish acceptable data from unacceptable data [56]. Collaborative filtering methods are most effective when users have expressed enough ratings. Since these methods need users to have mutually rated items, they perform poorly with respect to cold start users. Also, similarity metrics would not be helpful with cold start users. However the trust-based recommenders can make better recommendations since users can benefit from their trust relationships as well.

Some methods use random walks, so to use enough ratings without suffering from noisy data due to being far from source. TrustWalker proposed in [54] is a random walk model which combines trust-based and item-based recommendations. There are some algorithms such as Eigentrust [59], Appleseed [139] and another algorithm in [111] which use principal eigenvector to make trust computations. However, these methods produce ranks of trustworthiness of users, so they would be suitable for systems where ranks are considered. The TidalTrust model finds all raters with the shortest path from the source user and aggregates their ratings weighted by the trust between them. Another method is MoleTrust [6] where computation of trust value between two users is based on backward exploration. Also, trust values in recommender systems help to predict the behavior of those users who have rated fewer products [74]. A trust metric in [3] has been proposed in order to discover which users are trusted by members of an online network. Each user is assigned a capacity, where trust values will need to be normalized within that capacity, and for computing the trust of the entire network is required. Moreover, it only produces the nodes to trust; not the value of the trust. Since there is no distinction between trusted users, and number of users to trust is independent of users and items, this method is not appropriate for trust-based recommendation systems. Other work such as [84] uses similarity measures, however it is only designed to be used in systems with binary trust ratings.

2.2.6 User Preference Model

To provide personalized recommendation, there are two ways to capture users' preferences [48]: implicit and explicit. The implicit method gathers users' behavior to obtain their preferences [26]. Matrix factorization models built in [64] use implicit feedback from the system. The explicit method filters and analyzes interactions and feedback to obtain users' specifications [118]. In [78] a user-item matrix is considered with users' social trust graph to build a latent low-dimensional matrix for providing a better recommendation. Users opinion is modeled based on her own and her friends' opinions which reflect real life social interactions [76]. The similarity between users is incorporated in social recommender systems [74]. Also social recommendation algorithms with social regularization terms is used in [75] to constrain matrix factorization objective functions. In addition, using trust values in recommender systems would help to predict the behavior of those users who have rated fewer products [74].

2.3 Anomaly Detection in User Behavior

Recommender systems are vulnerable to profile-injection attacks: these systems collect user profiles, which represent the taste of users, and make recommendation based on these tastes. Profile injection attack was first introduced in [100]. Different attacks and defenses have been identified since then. In [88] the authors present various recommendation algorithms that use different model-based methods, particularly techniques based on k-means and probabilistic latent semantic

analysis (pLSA) that compare the profile of an active user to aggregate user clusters, rather than the original profiles. In [46] the authors presented different attack types, detection methods, robustness analysis and cost benefit analysis.

Generic of model-specific attributes capture different statistical features of user profiles which could be used to classify users. In [25], multiple metrics to distinguish between authentic and fake profiles such as number of prediction-differences (NPD), standard deviation in user's ratings, degree of agreement with other users, degree of similarity with top neighbors, and rating deviation from mean agreement (RDMA) were proposed. In [18, 19], weighted deviation from mean agreement (WDMA) and weighted degree of agreement (WDA) were proposed. Though WDMA is derived from RDMA, it puts more weight on rating deviations for sparse items which provides higher information gain. Length Variance (LengthVar) that measures the difference between a given profile rating and system's average rating was also proposed. Three classification methods have been used in [129] which were simple nearest-neighbor classification using kNN, decision-tree learning using C4.5, and support vector machine (SVM) classifier. The attributes used were RDMA, WDA, WDMA, degree of similarity with top neighbors (DegSim), and LengthVar. The discussed attributes so far are included in the generic category.

Model-specific attributes were also introduced in [18, 19] including Filler Mean Variance, Filler Mean Difference, Profile Variance for average attacks, and Filler Mean Target Difference (FMTD) for segment attacks. Also in [129], these model-specific attributes were used with Filler Average Correlation attribute for random attacks. Another method for attacker detection is based on out-lier identification. These out-lier detection methods can be based on distance, density,

clustering, or depth. In [87], the authors mentioned that attacker profiles are highly correlated, thus members of small clusters are considered attack profiles. Clustering has an advantage of being completely unsupervised compared to other approaches used for out-lier detection.

Statistical analysis such as statistical process control (SPC) can also help in detecting products that are under attack [12]. Also a method has been proposed [135] that can detect random attacks by computing the log-likelihood of each rating profile given the low dimensional linear model of the rating matrix. However, it cannot detect the average attacks. Algorithms using neighborhood selection and similarity weight transformations for attack detection and defense were proposed in [98].

CHAPTER 3: PROBABILISTIC INFORMATION DIFFUSION IN SOCIAL NETWORKS

To analyze information diffusion (recommendation spreading), we consider a probabilistic model where a node passes the information it receives to its connections in a probabilistic manner. A node recommends a product to all of its neighbors with probability $0 < w < 1$. Generalizing, each node will recommend to all its neighbors. However, the recommendation of a product has to begin at some node which we refer to as the *origin* node. We are interested in knowing how the recommendation will spread in the network, given that a node recommends to its neighbor(s) in a probabilistic manner.

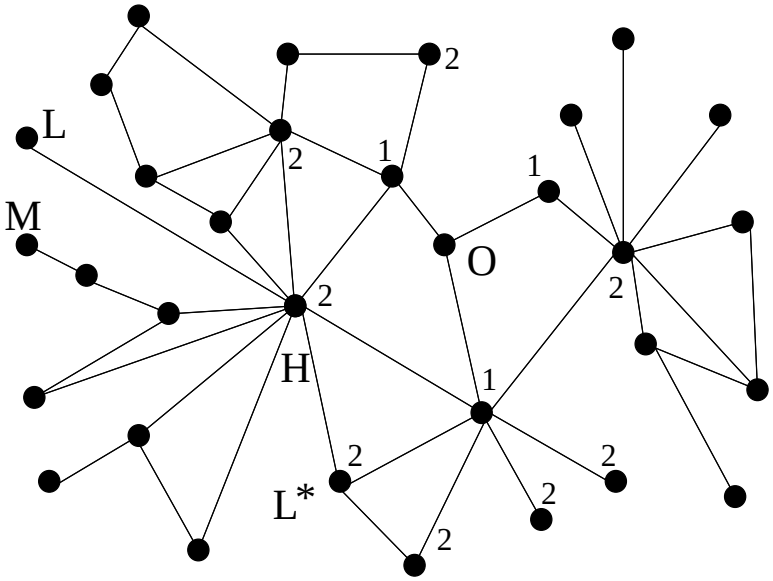

Figure 3.1 An example network showing origin O and nodes marked with hop-count from O.

3.1 Probabilistic Information Diffusion

Our objective is to investigate i) what fraction of nodes will receive the recommendation, ii) the effect of the distance of a node from the origin, iii) the impact of w, and iv) the effect of the nature of the origin (i.e., hub or not).

Let us consider an origin node O as shown in Fig. 3.1. The nodes that are direct neighbors of O (i.e., 1 hop neighbors) are referred to as layer-1 nodes. Similarly, nodes that are 2 hops away from O are referred to as layer-2 nodes, and so on. To have better tractability, we proceed by finding the probability that a layer-1 node will receive the recommendation from O. Then, we find the probability that a layer-2 node will receive the recommendation from *one or more* layer-1 nodes. We continue the process till we reach the node(s) in the farthest layer (e.g., node M in Fig. 3.1). It is to be noted that a node at layer-i could receive the recommendation from nodes at layer-$(i-1)$, nodes that belong to layer-$(i+1)$, and nodes that belong to the same layer-i.

We consider both previous layer and the next layer as recommendations propagate in all directions. At each time t, nodes that are influenced at time $t-1$ try to influence their neighboring nodes with some probability. Influence spreading is known to be an NP-hard optimization problem [61].

We use a directed probability model for recommendation probability spreading which considers the probability of a node being recommended by previous layer nodes (i.e., the in-degree), next layer nodes (i.e., the out-degree) and nodes that belong to the same layer. Obviously, the

enumeration of the layers is based on the distance and direction from the origin O. Since it would be impossible to compute the layers in advance, we compute the layers whenever a recommendation is initiated by a node. That node determines the out-degree, in-degree, and the same-layer probabilities. As pre-determining the distances of all nodes would cause significant amount of computation and storage overheads, we compute the layers dynamically when the origin node is known.

3.2 Recommendation Probabilities

To find the recommendation probability, we first consider the probability that a node is recommended from node(s) from the previous layer, i.e., nodes that are closer to the origin. We refer to this as the *outward* probability. Similarly, *inward* probability is defined as the probability that a node is recommended from nodes from a latter layer, i.e., nodes that are farther from the origin. We also define *same-layer* probability as the probability that a node gets recommendation from nodes in the same layer.

It is to be noted that these probabilities are dependent on their respective degree distributions. Thus we decompose the degree distribution into out-degree, in-degree, and same-layer degree distributions. Without loss of generality, we assume out-degree and in-degree distributions are identical and follow the power law distribution, denoted by $p_{out}(k)$ and $p_{in}(k)$, respectively. The same-layer degree distribution, denoted by $p_{sl}(k)$ follows a binomial distribution as discussed

in section 3.2.3. The outward, inward, and the same-layer probabilities can be combined to obtain the total probability.

In order to find the inward and outward probabilities, it is essential to know the inward and outward degree distribution. Assuming k_{out} is the number of out going edges, k_{in} is the number of incoming edges, and k_s is the number of edges in the same layer, the total degree of a node denoted by k is given by:

$$k = k_{out} + k_{in} + k_s \tag{3.1}$$

We consider a connected social network that obeys the power law for its degree distribution i.e., $p(k) = \alpha k^{-\gamma}$ and assuming identical inward $p_{in}(k)$ and outward $p_{out}(k)$ distribution as $p_{in}(k) = \alpha_{in} k_{in}^{-\gamma_{in}}$, and $p_{out}(k) = \alpha_{out} k_{out}^{-\gamma_{out}}$ where $-\gamma_{in}$ and $-\gamma_{out}$ can be approximated as $-\gamma$. As for the scale factors α_{in} and α_{out}, we assume both to be α_1.

To estimate inward and outward probabilities, we need to first find the distribution of $r = \frac{k_{out}}{k_{in}}$. The probability distribution for r is a joint distribution of variables k_{out} and k_{in} which is calculated as [90]:

$$p(r) = \int_0^\infty k_{in}(k_{in}r)^{-\gamma}\alpha_1(k_{in})^{-\gamma}\alpha_1 dk_{in} \tag{3.2}$$

$$p(r) = \int_0^\infty \alpha_1^2 k_{in}^{(-2\gamma+1)} r^{-\gamma} dk_{in} = \frac{\alpha_1^2 r^{-\gamma}}{-2\gamma+2} \tag{3.3}$$

Since k_{out} and k_{in} have identical distributions, the expected ratio (the average of $r = \frac{k_{out}}{k_{in}}$) would be 1. Thus,

$$\int_0^\infty rp(r) = 1 \tag{3.4}$$

From Eq. (3.3) and Eq. (3.4), we get

$$\alpha_1 = \sqrt{(-2\gamma + 2)(-\gamma + 2)} \tag{3.5}$$

With α_1 known, we can find the in- and out-degree distributions which can be used to find the inward and outward recommendation probabilities.

3.2.1 Outward Recommendation Probability

We compute the outward recommendation probability for one layer at a time, starting with layer-1 and moving outwardly away from the origin node.

Layer-1:

First layer nodes are immediate neighbors of the origin node, and therefore would get recommendation from the origin node with probability w. Thus the outward recommendation probability for all layer-1 nodes, denoted by P_1^{out}, is given by:

$$P_1^{out} = w \tag{3.6}$$

Layer-2: With the recommendation probability for layer-1 nodes known, we can find the recommendation probability for the layer-2 nodes. Being a layer-2 node necessarily means that it is connected to at least one layer-1 node. Thus, such a node can get the recommendation from one or more layer-1 nodes– the number of which is the in-degree of that node.

If a node has k links from the previous layer, i.e., an in-degree of k, then the probability of *not* getting recommended is $(1-w)^k$. Thus, getting a recommendation occurs with probability $1-(1-w)^k$. Since, $k \geq 1$ and is distributed as per $p_{in}(k)$, the average outward probability for a layer-2 node, denoted by P_2^{out}, can be found by the weighted sum of the probabilities. Thus,

$$P_2^{out} = P_1^{out} \sum_k p_{in}(k)(1-(1-w)^k) \tag{3.7}$$

The term P_1^{out} appears because each of the layer-1 nodes will get the recommendation with probability P_1^{out} as was shown in Eq. (3.6).

Layer-L: Continuing in the same manner and noting that layer-L nodes can only get recommended from directly connected layer-$(L-1)$ nodes, we can compute the recommendation probability for a layer-L as:

$$P_L^{out} = P_{L-1}^{out} \sum_k p_{in}(k)(1-(1-w)^k) \tag{3.8}$$

3.2.2 Inward Recommendation Probability

We compute the inward recommendation probability from the outer most layer and move towards the origin.

Layer-L:

The last layer nodes are farthest from the origin and thus cannot get recommendation from any farther node; thus their inward probability is zero. Thus,

$$P_L^{in} = 0 \tag{3.9}$$

Layer-$(L-1)$:

The inward probability for the layer-$(L-1)$ depends on what the recommendation probability was from layer-L nodes which was obtained in Eq. (3.9) and Eq. (3.8). Thus, we get P_{L-1}^{in} as:

$$P_{L-1}^{in} = (P_L^{out} + P_L^{in}) \sum_k p_{out}(k)(1-(1-w)^k) \tag{3.10}$$

Layer-1: Continuing to move towards the origin, we get P_1^{in} as:

$$P_1^{in} = (P_2^{out} + P_2^{in}) \sum_k p_{in}(k)(1-(1-w)^k) \tag{3.11}$$

3.2.3 Same-layer Recommendation Probability

A common feature of social networks is the circle or triangle of friends one knows. This tendency to cluster is reflected in the clustering coefficient [127]. Fig. 3.2 shows nodes A, B, C, D, and E that belong to the same layer (Layer-1) by virtue of being directly connected to the origin O. The same-layer probability depends on the number of links among the nodes in a given layer which is directly related to the clustering coefficient of the network. Consider node i that is connected to k_i nodes. Suppose those k_i nodes have E_i links/edges among them. Then the clustering coefficient of node i, denoted by C_i, is the ratio of E_i and the total number of links possible among the k_i nodes i.e., $C_i = \frac{E_i}{\binom{k_i}{2}}$ [10]. Though C_i is for node-i, the average clustering coefficient, C, of the network could be found [49], which we use as the connection probability of having links within the same layer.

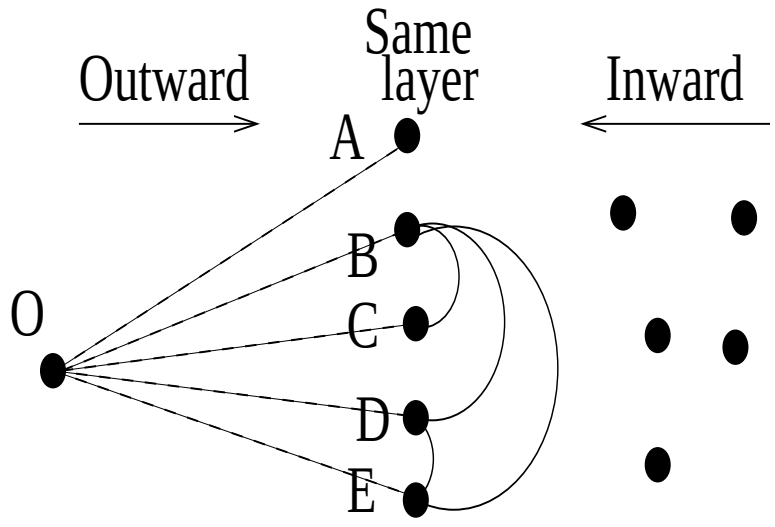

Figure 3.2 Example: Links among nodes within the same layer are shown with solid lines; links from O are shown with dashed lines; links of far away nodes are not shown.

The probability of having k links among the n nodes in any layer is binomially distributed as we assume the links appear with the same probability C and are independent of each other. Thus, the same-layer degree distribution of having k links, $p_{sl}(k)$, is given by

$$p_{sl}(k) = \binom{n}{k} \times C^k \times (1-C)^{(n-k)} \tag{3.12}$$

Given the degree distribution, we can find the same-layer recommendation probability as:

$$P_L^{sl} = \sum_n \sum_k p_{sl}(k)(1-(1-w)^k) \tag{3.13}$$

3.2.4 Total Recommendation Probability

Total recommendation probability is calculated by combining the inward, outward and the same-layer probabilities for each node. Noting that a node in layer-i could get a recommendation from one of the three layers, we proceed by finding the probability of *not* getting recommended– given by $(1-P_i^{out})$, $(1-P_i^{in})$, and $(1-P_i^{sl})$. The probability of not getting recommended from any layer is: $(1-P_i^{out})(1-P_i^{in})(1-P_i^{sl})$. Thus, the total probability of a layer-i node getting recommended is:

$$P_i^{tot} = 1-(1-P_i^{out})(1-P_i^{in})(1-P_i^{sl}) \tag{3.14}$$

Discussion: We considered the outward, inward, and same-layer recommendations only once. However in real systems, a node might get recommended multiple times over a time span

necessitating the need to consider the probabilities for the second time, third time and so on. We argue that those probabilities would be orders of magnitude smaller (as they multiply with each other multiple times) than the probability obtained for the first outward, inward, and same-layer recommendations. Thus, we ignore the those higher order terms.

3.3 Experimental Results

In order to verify the proposed mathematical framework, we used the data of Facebook from Stanford Network Analysis Project (SNAP). This data set includes 4039 nodes and 88234 edges. We confirm that the network is scale-free as the degree distribution follows a power law distribution as shown in Fig. 3.3 (in linear scale) and in Fig. 3.4 (in log-log scale). Using curve fitting, we obtain the scale (α) of the distribution as 4.928 and the exponent (γ) as 2.9277. Thus, the degree distribution is $p(k) = 4.928k^{-2.9277}$.

We analyze the inward, outward, and the same-layer recommendation probabilities of each layer based on mathematical framework described in section 3.2. The degree distribution for outward and inward probabilities are power law distribution $p_{in}(k)$ and $p_{out}(k)$ as defined earlier. Beside the probability w, we consider four types of origin nodes:

Case i): A highly connected node (i.e., hub shown as node H in Fig. 3.1).

Case ii): A neighbor of a hub but connected to other(s) (node L^* in Fig. 3.1).

Case iii): A neighbor of a hub that is only connected to the hub (node L in Fig. 3.1), which we call a *leaf*.

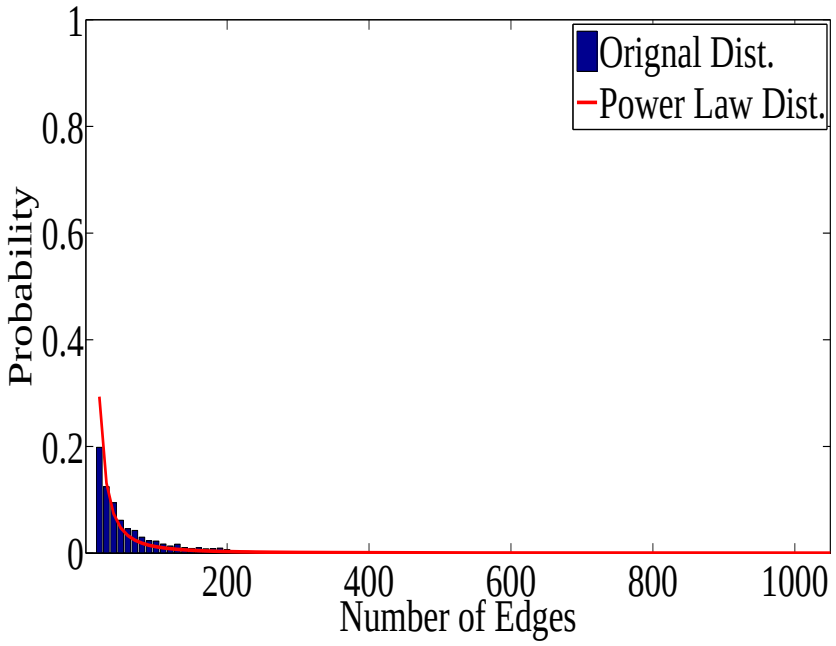

Figure 3.3 Degree distribution in linear scale

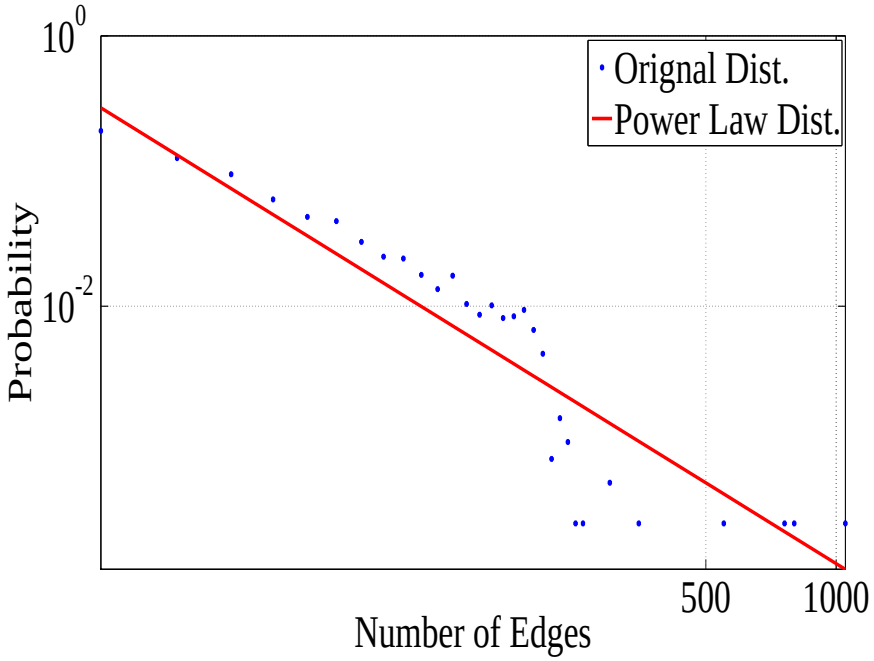

Figure 3.4 Degree distribution in log-log scale

Case iv): A leaf node far from the hub (node M in Fig. 3.1).

Our objective is to show the effects of the location and the degree of the originating node.

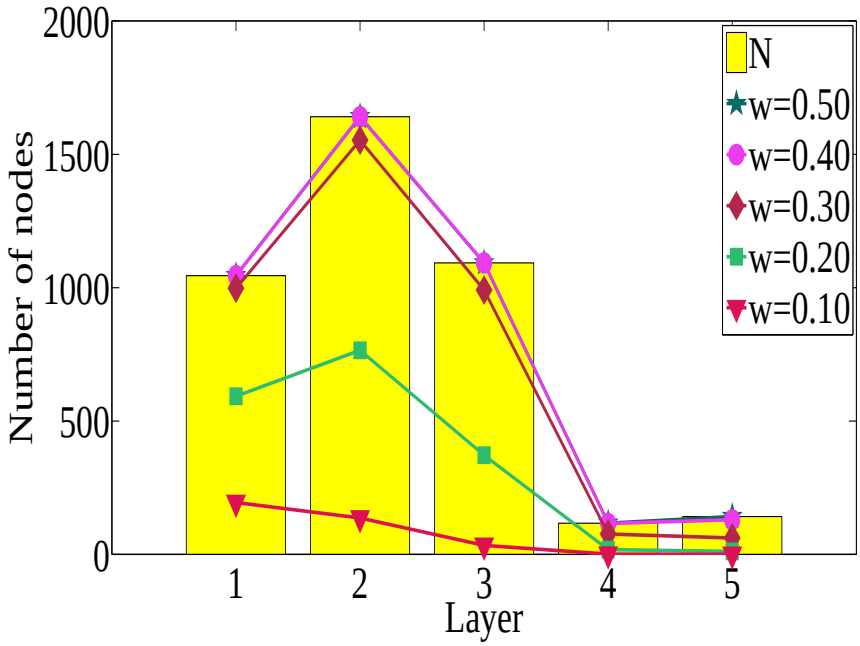

Figure 3.5 Total number of nodes in each layer with the number of recommended nodes with a hub as the origin

3.3.1 Case (i): Hub as the origin (H)

Fig. 3.5 shows the number of nodes in each layer along with the number of recommended nodes for $0.1 \leq w \leq 0.6$. The first layer contains 1047 nodes implying that the origin node is a hub. We choose the highest degree node as a representation for all high degree nodes. All the nodes in the network also lie within 5 hopes from the origin.

We show the outward, inward, and same-layer recommendation probabilities as obtained from Eq. (3.8), Eq. (3.10), and Eq. (3.13) in Figs. 3.6, 3.7, and 3.8, respectively. The total recommendation probability is shown in Fig. 3.9. Since the origin node is a hub, there is a relatively high number of nodes in the first layer. As evident from Fig. 3.5, most of the nodes are within the first three layers and not many in layers 4 and 5.

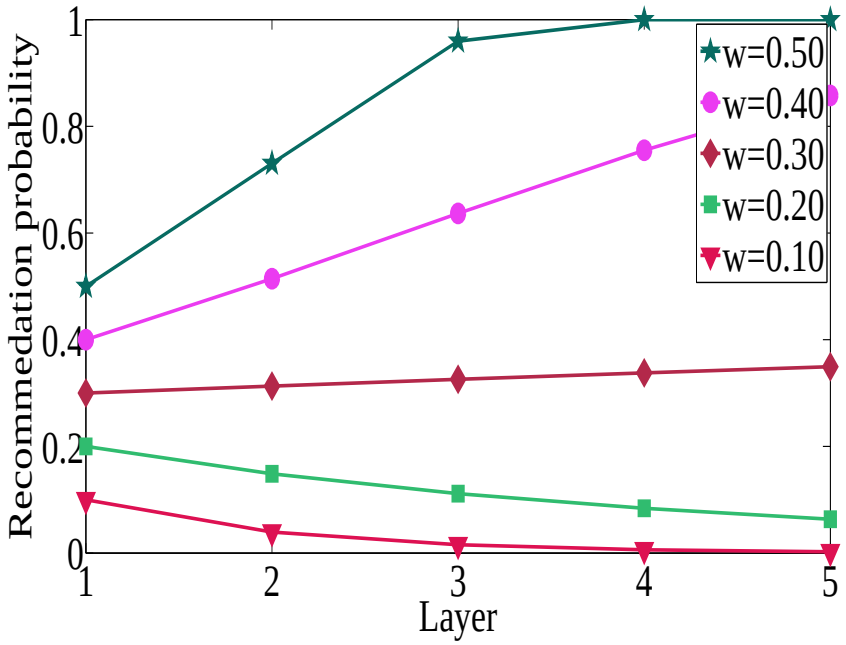

Figure 3.6 Outward recommendation probability with a hub as the origin.

From Fig. 3.9, we observe that the third layer has less recommendation probabilities than the first two layers. Comparing inward, outward, and same-layer probabilities, we can see the same-layer probability is directly related to the number of nodes in that layer. However, both inward and outward probabilities are independent of the number of nodes in each layer. As expected, both inward and outward probabilities decrease with increasing distance from the origin.

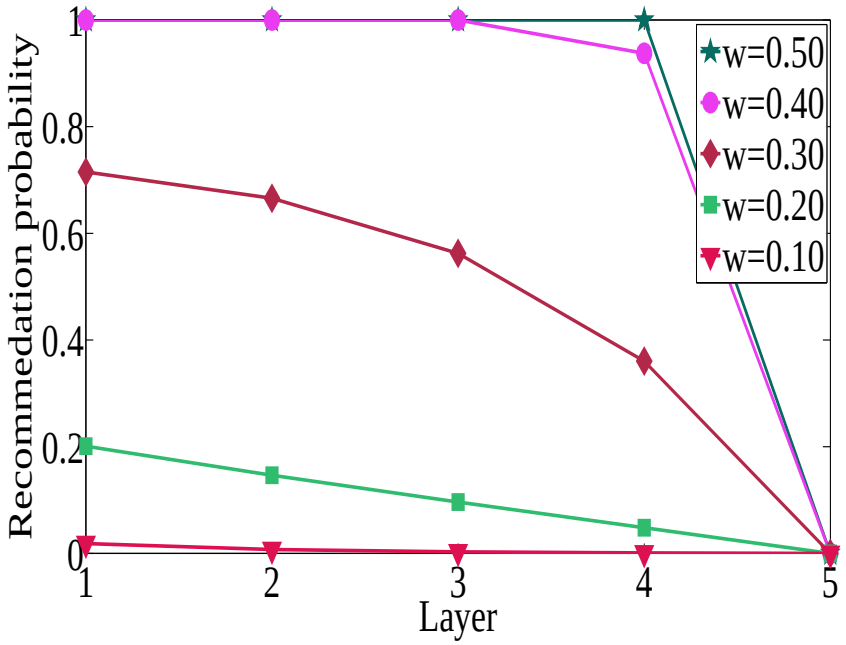

Figure 3.7 Inward recommendation probability with a hub as the origin.

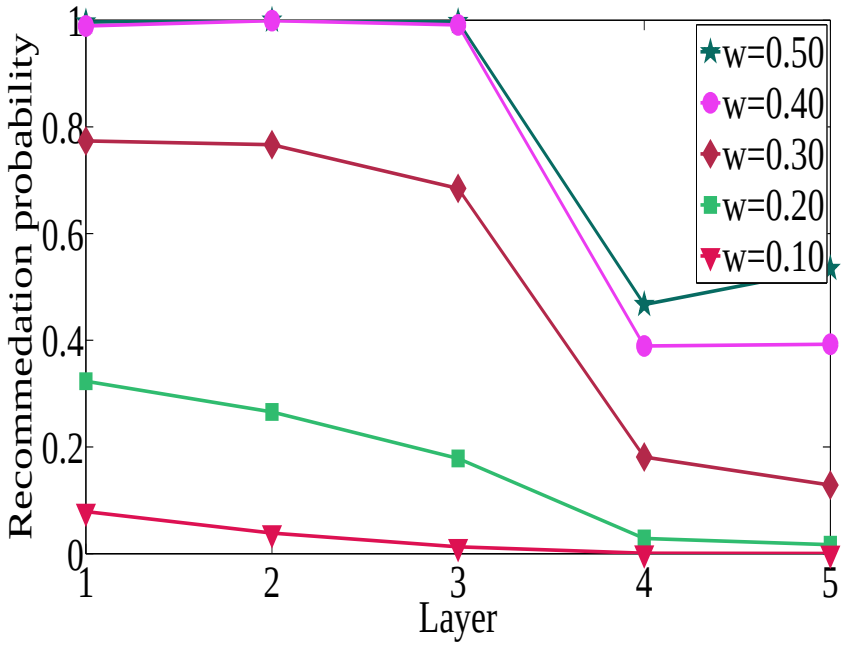

Figure 3.8 Same-layer recommendation probability with a hub as the origin.

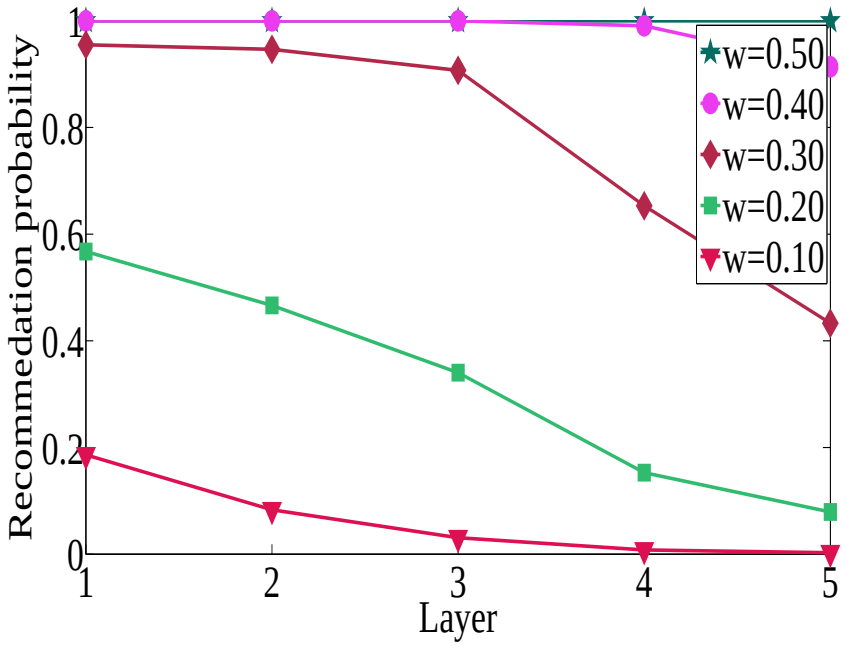

Figure 3.9 Total recommendation probability with a hub as the origin.

3.3.2 Case (ii): Neighbor of a hub as the origin (L^*)

The number of nodes along with the number of recommended nodes in each layer is shown in Fig. 3.10. Comparing with the case when the hub was the origin, we see the same trend but the high number of nodes continues for one more layer (4th layer comparing to third layer in Fig. 3.5). This is due to an additional layer that the recommendation should travel to get to a hub. Similarly, the total recommendation probability shown in Fig. 3.11 decreases a layer later than compared to Fig. 3.9. For this setting and the latter ones, we do not show the inward, outward, and same-layer probabilities.

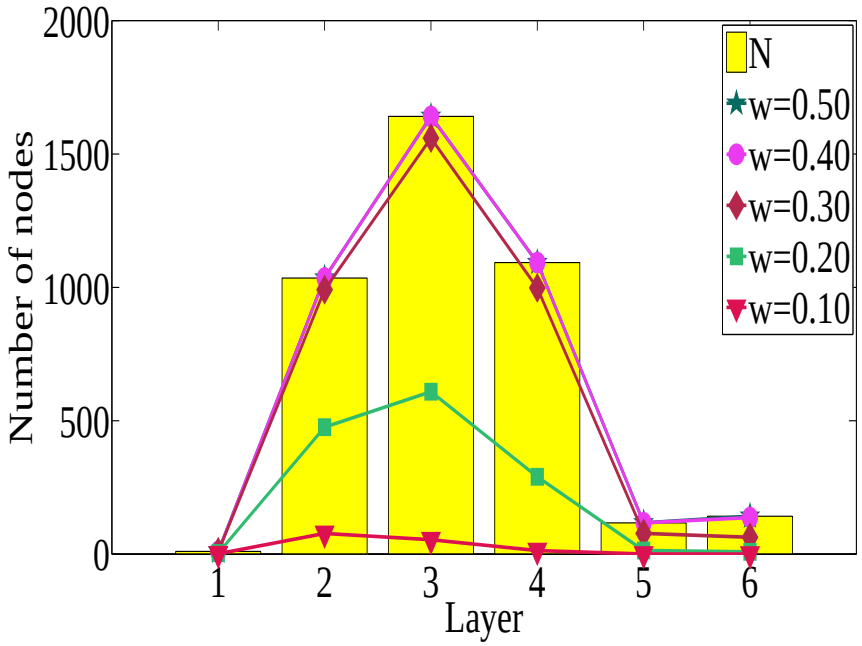

Figure 3.10 Total number of nodes in each layer with the number of recommended nodes with a neighbor of a hub as the origin

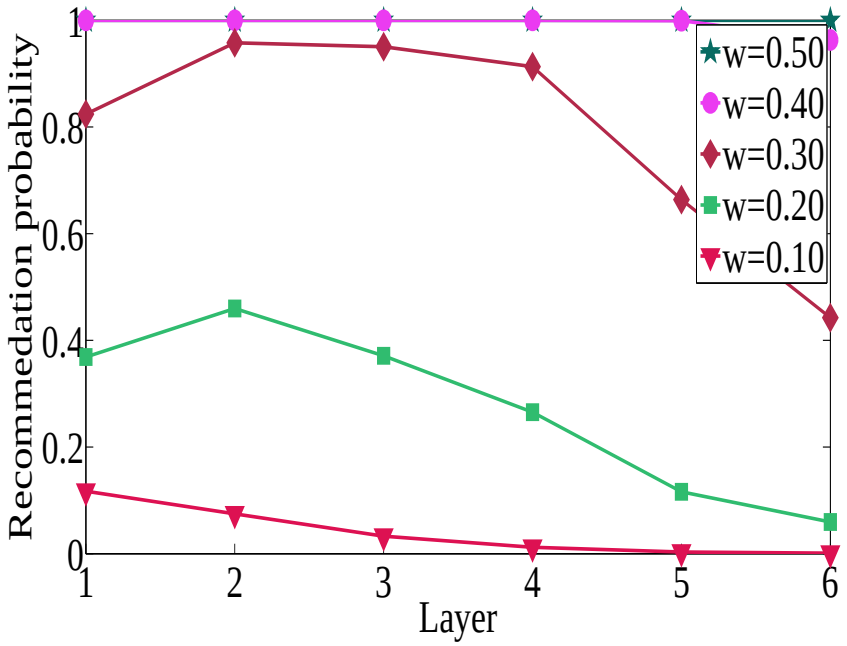

Figure 3.11 Total recommendation prob. with a neighbor of a hub as the origin.

3.3.3 Case (iii): Leaf as the origin (L)

When the origin node is *only* connected to a hub, the recommendation probability is dominated by the neighboring hub (see Figs. 3.9 and 3.11). Despite the minor differences, the pattern is almost the same as the two previous cases. However, the peak occurs one layer further (at the 4th layer). This implies that having a neighboring hub has a great impact on the number of nodes that receive the recommendation.

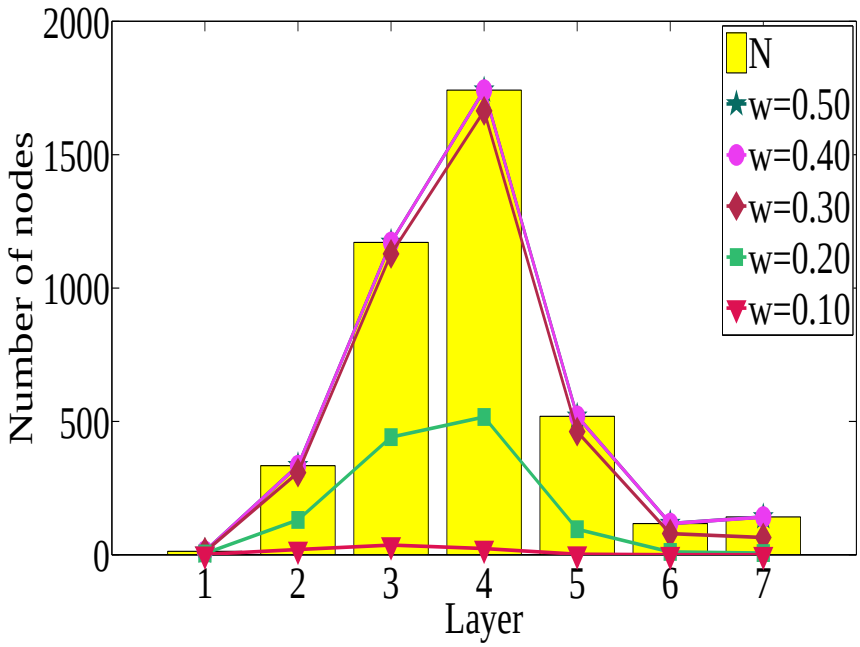

Figure 3.12 Total number of nodes in each layer with the number of recommended nodes with a leaf neighbor with a hub as the origin.

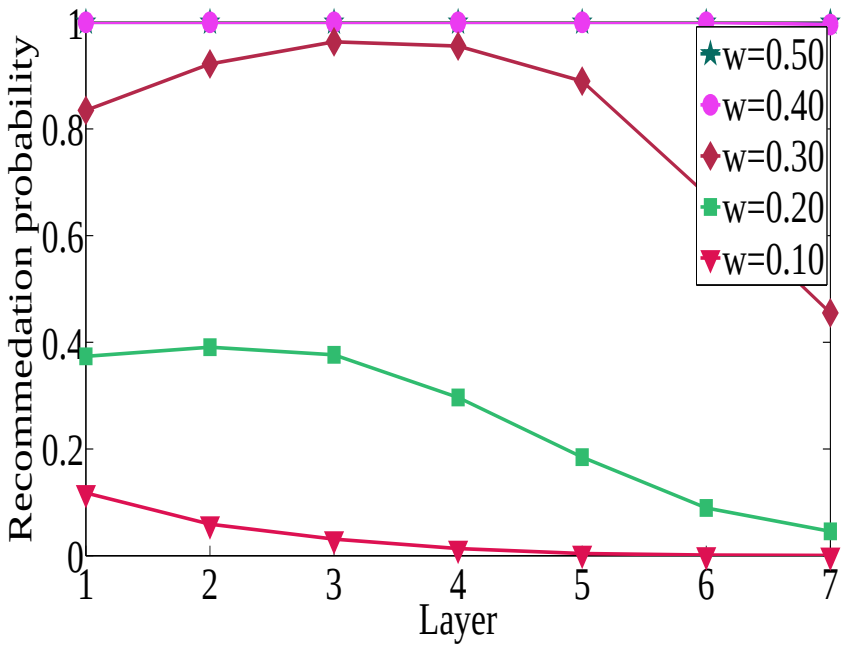

Figure 3.13 Total recommendation probability with a leaf neighbor with a hub as the origin.

3.3.4 Case (iv): Leaf far from the hub as the origin (M)

When the origin node is a leaf somewhat far from the hub, the maximum number of recommended nodes appears further from the origin node compared to all the previous cases. As shown in Fig. 3.14, the majority of the recommended nodes are within the sixth layer while the number of recommended nodes after the 4th layer is relatively small. With the origin far from most of the nodes (i.e., high average distance to others), the recommendation needs to travel more layers to reach more nodes which decreases the total recommendation probability and shifts the peak value to the right.

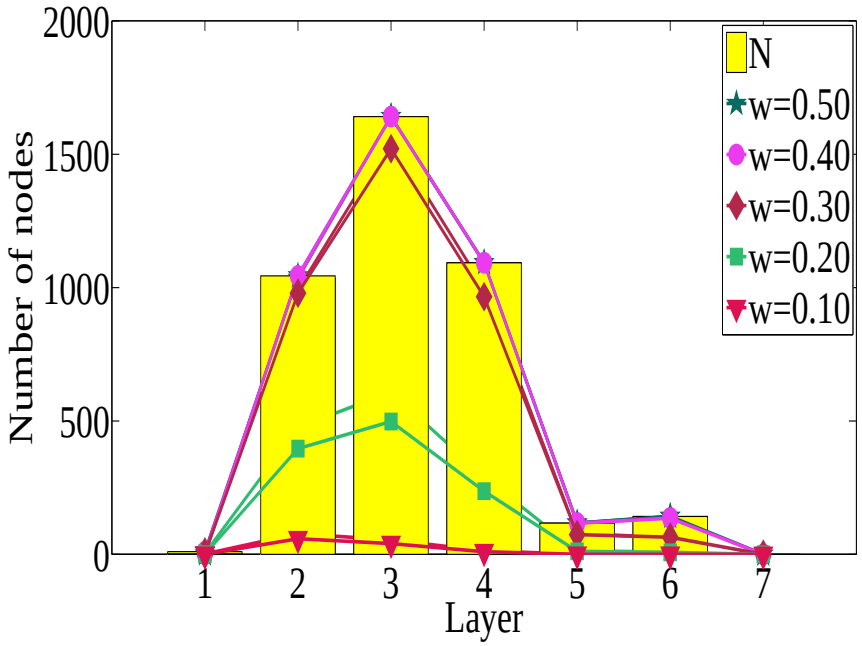

Figure 3.14 Total number of nodes in each layer with the number of recommended nodes with a leaf as the origin.

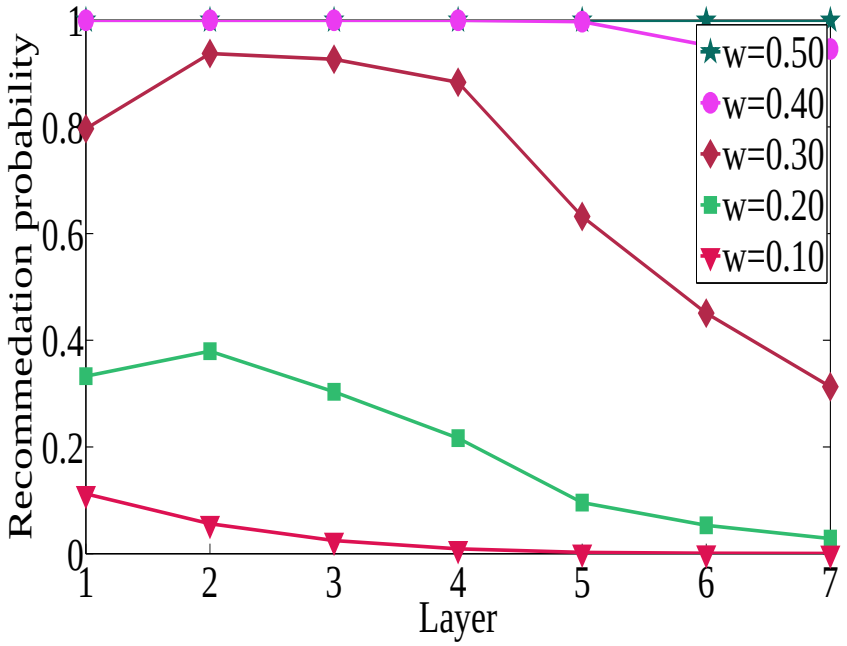

Figure 3.15 Total recommendation probability with a leaf as the origin.

3.3.5 Comparing the four cases

We compare the impact of the origin– the fraction of the recommended nodes is shown in Fig. 3.16. As expected, when the origin node is a hub, the spread of the recommendation is the highest. Also, we see that when the origin is not a hub but close to a hub, there is not much difference in how the recommendation propagates since the recommendation process is dominated by the neighboring hub. If the origin node is far from the hub, it has the lowest spreading probability.

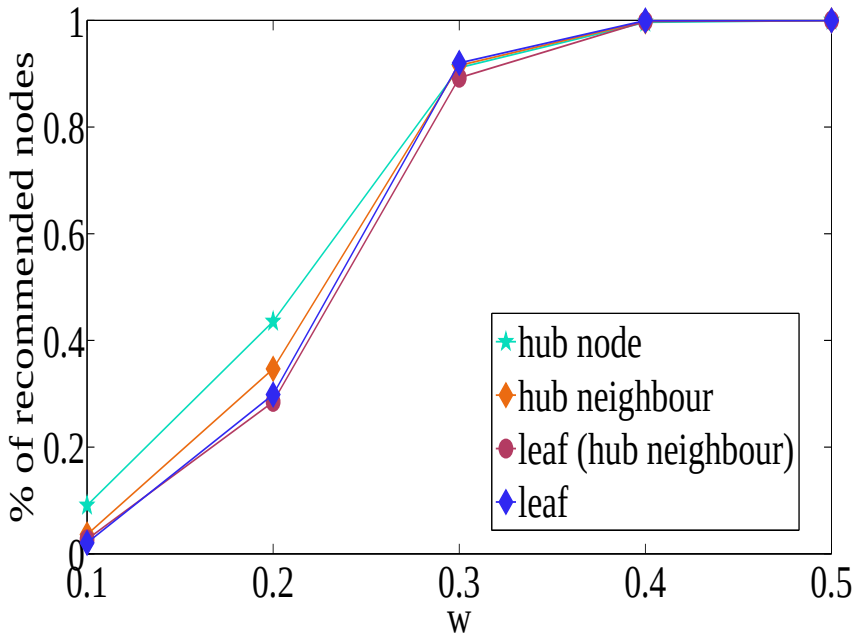

Figure 3.16 Comparing total recommendation probabilities based on type of the origin and distance from the hub.

3.4 Summary

With the growing popularity of social networks, recommendation systems are becoming important due to their commercial, social, and political impacts. Businesses are exploring ways on how to best exploit social links to spread recommendation about their products. In this chapter, we developed a model to investigate how a recommendation spreads when all nodes pass on the recommendation to their neighbors in a probabilistic manner. In our model, the nodes are categorized in layers based on their distances from the origin node. We derived the probabilities of nodes in any layer getting a recommendation from a node in the previous layer, from a node in the next layer, and from a node in the same layer. We validated the theoretical framework on a Facebook dataset and studied how various node parameters such as degree and distance from the origin affect the recommendation process.

CHAPTER 4: NON-LINEAR INFORMATION DIFFUSION IN SOCIAL NETWORKS

Having developed a probabilistic model, we proceed to develop a Diffusive Logistic model to characterize the temporal dynamics of information diffusion in online social networks. The logistic model is a non-linear model that represents the dynamics of the population in the system where the growth rate (reproduction) is proportional to the current population and the available resources [93]. This model has been used for different populations and growth prediction of bacteria and tumors.

4.1 Non-Linear Information Diffusion

In social networks, the information spreads through the users' interactions such as commenting, liking, forwarding, and other activities. We seek to answer the following question. Given an information initiated from a source, what is the fraction of influenced user after a period of time? Let the number of the influenced users at time t be denoted by $I(t)$. The growth process is modeled using the Logistic model which captures the user influence and is defined as:

$$\frac{\partial I(t)}{\partial t} = r \times I(t) \times (1 - \frac{I(t)}{K}) \tag{4.1}$$

where r denotes the intrinsic growth rate of influenced users and measures how fast the information spreads and K shows the carrying capacity of the influenced users. K represents the number of users that can be potentially influenced by a specific news or a story. Parameter K usually is assumed to be a constant; however, there are some evidences against this assumption. Figure 4.1 shows $I(t)$ for six different stories in Digg's data set. The figures clearly show a change in temporal dynamics of information spreading. For instance, the pattern of $I(t)$ for Story 70 changes significantly at $t = 1.8 \times 10^5 s$. The same pattern is observed for the other stories as well.

To better understand the pattern observed in Figure 4.1, one should investigate the mechanism utilized by Digg. In Digg, information spreading happens when a user votes for a news that his followee submitted. Also if a news makes it to the front page, users who are not the submitter' s follower can vote for it. Therefore, it can be concluded that the abrupt change in the trend of $I(t)$ (as explained for Figure 4.1) is attributed to the state when the story moves to the front page. This transition can also be explained in terms of change in the carrying capacity K [31]. Initially, the story can only be seen by the followers which indicated relatively smaller number of potential readers and therefore small K. However, moving to first page, increases the visibility of the story and consequently raises the carrying capacity. Based on the same reasoning, one can also expect another change in K when the users lose interest in the story since it is not new or interesting anymore. This is schematically shown in Fig. 4.2. The initial (before moving to the front page), the secondary (after moving to the front page), and the final carrying capacities are denoted by K_1, K_2, and K_3 respectively. It is also assumed that a change in the carrying capacity does not happen immediately and occurs between [I_1, I_2] and [I_3, I_4]. For the sake of simplicity, we assume that

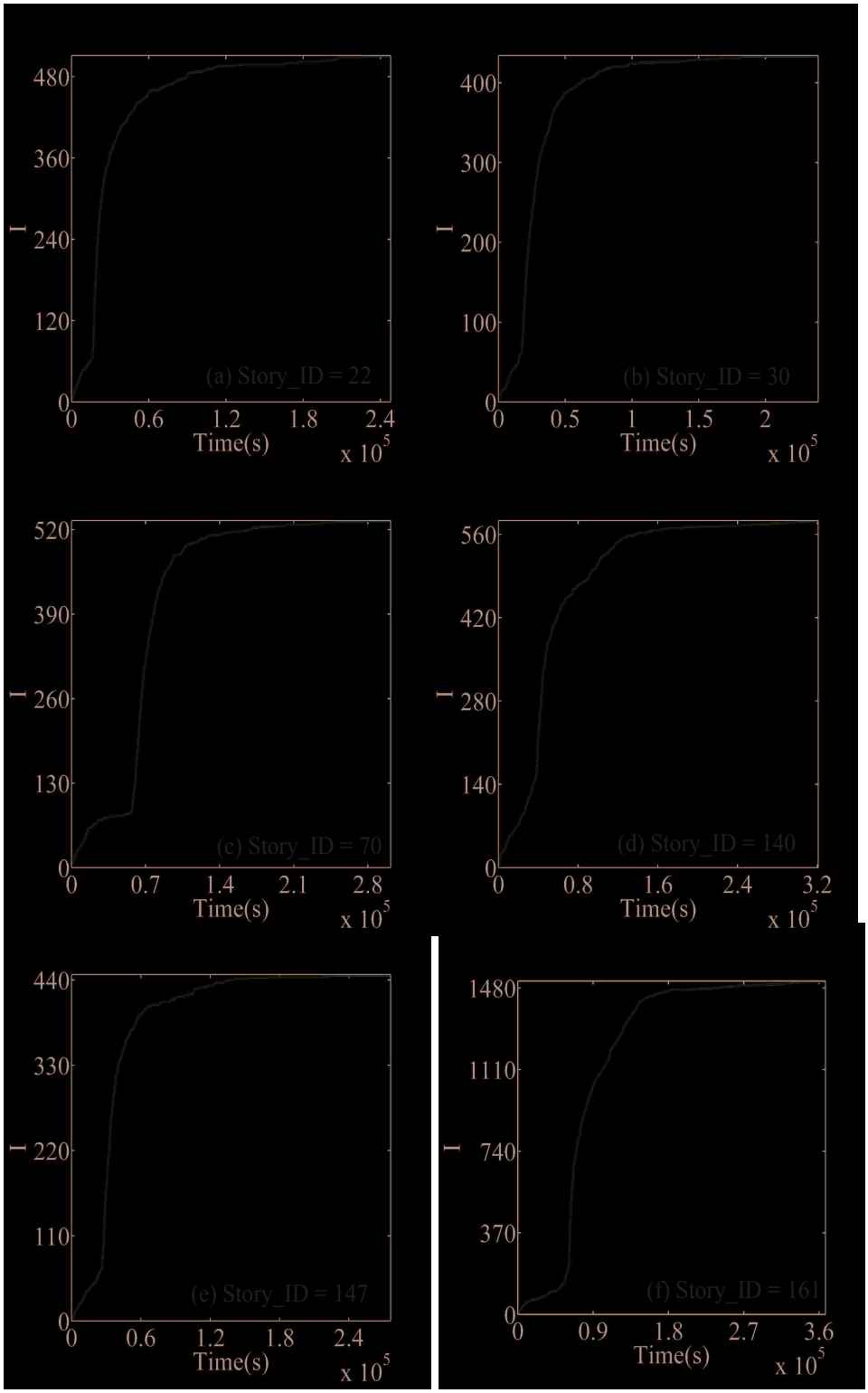

Figure 4.1 $I(t)$ for six different stories from Digg's data set.

K changes linearly during the transition phases between [I_1, I_2] and [I_3, I_4]. I_1 denotes when the followers of the submitter act on the story. With more and more followers acting on the story, K increases till I_2 when no further votes are made. This is when the carrying capacity is at its maximum, denoted by K_2. At I_3, the news loses its interest and is removed from the front page. The carrying capacity decreases; nevertheless, it is not zero as the news has already been exposed to a large number of users. Based on this discussion, Equation (4.1) can be written as:

$$\frac{dI(t)}{dt} = r \times I(t) \times (1 - \frac{I(t)}{K(I(t))}) \tag{4.2}$$

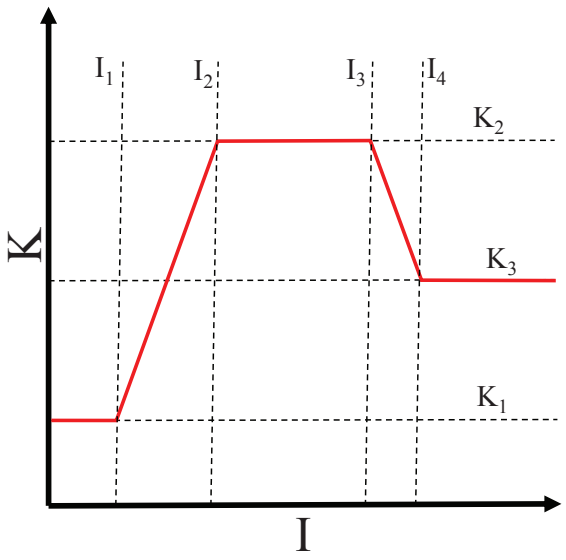

Figure 4.2 Schematic representation of temporal dynamics of carrying capacity.

We define parameter K as follow:

$$K(I) = \begin{cases} K_1 & I(t) \leq I_1 \\ \frac{K_2 - K_1}{I_2 - I_1}(I(t) - I_1) + K_1 & I_1 < I(t) \leq I_2 \\ K_2, & I_2 < I(t) \leq I_3 \\ \frac{K_2 - K_3}{I_3 - I_4}(I(t) - I_2) + K_2, & I_3 < I(t) \leq I_4 \\ K_3, & I(t) > I_4 \end{cases} \tag{4.3}$$

4.2 Experimental Results

4.2.1 Dataset Description

To validate our diffusion model we used the Digg dataset. Digg is a news sharing website where users post news links and also vote and comment on submitted news story. Users form following relationship resulting in a directed social graph The initiator or the source of the news is the first user who posts the news link. The data is time stamped based on the voting time. Diffusion of the story happens in two different ways: 1) a user shares a news link which all his followers can see and by voting for that news it become visible to their followers as well, and 2) high popularity of the news would bring it to the front page, which makes the non-friends/followers able to see the news and vote for it. This description makes the Digg dataset suitable for analysis of information

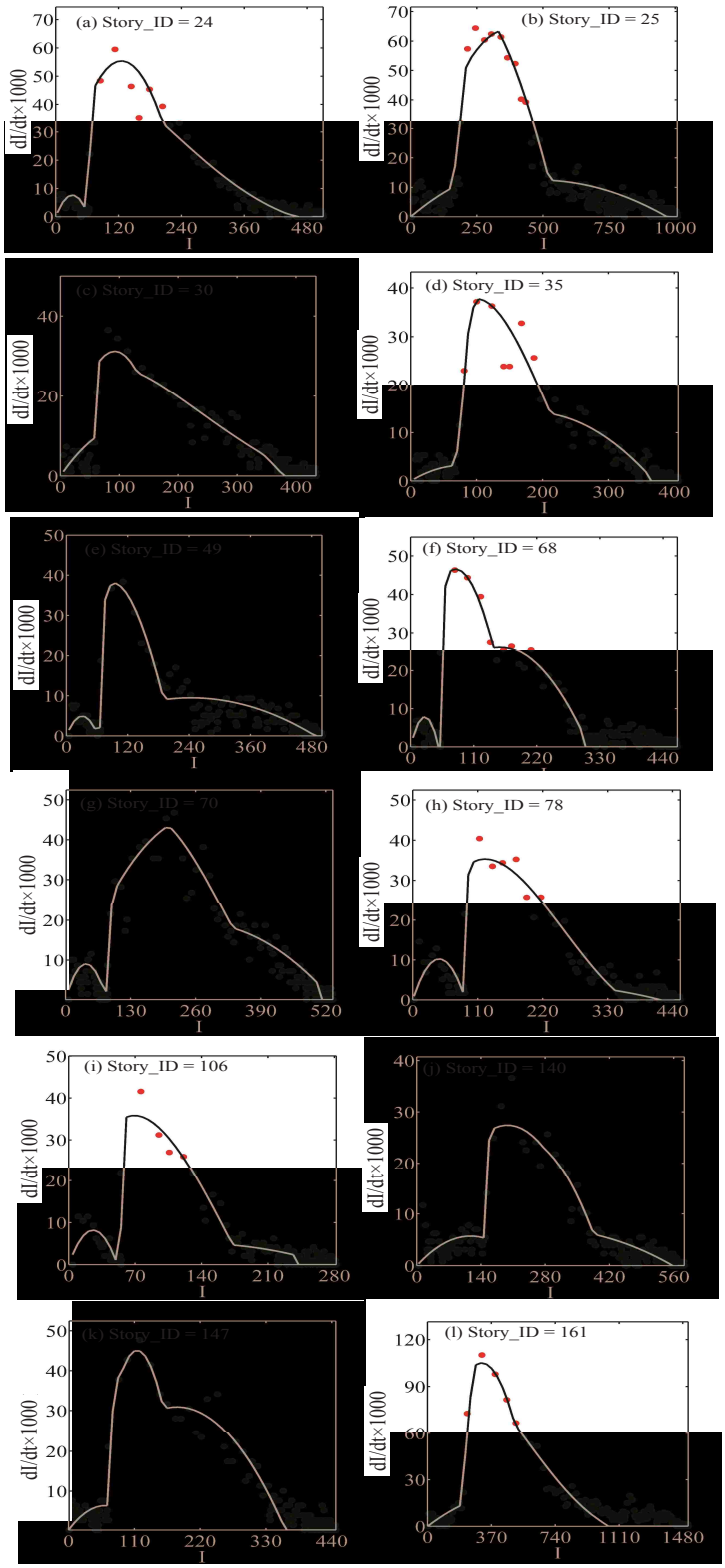

Figure 4.3 The observed (red dots) and extracted dI/dt versus I for selected stories in Digg data set.

spreading in a social platform setting. This dataset has 3553 news story for June 2009. The number of users were 139,409 who casted a total number of 3,018,197 votes.

4.2.2 Results

The proposed Diffusive Logistic model with variable carrying capacity has been applied to 200 stories in Digg data set. Figure 4.3 shows the observed $dI(t)/dt$ versus $I(t)$ (red dots) for some selected stories. $dI(t)$ is calculated using central differences based on finite differences approach.

$$\frac{dI(t)}{dt} \approx \frac{I(t+\triangle t) - I(t-\triangle t)}{2 \triangle t} \tag{4.4}$$

where $\triangle t$ is the time interval corresponding to 500 steps throughout the spreading process. The parameters of the model in previous Equations are extracted by minimizing the Mean Absolute Error (MAE) between the observed and predicted values of $dI(t)/dt$. The predicted curves are shown as the black solid lines. The minimization is done using Genetic Algorithm (GA) with random initial guess. In order to avoid local optimums, the optimization is performed 100 times for each story and the parameters with best agreement with the observations are selected.

As shown in Figure 4.3, the proposed Diffusive Logistic model with variable carrying capacities captures the temporal dynamic of information spreading in Digg data set. The initial, secondary, and final phases corresponding to K_1, K_2, and K_3 are accurately estimated using the proposed method. Table 1 shows 25, 50 and 75th percentiles of the extracted parameters. The wide

range of the parameters observed in Table 1 clearly indicates that the parameters are not unique and they change based on the story and the structure of the network around the source user.

Table 4.1 Values for influenced users and carrying capacity

Percentiles	$I_1(t)$	$I_2(t)$	$I_3(t)$	$I_4(t)$	K_1	K_2	K_3
25th percentile	56	73	111	170	70	178	291
50th percentile	70	105	165	284	122	289	407
75th percentile	115	187	316	511	341	505	668

4.3 Summary

In this chapter, we proposed a diffusion model to predict the information spreading in online social networks considering dynamic carrying capacity. Our model is able to predict the influenced users at any time by minimizing the Mean Absolute Error (MAE) between the observed and predicted values. We used Genetic Algorithm with random initial guess for the error minimization. We validated our model using real data from Digg dataset, a popular news sharing website. To the best of our knowledge, our work is the first attempt to propose dynamic carrying capacity to model and predict information diffusion in a large social platform.

CHAPTER 5: CONNECTION-BASED RATING PREDICTION

Users are overloaded with many choices when making on-line purchasing decisions, and recommender systems have become handy to alleviate this problem by providing customized recommendations. These systems offer a personalized experience based on social interactions or user preferences. In this chapter, we propose multiple methods for product rating prediction considering a recommender system with a dynamic set of users and their social connections.

5.1 Connection, Trust and Centrality-Based Rating Prediction

Users and Products: We denote the set of users present in the system at time t by $U(t)$, where $N_U(t) = |U(t)|$ is the number of users at time t. These users have the option to rate some product(s) from the set of products at any time. We denote the set of products as $P(t)$, where where $N_P(t) = |P(t)|$ is the number of products at time t. Let the rating by user i for product j at time t be given by $R_{i,j}(t)$. All such ratings at time t is given by the matrix $\mathbf{R}(t)_{N_U(t)\times N_P(t)}$. The ratings are typically integer values between a predefined minimum and a maximum value.

Trust Relationships: The social connections among users are usually given by an adjacency matrix $\mathbf{A}(t)_{N_U(t)\times N_U(t)}$ which has binary values that represent if two users are connected or not. As discussed earlier, each user trusts her connections with varying degrees. A real number $0 < T_{l,m}(t) \leq 1$ represents how much user l trusts user m at time t. If users l and m are not connected, we set $T_{l,m}(t) = 0$. Matrix $\mathbf{T}(t)_{N_U(t)\times N_U(t)}$ captures all trust relationships at time t. It is to be noted that $T_{l,m}(t)$ is not necessarily equal to $T_{m,l}(t)$.

Rating of Products: The rating of a product by an individual user depends on two factors: i) the impression on the product from the user's connections and ii) the impression from others (non-connected users). Trusted users also affect the opinion of a user towards a specific product. As users interact socially with their connections and exchange views on a product, different opinions emerge. Based on how much a user trusts a particular connection, the views on the product are regarded accordingly. As the number of ratings observed from non-connected users is usually large compared to the number of connections of a user, we tend to consider the ratings even by others even though there is no interaction. In some cases, a user may get the first impression about a product on commercial websites (e.g., Amazon, eBay, and Epinion) even before interacting with her connections.

Problem Statement: Our objective is to predict the rating user i will assign to product j at time $t+1$ (i.e., $R_{i,j}(t+1)$) given the state of the system up to time t i.e., given $\mathbf{R}(t)_{N_U(t)\times N_P(t)}$ and $\mathbf{T}(t)_{N_U(t)\times N_U(t)}$.

5.1.1 Analysis of Rating Prediction

We argue that the rating of a product by a user depends on how others have rated the product so far and how the user's connections view that product. To that end, we propose a linear combination of these two factors and use an exponentially weighted moving average to capture the temporal variations of the ratings. We also make use of the trust matrix to find how much a user is trusted by her connections and weigh her opinion accordingly.

Based on the information available at time t, we find the rating of product j by user i at time $(t+1)$ as:

$$R_{i,j}(t+1) = \lambda \times R^{C}_{i,j}(t) + (1-\lambda) \times R^{NC}_{i,j}(t) \tag{5.1}$$

where $R^{C}_{i,j}(t)$ is the weighted average of the ratings for product j by the connections of user i and $R^{NC}_{i,j}(t)$ is the average rating up to time t by the non-connections of user i who rated product j. The social factor, $0 \leq \lambda \leq 1$, weighs the impressions from the connections and non-connections. $\lambda = 0$ implies that there is no societal impact from connections and $\lambda = 1$ refers to pure social impact in which the user only follows her connections.

Effect of Connections

It is to be noted that both i) the ratings provided by connections and ii) how much one trusts her connections are functions of time. In order to consider the effect of user's connections on the rating, we must consider how one's connections have rated a product in the past and how would they rate it now. As products undergo modifications, we must put more weight on the latest

version, but at the same time should not ignore the history of the product. To that end, we propose an exponentially weighted moving average (EWMA) where we use a weight of α for the latest rating and $1-\alpha$ for all the past ratings. We get the overall rating from i's connections at time t by:

$$R^{C}_{i,j}(t) = (1-\alpha) \times R^{C}_{i,j}(t-1) + \alpha \times R^{ins}_{i,j}(t) \tag{5.2}$$

where $R^{ins}_{i,j}(t)$ is the instantaneous ratings for product j.

For calculating $R^{ins}_{i,j}(t)$, each neighbor of i is weighted individually based on their previously measured importance. It is to be noted that not all connections are trusted equally and therefore we must consider how i trusts her connections.

Centrality measures are typically used to determine one's importance and there are multiple ways of defining what importance is. If $C_l(t)$ is the centrality measure of l at time t, then $R^{ins}_{i,j}(t)$ is obtained as:

$$R^{ins}_{i,j}(t) = \frac{\sum_{l \in \mathcal{N}_i} I_{l,j}(t) \times R_{l,j}(t) \times C_l(t)}{\sum_{l \in \mathcal{N}_i} I_{l,j}(t) \times C_l(t)} \tag{5.3}$$

where $\mathcal{N}_i$ refers to the connections (neighbors) of i. We use the indicator function $I_{l,j}(t)$ as not all connections of i would rate the product j and therefore, we define this binary function as:

$$I_{l,j}(t) = \begin{cases} 1 & \text{if user } l \text{ rated product } j \\ 0 & \text{if user } l \text{ did not rate product } j \end{cases} \tag{5.4}$$

We use degree centrality and eigen-vector centrality [95] to quantify the trust of the connections of i at time t.

Degree centrality is the simplest indication of one's importance which is quantified as the *number of connections*, i.e., the number of incoming edges (in-degree). Thus, the degree centrality of l is given as:

$$C_l(t) = \sum_{\forall m, l \neq m} A_{l,m}(t) \tag{5.5}$$

Obviously, a higher in-degree means higher importance.

Eigen-Vector centrality of l is quantified as the *sum of the trust* of all connections of l which is given as:

$$C_l(t) = \sum_{\forall m} T_{l,m}(t) \times C_m(t-1) \tag{5.6}$$

The initial values for the eigevector centralities are usually set to 1 i.e., $C_l(0) = 1$ for all i which evolves over time based on Eqn. (5.6).

Effect of Non-connections

For the non-connections of i, we treat all equally to compute the average rating (i.e., $R_{i,j}^{NC}(t)$) which is given as:

$$R_{i,j}^{NC}(t) = \frac{\sum_{i=1}^{X_j} \sum_{k=1}^{t} I_{i,j}(k) \times R_{i,j}(k)}{\sum_{i=1}^{X_j} \sum_{k=1}^{t} I_{i,j}(k)} \tag{5.7}$$

where $X_j \in N$ are the ones that rated product j.

Using $R_{i,j}^{C}(t)$ from Eqn. 5.2 and $R_{i,j}^{NC}(t)$ Eqn. 5.7, we can find $R_{i,j}(t+1)$.

5.1.2 Error Metric

We would like to verify how accurate is our rating prediction model. To that end, we use the Mean Absolute Error (i.e., MAE) which is defined as the difference between the predicted rating and the actual rating and is denoted by:

$$MAE = \frac{\sum_i \sum_{j=1}^{J_i} |R_{i,j}^{pre} - R_{i,j}^{act}|}{J_i} \tag{5.8}$$

where R^{pre} and R^{act} are the predicted rating and actual rating respectively, and J_i is the set of products rated by $(i \cup \mathcal{N}_i)$.

5.1.3 Simulation Model and Results

In order to test the proposed rating prediction framework, we used Epinions dataset [122]. We used λ as an indication of social effect of connections and consider the entire range from 0 to 1. We calculate the mean absolute error with and without centrality.

Dataset Description

In the Epinions dataset, we use the item rating matrix in addition to the trust relationship matrix. In order to deal with a particular product, we use the product ID, the product category, and time-stamps of creation of the ratings. Though our method works for real trust values, this data set provides only binary values for trust; $T_{i,j} = 1$ when i is connected to j and $T_{i,j} = 0$ when i is not connected to j.

An important reason for using Epinions dataset is that is provides evolving trust relationships between users over a total of 11 time periods. In period 1, there were 155,323 trust relationships and 135,859 rating incidents, which increased to 300,545 trust relationships and 348,773 ratings by the end of the 11^{th} period. We chose a product such that all the rating incidents for that product occurring after 1^{st} period had at least 1 previous rating from the connections.

Simulation Results

Fig. 5.1 shows the impact of social factor (λ) on the mean absolute error (MAE) of the predicted ratings for *all* products. Here, $\lambda = 0$ implies no social impact from one's connections. In this case, the estimated rating is only affected by the average rating by the non-connections. On the other hand, $\lambda = 1$ refers to a pure social impact in which each user is only affected by her

connections. We show effects of connections when eigen-vector and degree centralities are used compared to when no centrality measure (i.e., all the neighbors are equally weighted) is used.

Increasing λ initially enhances the rating prediction up to a certain point, which is $\lambda = 0.35$ in this case, followed by increasing errors. In other words, besides the average rating of a product which reflects the general quality, social impact from immediate neighbors can also affect the rating. $\lambda = 0.35$ suggests that the impact from overall rating is relatively higher than the social impact. Considering the impacts of centrality measures, modeling social impacts using eigen-vector centrality leads to better performance. However increasing the social factor, reduces the positive effects of centrality measures on rating estimation.

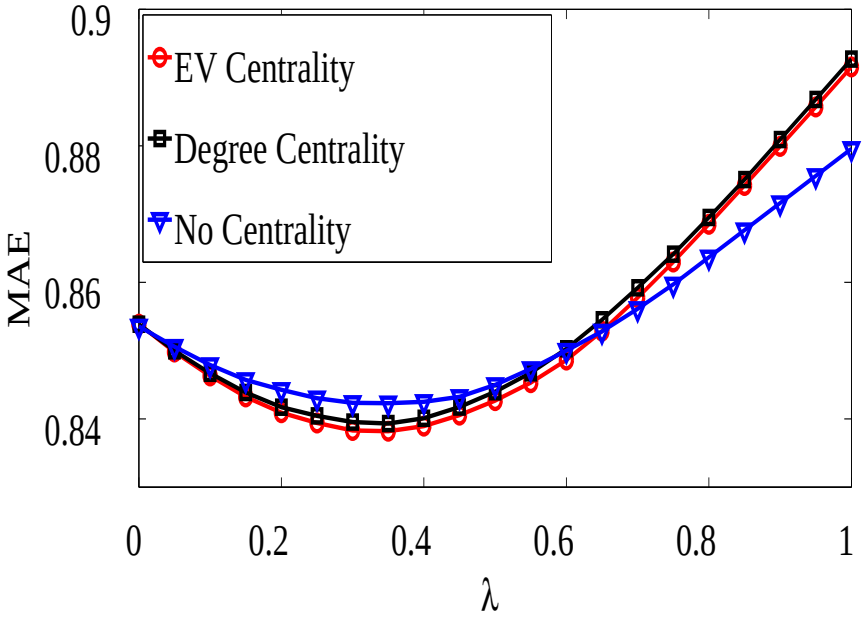

Figure 5.1 The effects of social factor (λ) and on the MAE. The least error occurs for $\lambda = 0.35$ and for eigen-vector centrality.

Fig. 5.2 shows the probability density function (pdf) of the mean absolute error using eigen-vector centrality for three different values of the social impact factor. The majority of estimated ratings (almost 70%) contain error of less than 1 when eigen-vector centrality is used. The probability of MAE decreases sharply for higher values of error.

The pdf of the mean error (ME) considering positive and negative values are shown in Fig. 5.3 for three values of λ. The right-skewness of the pdf indicates that the ratings have been overestimated.

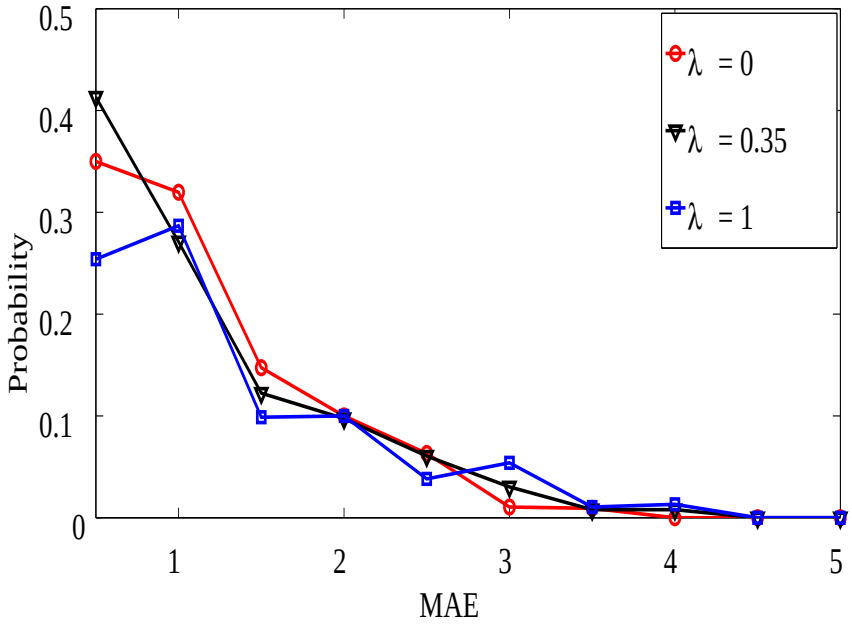

Figure 5.2 The probability distribution of MAE using eigen-vector centrality for $\lambda = 0$ (i.e., no social impact), $\lambda = 0.35$ (i.e., optimal social impact), and $\lambda = 1$ (i.e., pure social impact).

In Fig 5.4, the MAE is plotted as the function of actual rating. Interestingly, for smaller values of actual rating, the estimated ratings have relatively high errors. For instance, when actual rating is 1, the error is almost 2, which is higher than the error for actual rating equal of 5 which has

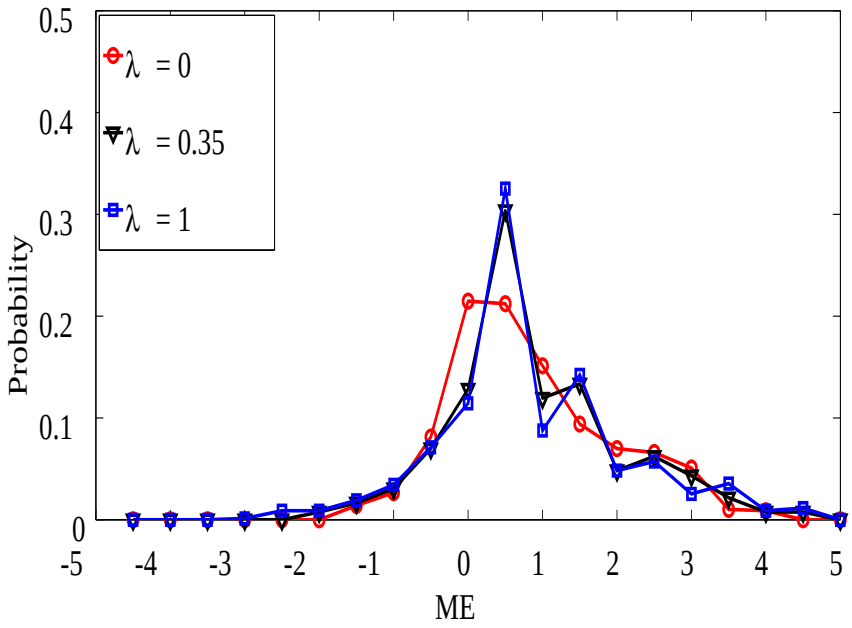

Figure 5.3 The probability distribution of mean error (ME) for *all* ratings using eigen-vector centrality

an error of 0.5. This trend suggests that, for small ratings (i.e., less than 2) the rating mechanism is different from the mechanism governing higher rating values. We believe that small actual ratings are impacted more by biased opinions and the social connections did not play a crucial role. In such cases, we choose not to buy the product and as a result there would be less number of ratings. The overestimated ratings shown in Fig. 5.3 can be attributed by this fact.

In order to further analyze the skewness, we remove the low ratings (i.e., lower than 2) and use ratings that are more than 2. In Fig. 5.5, we show the pdf of mean error (i.e., ME) for products with higher ratings. Comparing Figs. 5.3 and 5.5, we observe that the pdf is more symmetric as expected, for mean error.

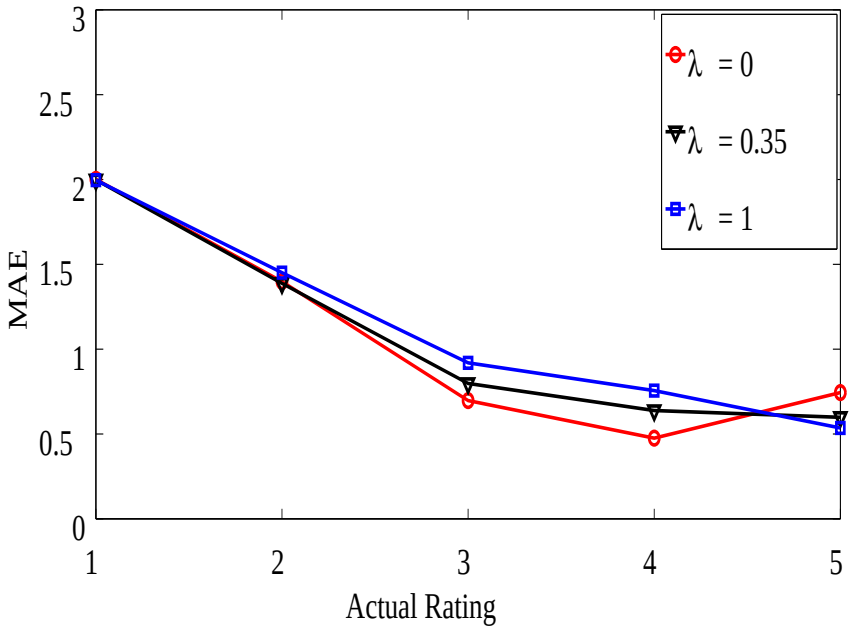

Figure 5.4 MAE for actual ratings using eigen-vector centrality

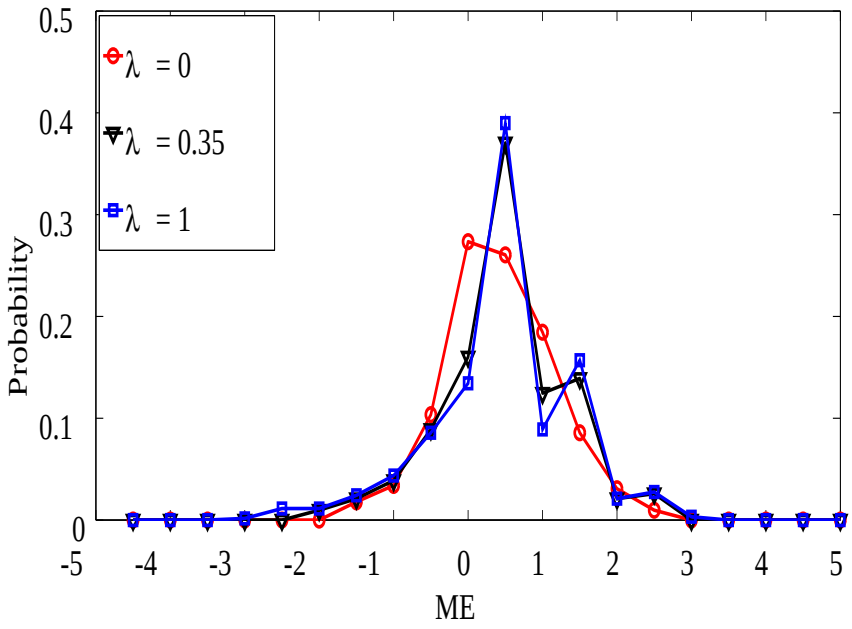

Figure 5.5 The probability distribution of mean error (ME) for *high* ratings using eigen-vector centrality

5.2 Summary

Recommender systems do not always consider the role of social interactions and the fact that users tend to trust the views of their connections more than non-connections. In this chapter, we studied how the rating of a product can be predicted using the user-item matrix and the trust relationship matrix. Using eigen-vector centrality, we modeled the trustworthiness of the connections one has. We proposed a framework to predict how a user would rate a product based on how her connections and non-connections rated that product which are linearly combined using the social impact factor. We updated the predicted rating using an exponentially weighted moving average. For evaluating the prediction accuracy of our framework, we used the mean absolute error. To validate, we used the Epinions dataset that supports the hypothesis that using centrality measures to quantify the importance of users improves the performance of rating estimation. We found that a social impact factor of 0.35 leads to the best prediction accuracy. We also found the probability density functions for the absolute error and mean absolute error.

CHAPTER 6: SIMILARITY AND CENTRALITY-BASED RATING PREDICTION

We consider a social recommender system for a social network that is represented as a weighted directed graph of users where edges represent the social trust relationship between users. The existence of a social connection between two users would not necessarily reflect their level of trust in each other. The method presented here is based on the assumption that the trust between users is impacted by similarity and importance of users. Our objective is that in a given recommender system, how can we predict the rating that user i would assign to product j, when the social relationship graph and the user-item rating matrix are given.

6.1 Proposed Social Trust Model

We model a social recommender system as a social network represented as a weighted directed graph with M users. In this social network, edges represent the social trust relationship between users. The users rate their items of interests on a scale of 1 to 5. The social relationships (connections) between users are built into the adjacency matrix $A_{M\times M}$. The rating assigned by each user to each item is represented by the user-item rating matrix $R_{M\times N}$, where N represents number of items (products).

6.1.1 Similarity-based Trust

A critical part of collaborative filtering is to compute similarities among users by building a user-item rating matrix. However, collaborative filtering methods suffer from various issues such as data sparsity and cold start users. To address this issue, some studies have incorporated user similarity in trust models. In [70], user similarity and weighted trust propagation are used to reconstruct trust matrix which helps with the cold start problem. In [38], the authors proposed an algorithm for trust score which combines the number of items with the similarity score between users, and build a trust relationship matrix. Another study [137], proposed a trust model which is based on using propagated trust and similarity of users rating habits. A novel algorithm based on the trust model combined with the user similarity factor has been proposed in [132]. Our method assumed that the trust between users is impacted by similarity between two users and importance of each user. Similarity between users is one of the most important factors that affect the value of trust between users since two users with the same taste are more likely to trust each other. Here we apply both rating-based and connection-based methods to capture the similarity between two users.

6.1.1.1 Rating Similarity

We apply similarity algorithms to identify the similarity between users. The VSS algorithm utilizes the common items that have been rated by both users i and f to compute similarity which is given

by:

$$Sim(i,f) = \frac{\sum_{j \in I(i) \cap I(f)} R_{i,j} \cdot R_{f,j}}{\sqrt{\sum_{j \in I(i) \cap I(f)} R_{i,j}^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} R_{f,j}^2}} \tag{6.1}$$

where j is an item that both users i and f have rated and $R_{i,j}$ is the rating that user i assigned to item j. $I(i)$ represents the set of items rated by user i. VSS is defined in $[0, 1]$; a larger value implies more similarity between user i and user f.

The trust values enforced by similarity can be modeled by weighted average rating of the users using the similarity scores as the weights. Consequently, a connection with high similarity will have more impact on the user's rating. When calculating the VSS value, the difference in user's rating style is not considered (e.g., always high rating or always low rating). The PCC method can obtain better performance than the VSS approach, since the PCC method considers the differences of user ratings. So we apply the PCC algorithm to identify the similarity between users. The similarity between users that have been rated by both users i and f is given by:

$$Sim(i,f) = \frac{\sum_{j \in I(i) \cap I(f)} (R_{i,j} - \overline{R_i}) \cdot (R_{f,j} - \overline{R_f})}{\sqrt{\sum_{j \in I(i) \cap I(f)} (R_{i,j} - \overline{R_i})^2} \cdot \sqrt{\sum_{j \in I(i) \cap I(f)} (R_{f,j} - \overline{R_f})^2}} \tag{6.2}$$

where $\overline{R_i}$ is the average rating of user i. We use the mapping function, $f(x) = (x + 1)/2$, to map PCC values to $[0, 1]$. It is important to note that the value of similarity could be negative and its magnitude signifies the dissimilarity degree.

6.1.1.2 Connection Similarity

There are some drawbacks with using the rating-based similarity methods. These methods (VSS and PCC) are rating-based, so they would not be applicable if two users have not mutually rated the same product. Also these similarity measures are restricted to symmetric ones such that the similarity between users u and v are the same for v and u, although the symmetry may not hold in many real world applications specifically in a social network modeled by a directed graph.

The similarity between two users can be measured by the connections they have in common. This can be done using each user's list of connections. A larger value is an indication of the users having more similarity which shows that their connection is more valid in shaping the trust [33]. The list of friends for each user i is defined as $F(i)$. The proportion of mutual friends to the total number of friends is defined as follows:

$$Sim(i, f) = \frac{F(i) \cap F(f)}{F(i)} \tag{6.3}$$

6.1.2 Centrality-based Trust

A user with high importance (i.e., high impact) is more likely to be followed by her friends regardless of their similarities. This aspect of trust relationship is modeled by considering the importance of users which can be quantified using centrality measures. To obtain the importance of users, we use degree centrality, eigen-vector centrality, Katz centrality and PageRank [95]. We choose these

centrality measures since they consider the connections and also the importance of each connection by adding the free initial centrality to deal with special cases.

Degree centrality is used as the basic indication of a user's importance which can be defined as the *number of connections*. In our case, it is the number of incoming edges (in-degree) in the social graph. Recall from chapter 5, we define the degree centrality C_l of a user l as:

$$C_l = \sum_{\forall m, l \neq m} A_{l,m} \tag{6.4}$$

where $A_{l,m}$ is the element of the adjacency matrix which represents the connection between user l and user m. Thus, with all connections treated equally, a user with more incoming edges has higher importance in the network.

Eigen-vector centrality gives each node a value which is proportional to the sum of values of its neighbors. Eigen-vector centrality has a property: it can be large either because a node has many neighbors or because it has important neighbors (or both). Recall from chapter 5, eigen-vector centrality of user l at time t is the defined as *sum of the centrality* of all connections of user l which is given as:

$$C_l(t) = \sum_{\forall m} A_{l,m}(t) \times C_l(t-1) \tag{6.5}$$

where $C_l(t-1)$ is the centrality of user l at time $t-1$. In contrast to the degree centrality, the eigen-vector centrality considers both the number of incoming edges and also the centrality of the neighboring users. The eigen-vector centrality is computed iteratively by setting all initial values to 1 i.e., $C_l(0) = 1$ for all user l.

Katz centrality is similar to eigen-vector centrality except that it adds a free centrality value to each node. In this centrality, we consider a value which is called free centrality. We add the free centrality to account for users that do not have any outgoing edges. The Katz centrality of user l at time t is defined as:

$$C_l(t) = \alpha \times \sum_{\forall m} A_{l,m}(t) \times C_l(t-1) + \epsilon \tag{6.6}$$

where ϵ is the free centrality value. By adding this second term, even nodes with zero in-degree still get centrality ϵ, and once they have a non-zero centrality, then the nodes they point to derive some advantage from being pointed to. This means that any node that is pointed to by many others will have a high centrality, although those that are pointed to by others with high centrality themselves will still do better.

PageRank centrality A problem with with Katz centrality is that if a node with high Katz centrality points to many others then those others also get high centrality. The centrality gained by virtue of receiving an edge from a prestigious node is diluted by being shared with so many others. The PageRank centrality fixes this by defining a variation of the Katz centrality in which the centrality a node derives from others is proportional to their centrality divided by their out-degrees ($k_{out} \neq 0$). Nodes that point to many others pass only a small amount of centrality to each of those others, even if their own centrality is high. In mathematical terms, we define this centrality by:

$$C_l(t) = \alpha \times \sum_{\forall m} A_{l,m}(t) \times \frac{C_l(t-1)}{k_{out}(t-1)} + \epsilon \tag{6.7}$$

6.1.3 Linear Social Trust Ensemble

To model the social trust between users in a social recommender system, we use a linear combination of similarity and centrality to represent the trust of user i on user k as [34, 35]:

$$\Gamma_{i,k} = \beta \frac{Sim(i,k)}{\sum_{l \in \mathcal{T}(i)} Sim(i,l)} + (1-\beta) \frac{C_k}{\sum_{l \in \mathcal{T}(i)} C_l} \tag{6.8}$$

Here, β is the parameter that defines the contribution of similarity and centrality to the overall trust. $\beta = 0$ implies purely centrality enforced trust while $\beta = 1$ refers pure similarity-based trust values. $\mathcal{T}(i)$ refers to the set of trusted friends of user i. C_k refers to the centrality (i.e., measured using either degree or eigen-vector centrality) of user k.

6.2 Social Trust Model using Matrix Factorization

Matrix factorization has been widely used to develop social recommender systems as it helps to estimate either the user-item rating or user-trust matrix [74] using low-dimensional representative latent matrices. Here, matrix factorization for social recommendation proposed in [77] is employed to examine the performance of the proposed trust relationship.

The user-item rating matrix is factorized to learn two $l-$dimensional feature representation of users U and items V matrices. The user-item rating matrix R consists of M users and N items with rating values in the range $[0, 1]$. U_i and V_j represent the $l-$dimensional user-specific and

item-specific latent feature vectors of user i and item j. A low-rank matrix factorization approach seeks to approximate the matrix R by multiplication of $l-$dimensional factor $R \approx U^T V$, where $U \in R^{l \times M}$ and $V \in R^{l \times N}$ with $l \leq min(M, N)$. In real datasets, matrix R is usually very sparse.

The conditional distribution for R, given Γ, U, V and σ_R^2 is defined as [77]:

$$p(R|\Gamma, U, V, \sigma_R^2) = \prod_{i=1}^{M} \prod_{j=1}^{N} [\mathcal{N}(R_{ij}|g(\sum_{k \in \mathcal{T}(i)} \Gamma_{ik} U_k^T V_j), \sigma_\Gamma^2)]^{I_{ij}^R} \tag{6.9}$$

where $\mathcal{N}(R_{i,j}|\mu, \sigma_\Gamma^2)$ is probability density function of the Gaussian distribution with mean μ and variance σ_Γ^2. Here, Γ is the proposed trust parameter given by Eq. (6.4), $\Gamma_{i,k}$ is the trust value between users i and k. $R_{i,j}$ is the rating given to item j by user i, and σ_R^2 is the rating variance. I_{ij}^R is an indicator function representing whether user i rated item j. Based on the Bayesian inference and assuming Γ is independent of U and V, the conditional probability of U and V, given R, Γ, σ_R^2 , σ_U^2, and σ_V^2, is defined as:

$$p(U, V|R, \Gamma, \sigma_\Gamma^2, \sigma_U^2, \sigma_V^2) = \prod_{i=1}^{M} \prod_{j=1}^{N} [\mathcal{N}(R_{i,j}|g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j), \sigma_\Gamma^2)]^{I_{i,j}^R} \times \prod_{i=1}^{M} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \times \prod_{i=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}) \tag{6.10}$$

where σ_U^2 and σ_V^2 are the variances of user and item feature matrices. $\mathbf{I}$ is the identity matrix. The function $g(x) = 1/(1 + exp(-x))$ is a mapping function whose range is within $[0, 1]$. The set $\mathcal{T}(i)$ contains user i's trusted friends.

The proposed social recommender system is based on the idea that user's ratings are impacted by her own taste and her immediate friends' tastes. The parameter α is used to balance between these two factors. The term $U_i^T V_j$ represents the estimated taste of user i of item j, while $\sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j$ term reflects her immediate friends' taste, given as the weighted average of their taste using the trust value as weights.

6.2.1 User-Specific and Item-Specific Matrices

In this section, we seek to find the U and V matrices. The log of posterior distribution for the recommendation is given by:

$$\ln p(U, V | R, \Gamma, \sigma_\Gamma^2, \sigma_U^2, \sigma_V^2) = -\frac{1}{2\sigma_\Gamma^2} \sum_{i=1}^{M} \sum_{j=1}^{N} I_{i,j}^R (R_{i,j} - g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j))^2$$
$$- \frac{1}{2\sigma_U^2} \sum_{i=1}^{M} U_i^T U_i - \frac{1}{2\sigma_V^2} \sum_{j=1}^{N} V_j^T V_j - \frac{1}{2} (\sum_{i=1}^{M} \sum_{j=1}^{N} I_{i,j}^R) ln\Gamma^2 - \frac{1}{2} (Ml \ln \sigma_U^2 + Nl \ln \sigma_V^2) + \mathcal{C} \tag{6.11}$$

Here $\mathcal{C}$ is a constant independent of other parameters. Maximizing the log-posterior over the two latent features is equivalent to minimizing the sum-of-squared-errors objective functions with quadratic regularization terms to derive U and V:

$$\mathcal{L}(R,\Gamma,U,V) = \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N} I_{i,j}^{R}(R_{i,j} - g(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j))^2 + \frac{\lambda_U}{2}||U||_F^2 + \frac{\lambda_V}{2}||V||_F^2 \tag{6.12}$$

Here $\lambda_U = \frac{\sigma^2}{\sigma_U^2}$, $\lambda_V = \frac{\sigma^2}{\sigma_V^2}$ and $||.||_F^2$ is the Frobenius norm. λ_U and λ_V are user and item latent variance ratios.

The gradient decent approach can be used to solve the minimization problem given in Eq. (6.11) for finding U and V. Gradient decent is a local optimization method based on the partial derivative of the objective function with respect to the decision variables (i.e., U and V). The partial derivatives of $\mathcal{L}$ with respect to U and V are given in Eqs. (6.12) and (6.13).

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial U_i} = \alpha \sum_{j=1}^{N} I_{i,j}^{R} g'(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j) V_j \times (g(\alpha U_i^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j - R_{i,j}) \\ + (1-\alpha)\sum_{p\in\phi(i)}\sum_{j=1}^{N} I_{p,j}^{R} g'(\alpha U_p^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(p)} \Gamma_{p,k} U_k^T V_j) \times (g(\alpha U_p^T V_j + (1-\alpha)\sum_{k\in\mathcal{T}(p)} \Gamma_{p,k} U_k^T V_j) \\ - R_{p,j})\Gamma_{p,i} V_j + \lambda_U U_i \end{aligned} \tag{6.13}$$

$$\frac{\partial \mathcal{L}}{\partial V_j} = \sum_{i=1}^{M} I_{i,j}^{R} g'(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j) \times (g(\alpha U_i^T V_j + (1-\alpha) \sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T V_j - R_{i,j}) \times (\alpha U_i + (1-\alpha) \sum_{k \in \mathcal{T}(i)} \Gamma_{i,k} U_k^T) + \lambda_V V_j \quad (6.14)$$

Here $g'(x)$ is the derivative of logistic function where $g'(x) = exp(x)/(1 + exp(x))^2$. $\phi(i)$ is the set of the users who trust user i [75].

6.3 Accuracy Measures

In order to test the validity and accuracy of the proposed rate prediction framework, we conduct extensive simulation experiments with data from Epinions [122].

6.3.1 Data Source

We base our experimental analysis on a dataset based on trust-based product review website Epinions.com which is a product comparison website that features products reviews with a social component. It allows users to post reviews about products with a rating from 1 to 5. It also allows users to create directional social links that can be defined as trust and distrust links towards other users. Since the distrust links are not publicly available, we study only the trust links. Also users can provide feedback about the quality of product reviews written by other users. Each review

has a helpfulness score summarized as very helpful, somewhat helpful, helpful, not helpful, or no feedback. The Epinions website takes into account the trust links in order to make personalized recommendations.

The social connections in this dataset are binary values and do not represent the actual trust values. The dataset includes 22166 users and 355754 social connections, leading to 0.0724 percent density in the user social relationship matrix. The total number of items is 296277, with a total of 922267 ratings, which results in a very sparse item-rating matrix with 0.0140 percent density. As a result, the user-item rating matrix is also relatively sparse. On average, users have 16.05 trusted friends. The maximum number of friends for a user is 1551 and the most trusted user has 2023 other users trusting her.

6.3.2 Accuracy Metric

Evaluation measures for recommender systems are divided into three classes of prediction accuracy metrics: i) Predictive accuracy measures (such as MAE, RMSE), evaluate how close the recommender system is in predicting actual rating values, ii) Classification accuracy measures (such as Precision, Recall, F1) which measure the frequency with which a recommender system makes correct/incorrect decisions regarding items based on the relevancy of the recommended items, and iii) Rank accuracy measures (such as Discounted cumulative gain (DCG) and Mean Average Precision (MAP)) which evaluate the correctness of the ordering of items performed by the recommendation system. Since our proposed model focus on the error in the rating prediction, we use the metrics

in the first category which evaluate the prediction accuracy of the recommender system. The other two categories are typically used for classification and ranking, are therefore not considered.

6.3.3 Predictive Accuracy Measures

Different types of error metrics are defined as follows.

Mean Absolute Error (MAE): This metric measures the average variation in the predicted rating vs. the actual rating. Let $R_{i,j}^{pre}$ be the predicted rating and $R_{i,j}^{act}$ be the actual rating given by the user i to the product j. Recall from chapter 5, the MAE is defined as follows:

$$MAE = \frac{\sum_{i,j} |R_{i,j}^{pre} - R_{i,j}^{act}|}{M} \tag{6.15}$$

Root Mean Squared Error (RMSE): This metric is the most popular metric used in evaluating accuracy of predicted rating. It is a variant of mean square error and is defined as follows:

$$RMSE = \sqrt{\frac{\sum_{i,j} |R_{i,j}^{pre} - R_{i,j}^{act}|^2}{M}} \tag{6.16}$$

All these metrics measure the accuracy of the actual predictions and are easy to compute efficiently. Moreover, MAE and MAE-based error estimates have well known statistical properties. These characteristics makes MAE and RMSE good representative of error metrics to analyze the accuracy of the proposed model.

6.4 Results and Discussions

We present how our trust models perform with the data obtained from Epinions. Based on the proposed model, the trust relationships between users are built on the two components of centrality and similarity measures. We demonstrate the probability density function of centrality, normalized similarity, and trust. These distributions reveal what and how much impact each of these parameters have for various values of the parameter in question.

6.4.1 Distribution Analysis

Fig. 6.1 shows the distribution of different centrality measures that have been analyzed in our model: degree, eigen-vector, Katz, and PageRank centrality.

In Fig. 6.2, the distribution of rating-based (i.e., VSS and PCC) and connection-based similarity are shown. VSS and PCC have a relatively sparse distribution due to the lack of mutually rated products by two friends in many cases. The trust values are calculated as the weighted summation of centrality of similarity using the weight constant β.

Fig. 6.3, Fig. 6.4, and Fig. 6.5 show the distribution of trust values for different types of similarity being applied; PCC, VSS, and connection-based similarity. These figures show the distribution of trust values using $\beta = 0.5$. The proposed trust model is used to predict users' rating based on the discussed matrix factorization technique using 75 percent of the data as the training set. According to Eq. (6.5), a user's opinion about a particular product would be a linear function

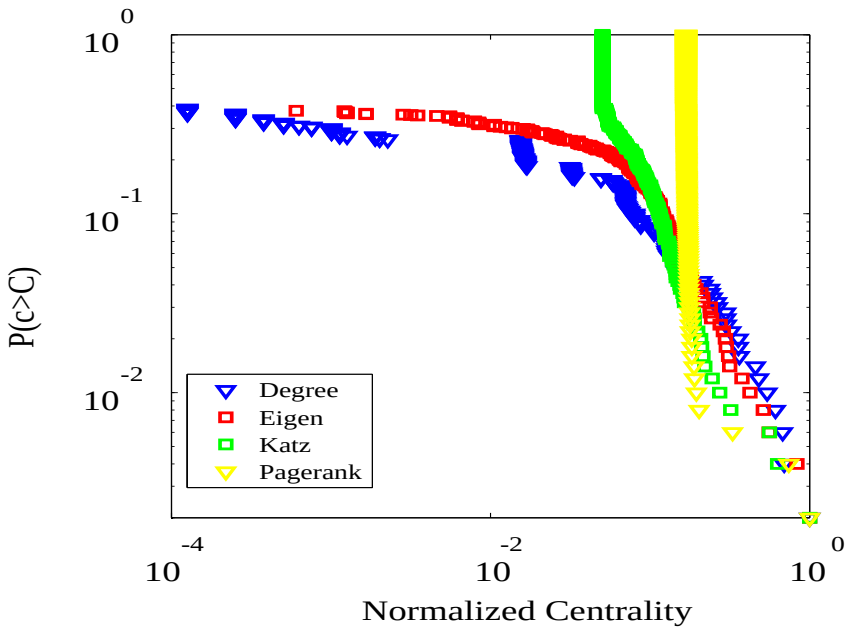

Figure 6.1 Distribution of centralities

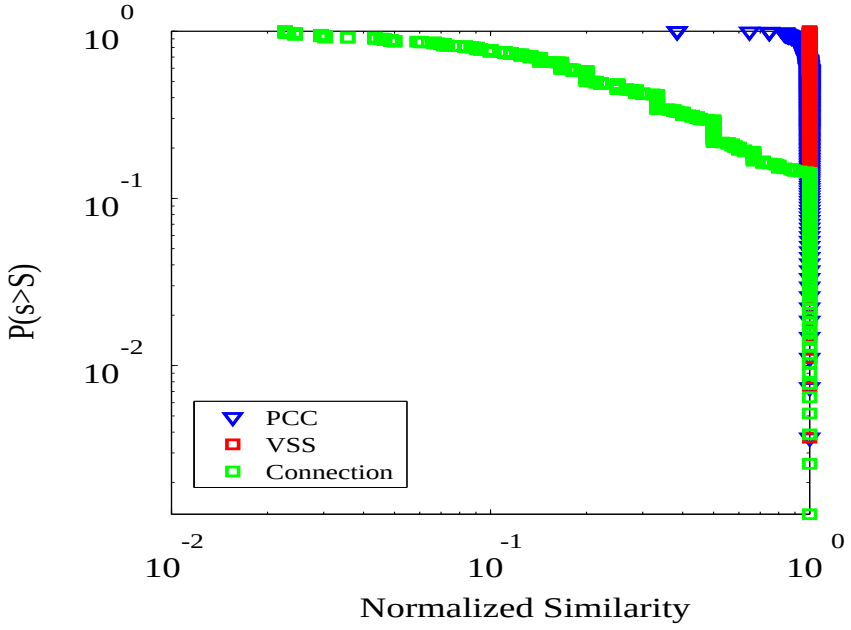

Figure 6.2 Distribution of similarity

of her connections' taste and her own taste using a weighting factor α. Smaller values of α is an indication of less impact from neighbors. As previously defined in Eq. (6.4), the trust model is presented as the linear combination of centrality and similarity using the weighting factor β. Higher values of β indicate higher impact of similarity rather than centrality on the trust values. Here, user and item latent variance ratio (λ_U and λ_V) are set to 0.001. The latent size is $L = 4$, $\alpha = 0.4$, and the number of iterations is 300.

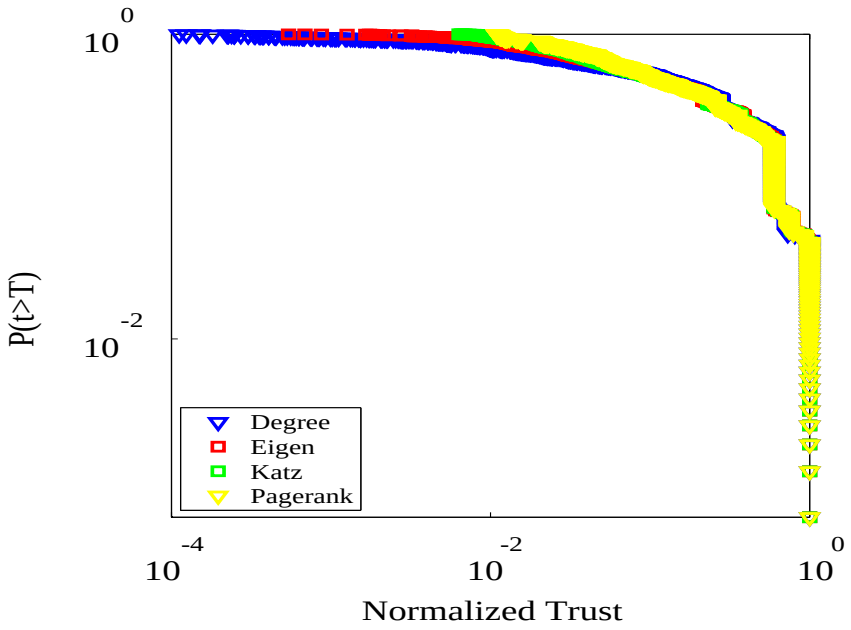

Figure 6.3 Distribution of trust values for PCC similarity

6.4.2 Performance Analysis

The performance of the proposed trust model for different values of β in terms of MAE is shown in Fig. 6.6 for PCC similarity, Fig. 6.7 for VSS similarity, and Fig. 6.8 for connection-based similarity.

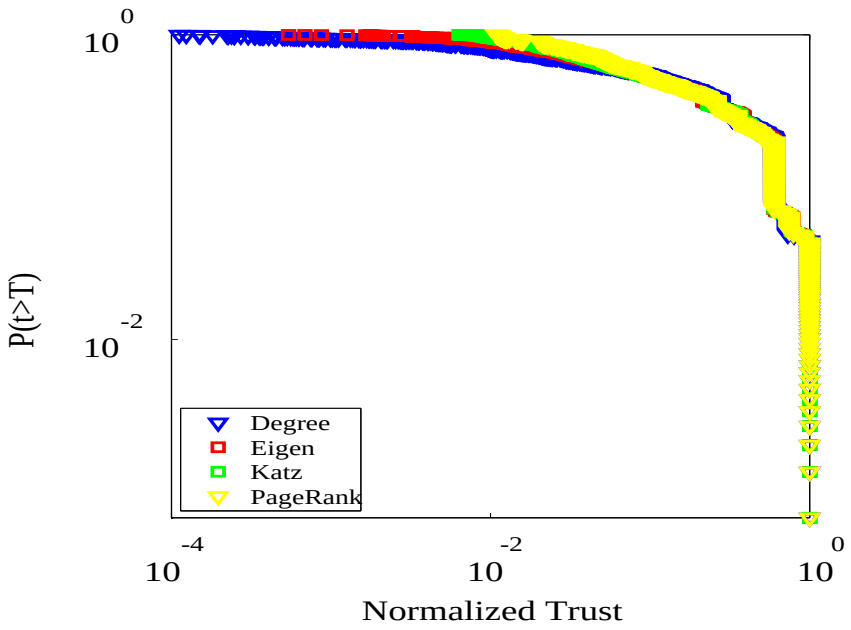

Figure 6.4 Distribution of trust values for VSS similarity

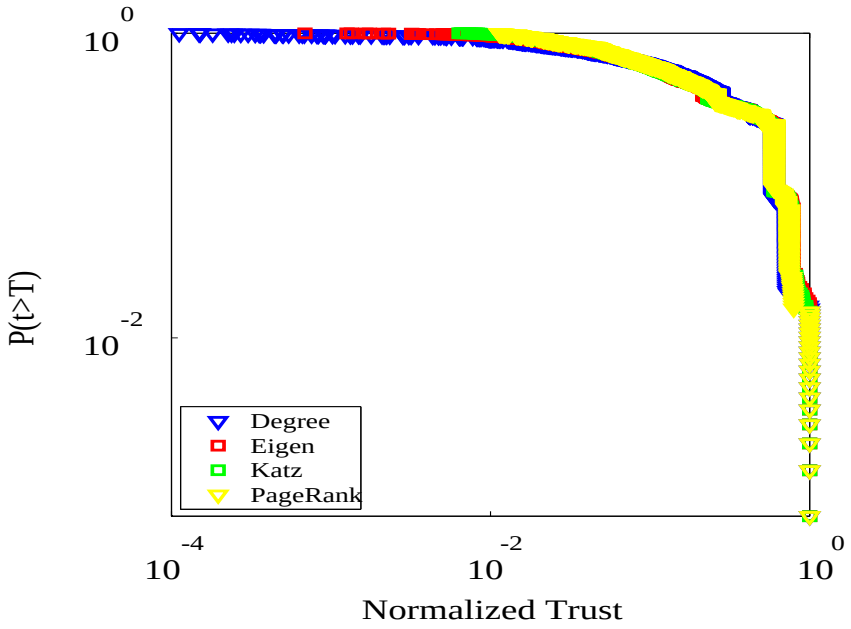

Figure 6.5 Distribution of trust values for connection similarity

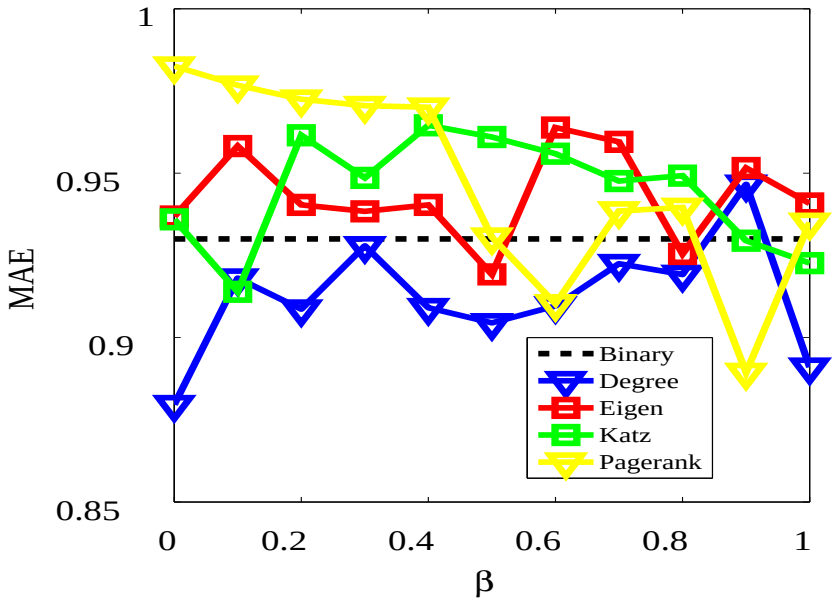

Figure 6.6 MAE using binary trust and the proposed trust model for PCC similarity

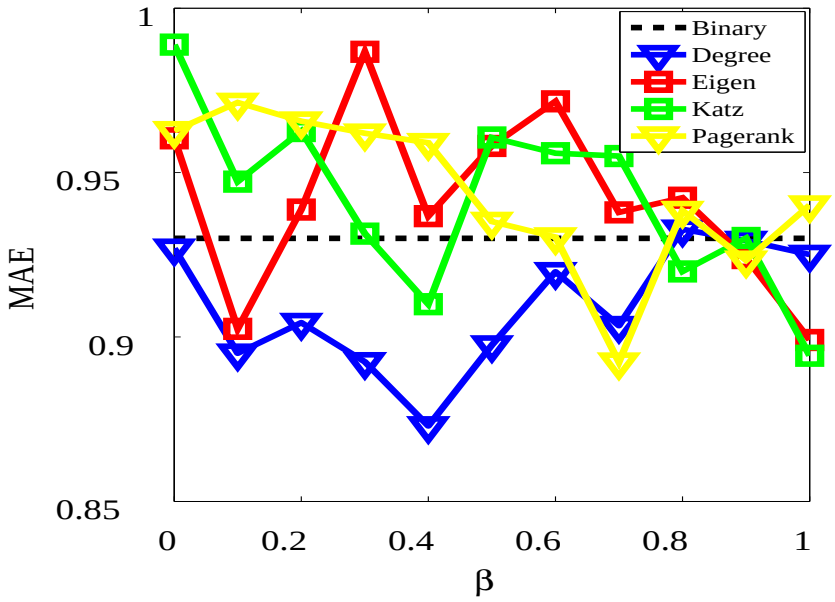

Figure 6.7 MAE using binary trust and the proposed trust model for VSS similarity

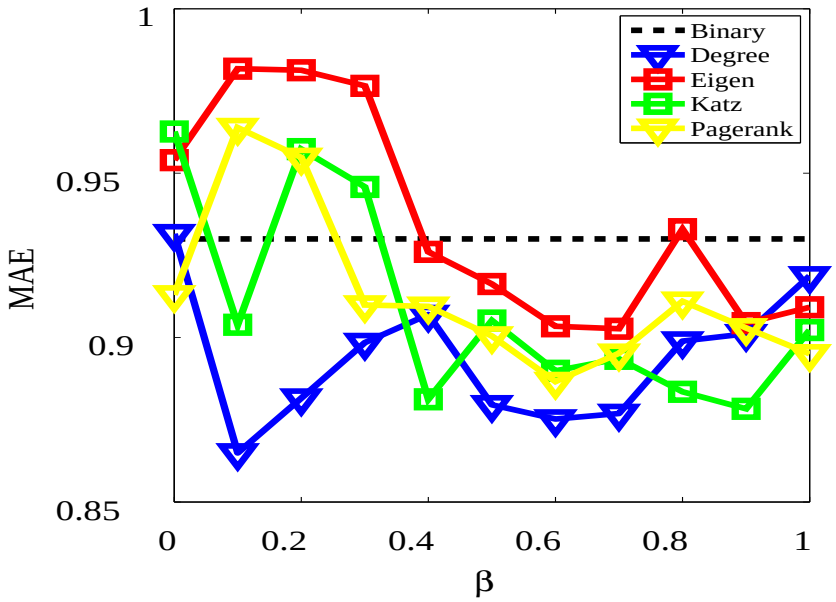

Figure 6.8 MAE using binary trust and the proposed trust model for connection similarity

The same is shown for RMSE for the performance of the proposed trust model for different values of β in Figs. 6.9, 6.10, and 6.11.

Compared to the binary trust model (dashed black lines), the proposed trust model has better performance. Comparing different definitions of trust reveals that degree centrality is the better measure to model trust compared to using other centrality measures. The same is true for connection-based similarity compared to rating-based. An interesting observation is that, although including centrality in trust model enhances the recommendation performance compared to the binary trust model, the trust models solely based on similarity (i.e., $\beta = 1$) show the best performance for the studied network.

The probability distribution of rating estimation error (i.e., estimated rating minus actual rating) for the binary trust and proposed trust model is shown in Fig. 6.14. Both probability dis-

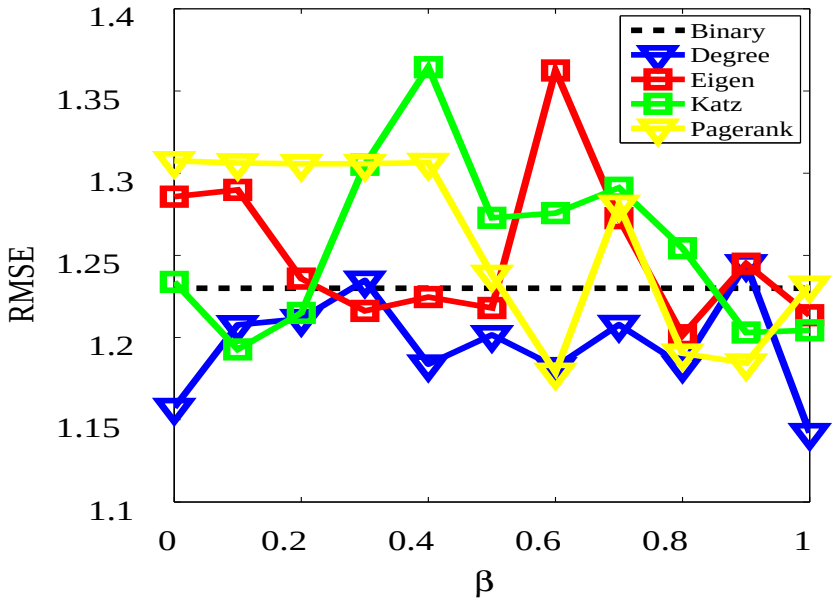

Figure 6.9 RMSE using binary trust and the proposed trust model for PCC similarity

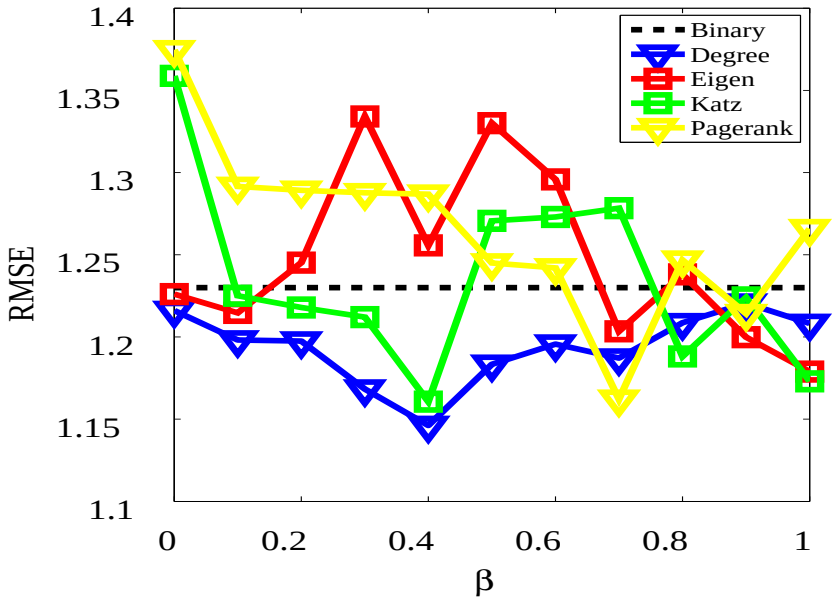

Figure 6.10 RMSE using binary trust and the proposed trust model for VSS similarity

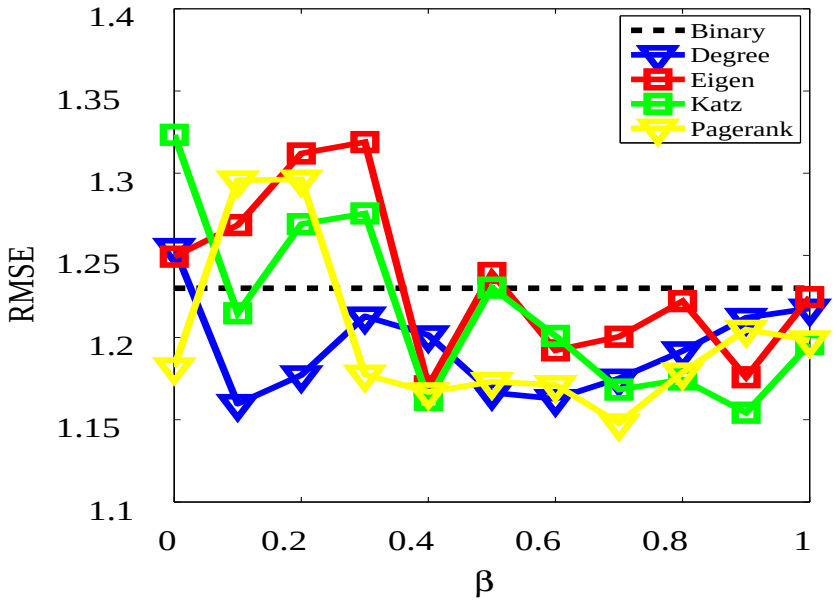

Figure 6.11 RMSE using binary trust and the proposed trust model for connection similarity

tributions are right skewed, implying over-estimation. However, the proposed trust model seems to have relatively better performance especially for errors between 0.5 and 2, since it estimates more between 0.5 and 1 and less between 1 and 2 compared to the binary model. The probability distribution of absolute error ratio (i.e., absolute error divided by the actual rating) is shown in Fig. 6.15. The proposed trust model leads to lower error ratio between 1 and 2 and more between 0 and 1 which implies relatively better performance.

6.4.3 Error Analysis

The performance of the trust model (the definition which had the best performance in Figs. 6.3, 6.4, and 6.5) for different latent sizes and training percentages are shown in Fig. 6.12 and Fig. 6.13.

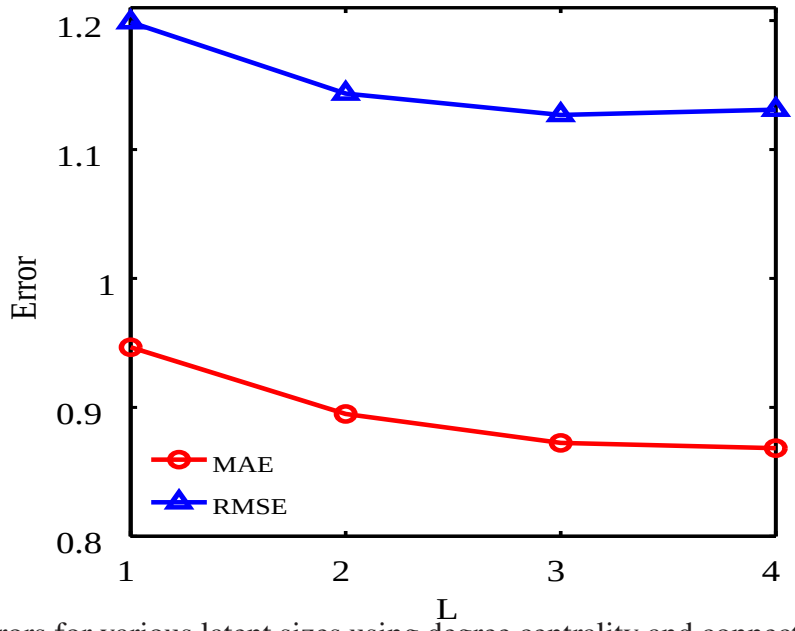

Figure 6.12 Errors for various latent sizes using degree centrality and connection-based similarity

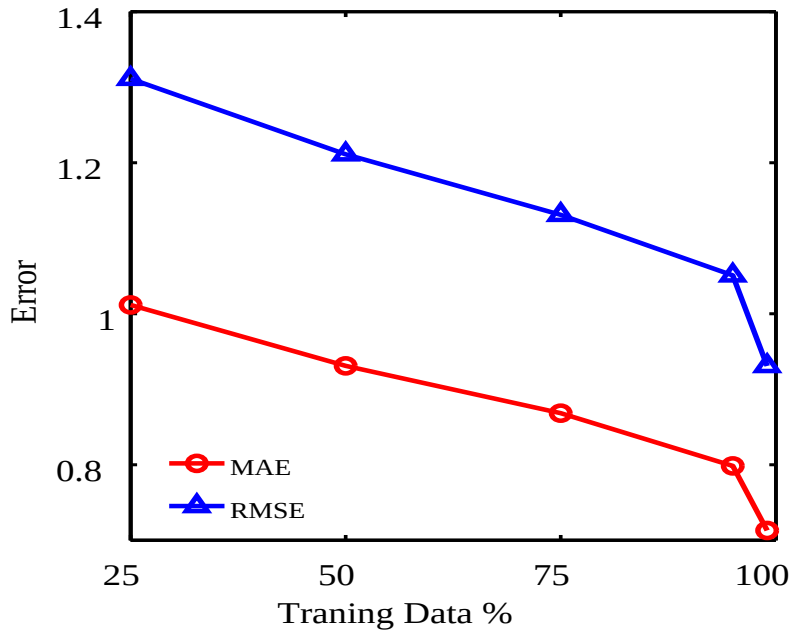

Figure 6.13 Errors for various training set sizes using degree centrality and connection-based similarity

Generally, increasing the latent size as well as using more training data enhance the performance of the recommender system.

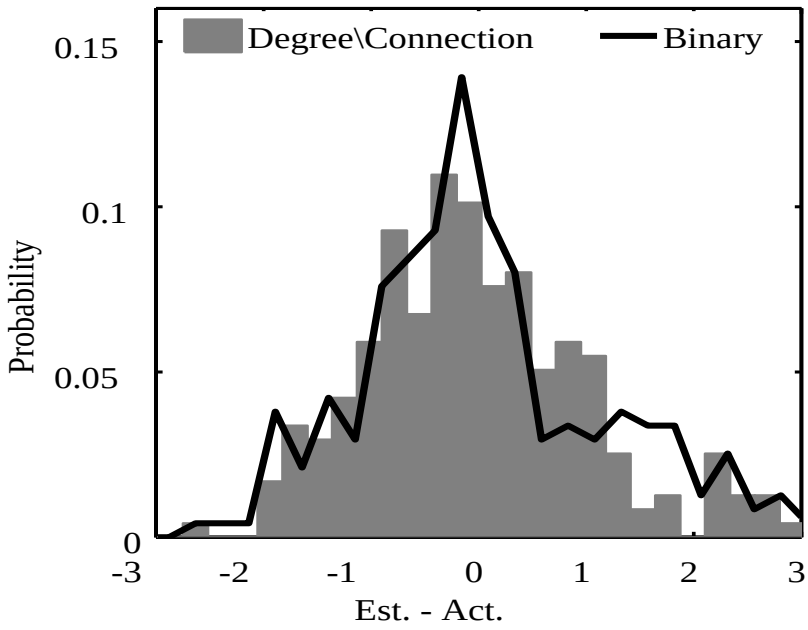

Figure 6.14 The probability distribution of error for rating estimation using binary trust and the proposed trust model

Figs. 6.16 and 6.17 show the estimated versus actual ratings for the proposed and the binary trust models. The boxes illustrate the lower, upper, and inter quartiles, while the red line is the medium. The height of the boxes represents the variation of the estimated rating. Comparing Figs. 6.16 and 6.17, it is observed that the proposed trust model produces better estimations for low ratings (1 and 2) by slightly undermining the estimation. In addition, for high ratings, the proposed trust model reduces the variation of estimations, i.e., the height of the quartile boxes.

As previously discussed, the trust-walker method use random walk; however our method uses similarity and centrality metrics. The similarity and centrality elements used to build the

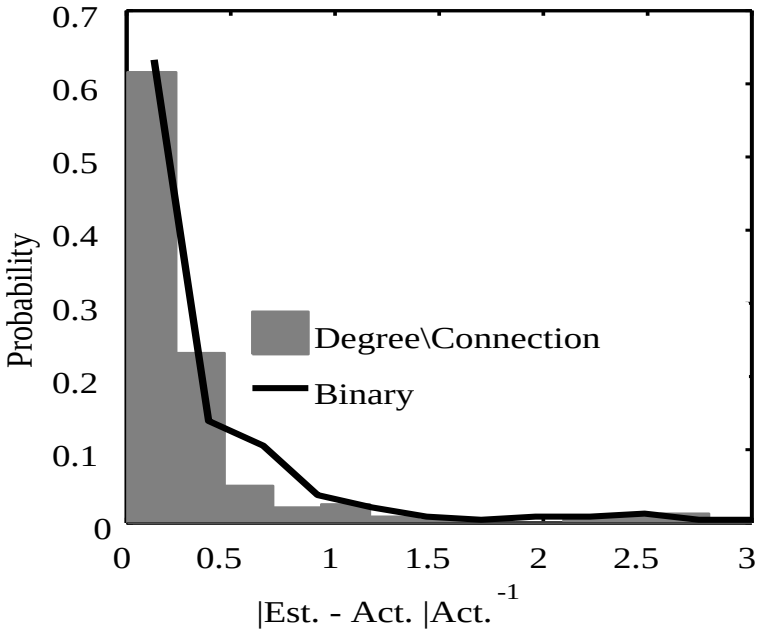

Figure 6.15 Absolute error ratio for rating estimation using binary trust and the proposed trust model

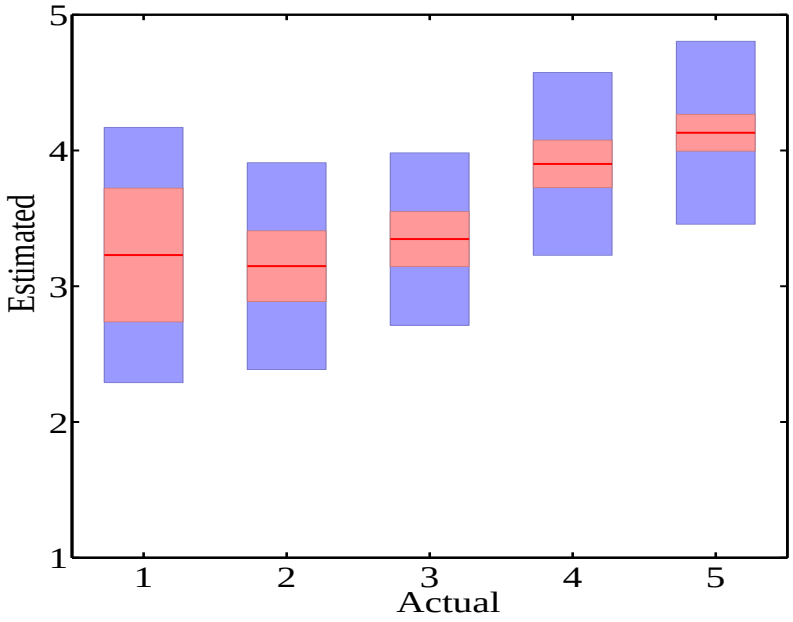

Figure 6.16 The quartile plot of actual versus estimated rating for the proposed trust model

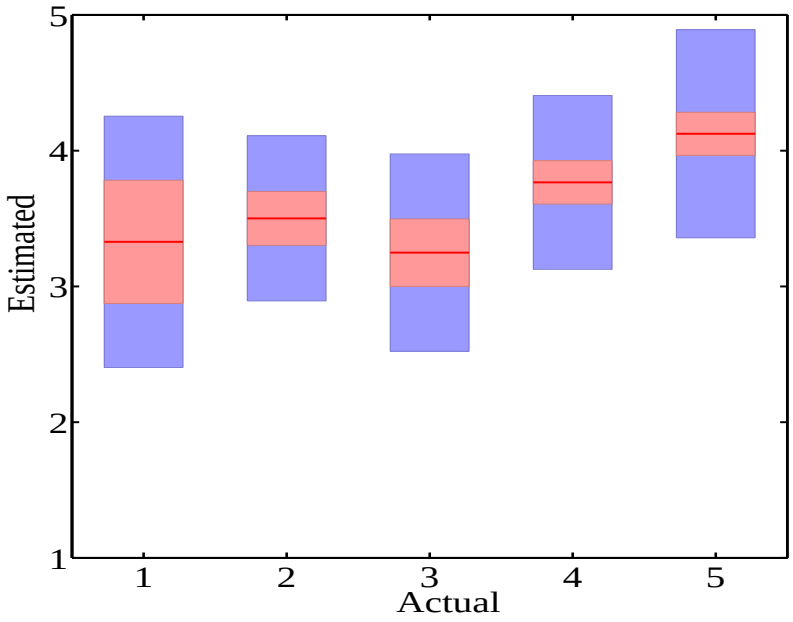

Figure 6.17 The quartile plot of actual versus estimated rating for the binary model.

trust gives individual value to each user affected by new neighbor and new place in the social trust network. Compared with TrustWalker, TidalTrust and MoleTrust, our method shows equal or lower RMSE values for the trust model. Considering centrality of the users to build the trust between users also helps with the cold start problem with users who have rated few items.

6.5 Summary

With emerging applications of social networks and considering the role of social interactions in our daily life decisions, extracting information from user's social relationships is becoming a popular method for predicting user's behavior. To consider and balance these factors, this paper proposes a social trust model that incorporates the preference similarity, user's centrality, and social relation

in order to predict the rating for the social recommender system. We capture the trust relationships between users considering users with similar profile and their importance. We argue that users with more similarity would trust each other more; also users with higher importance would be trusted more. Similarity is quantified by using rating-based approaches and a connection-based centralities. The importance of users is modeled by degree, eigen-vector centrality, Katz and PageRank centralities. We define trust as a linear combination of similarity and centrality using a weighting parameter. The proposed framework is validated using real data from Epinions. Our result indicates that the proposed trust model produces better rating estimation in terms of the mean absolute error (MAE), the root mean squared error (RMSE) and error distribution, compared to the traditional binary trust model which is widely used in recommender systems. Trust enforced by degree centrality shows better performance compared to other centrality methods. The same conclusion is valid for connection-based similarity compared to rating-based. The trust relationships are also observed to be more dependent on the similarity rather than centrality. The proposed framework can thus be effectively applied to electronic retailers in promoting their products and services.

CHAPTER 7: USER PROFILE ANOMALY DETECTION

Recommender systems are subject to profile-injection attacks due to the recommender database being populated by users inputs. We propose a detection approach where each user profile is examined to extract attributes which are used to identify and label each user profile as either an attacker or a genuine user.

7.1 Detection Attributes

Our hypothesis [32] is based on the fact that the features of attackers would be significantly different from the overall statistical characteristics of all user profiles. This difference in features can be extracted from two different sources: i) the rating a user assigns to a product and ii) the relationship between those users. Earlier studies have shown that it is unlikely, if not unrealistic, that an attacker to have complete knowledge of the ratings or the connections in a real system. So the synthetically generated user profiles would be different from authentic user profiles. These differences can be quantified in different ways, including abnormal deviation from user's friends ratings assigned to the products, or a connection between two users with a low similarity value. As a result, a carefully designed criteria can capture the abnormalities and deviations which can help to identify potential attacker profiles.

One of the attributes is Rating Deviation from Mean Agreement (RDMA), which identifies attackers through examining the user's average deviation per item, weighted by the inverse of the number of ratings for that item. Motivated by RDMA, we propose a variant of it that is found to be valuable when used in conjunction with a clustering technique which is based on attributes derived from each individual profile. We propose the following attributes that can be used to differentiate between a genuine profile and an attacker profile.

7.1.1 Deviation from Predicted Rating

Several attributes for detecting the differences that occur in the presence of attackers were introduced in [25]. Users who deviate from their own prediction for a particular product can be considered as being malicious. We use the differences from the predicted value for all the ratings as a measure of *deviation* which can also be used to measure the error in the rating system. The matrix factorization method [55] is a popular prediction technique. Suppose $r^*_{u,j}$ is the predicted rating obtained via the matrix factorization method for user u for product j. Then the deviation by user u, denoted by $D(u)$, is obtained as:

$$D(u) = \frac{\sum_{j \in I_u} |r_{u,j} - r^*_{u,j}|}{nr_u} \tag{7.1}$$

where $r_{u,j}$ is the actual rating by the user u for the product j, and nr_u is the number of ratings by user u.

7.1.2 Similarity among Two Users

Connections between users in a social network are usually created between similar users or users who have similar interests. When an attacker joins the network, he tends to connect to users in a random fashion. Such artificiality created connections result in low similarity between the attacker and the users he connects to. Here, we capture the similarity between two users based on: i) the mutually rated products, and ii) the common connections that both have.

Rating Similarity

High similarity between users reveal that the users which are very likely to have the same taste are more likely to connect to each other. The effect of similarity has been incorporated in social recommender systems for predicting user rating. Here we apply the VSS algorithm to identify the similarity between users utilizing the common items that have been rated by both users v and u. Recall from chapter 6, the similarity is given by:

$$R(u,v) = \frac{\sum_{j \in I(u) \cap I(v)} R_{u,j} \cdot R_{v,j}}{\sqrt{\sum_{j \in I(u) \cap I(v)} R_{u,j}^2} \cdot \sqrt{\sum_{j \in I(u) \cap I(v)} R_{v,j}^2}} \tag{7.2}$$

where j is an item that both users u and v have rated and $R_{u,j}$ ($R_{v,j}$) is the rating that user u (v) assigned to item j. $I(u)$ represents the set of items rated by user u.

If we want to study from only one user's (attacker's) perspective, say user u, then there might not be many products that both users u and v have rated i.e., the set $I(u) \cap I(v)$ could be small. To expand the set, we also consider all the first hop neighbors of user u, denoted by $N(u)$.

We modify Eqn. 7.2 as:

$$R(u) = \frac{\sum_{v \in N(u)} R(u,v)}{|N(u)|} \tag{7.3}$$

Connection Similarity

The similarity between two users can also be measured by the connections they have in common. More mutual connections would indicate a larger similarity. For malicious connections, it is expected that the number of mutual connections would be low, and hence a small similarity value. We define the connection similarity as:

$$C(u,v) = \frac{|N(u) \cap N(v)|}{|N(u)|} \tag{7.4}$$

Just like the rating similarity of user u, we define the connection similarity for user u as:

$$C(u) = \frac{\sum_{v \in N(u)} C(u,v)}{|N(u)|} \tag{7.5}$$

7.1.3 Abnormal Rating Behavior

Now, we consider users with abnormal rating behavior. Abnormality could be manifested through several ways. We consider two cases.

Extreme Rating Behavior

Here, the users only assign either high ratings (e.g., 5) or low ratings (e.g., 1) to products. In such cases, we can expect the very low deviations among the ratings. We capture these extreme ratings for user u as:

$$E(u) = 1 - \frac{\sigma(R_u)}{R_h - R_l} \tag{7.6}$$

where $\sigma(R_u)$ is the standard deviation of all the ratings by user u. R_h and R_l are the highest and lowest ratings allowed by the recommender system. For extreme behavior, $E(u)$ will be close to 1. It is to be noted, a fix-rater, where the user always assigns products the same rating, also would have $E(u)$ close to 1.

Different Rating Behavior

Here, we try to identify users whose ratings vary significantly from their connections. This difference could be for the same product or it could be for a range of products. For user u, we consider the deviations of the ratings from $N(u)$ for product j as:

$$B_j(u) = \frac{\sum_{v \in N(u)} |r_{u,j} - r_{v,j}|^2}{N(u)} \tag{7.7}$$

A generalization of equation 7.7 would be to include not just the neighbors of user u but *all* users who rated product j.

$$B(u) = \frac{1}{Nu(j)} \sum_j B_j(u) \tag{7.8}$$

where $Nu(j)$ is number of products.

7.1.4 k-Means Clustering

The k-means algorithms are based on finding k centroids for the k clusters in a higher dimensional space. A standard model-based collaborative filtering algorithm uses k-means to cluster similar users. Given a set of user profiles, the space can be partitioned into k clusters– users belonging to a cluster are close to each other based on a measure of similarity. We use the similarity based on a user profile being authentic or being an attacker. So the clustering aims to make two clusters of user profiles based on the attributes discussed in section 7.1.

7.2 Experimental Evaluation

In order to verify the efficacy of our proposed framework, we conducted extensive simulation experiments. Before, we present the results, let us first discuss the performance metrics, datasets, and the attack models.

7.2.1 Evaluation Metrics

There are various metrics that are used to evaluate recommender systems [34]. Our aim is to measure the effectiveness of the method in differentiating between attackers and authentic users.

We use *precision* an *recall* for identifying attackers. Precision is a measure of exactness and is the ratio of the number of attackers identified to the total number of users who are identified as attackers. Recall is a measure of completeness and is the ratio of number of attackers identified to the total number of attackers in the system. These are defined as follows.

$$Precision = \frac{n_{TP}}{n_{TP} + n_{FP}} \tag{7.9}$$

$$Recall = \frac{n_{TP}}{n_{TP} + n_{FN}} \tag{7.10}$$

Here, n_{TP} is number of true positives which represents the number of attackers (user profiles) correctly classified as attackers, and n_{FP} is number of authentic profiles misclassified as attack profiles (i.e., false positives), and n_{FN} is the number of attack profiles misclassified as authentic profiles (i.e., false negatives).

7.2.2 Dataset

We use the publicly available Epinions dataset [122]. This dataset consists of 922267 ratings on 296277 products by 22166 users having 355754 relationships. All ratings are integer values between 1 and 5 where 1 is the worst and 5 is the best. From the dataset, we use the users-connection matrix in addition to the item-rating matrix. The density of user connections matrix is 0.0724 and for item-rating matrix is 0.0140. It can be observed that these matrices are relatively

sparse. For each attack profile, we also consider different attack sizes and filler sizes. For each attack, we injected a number of attack profiles and evaluated each scenario. The data has been tested by inserting a mix of attack models discussed next.

7.2.3 Attack Models

An attack type defines the algorithm for assigning ratings to the set of filler products and the target product. The set of filler products represents a group of randomly selected products in the database that are assigned ratings within the attack profile. For specific attack types, we selected a subset of filler products before a specific impact on the recommender system. For each attack profile, we considered four sets of products: i) a set of unrated products, ii) a set of filler products, iii) a set of products with specific characteristics which is determined by the attacker, and iv) one or more target products.

The two types of attacks that we considered are the random attack and the average attack which were introduced in [65]. In random attack, a maximum (push attack) or minimum (nuke attack) rating is assigned to the target product and random ratings are assigned to the filler products. For this attack model the selected product set is empty. In average attack model, the rating for each filler product is based on the mean rating of that product across the rating matrix. Generally an average attack is more effective than a random attack. However, it requires more knowledge about the system rating behavior and distribution. The cost of this knowledge can be minimized considering that an average attack could be successful with a smaller number of filler items. How-

ever, in random attack there needs to be a rating for every product so that it makes this attack more effective and efficient.

7.2.4 Experimental Setup

Several attack scenarios are simulated to evaluate the performance of the proposed framework. In each attack scenario, a number of fake profiles are injected into the system. The fake profiles start their attack by sending a number of connection requests to get attached to the existing users. Added profiles add-back the attackers randomly with some with probability, P_{ab}. In order to evade detection, the attackers also rate a number of fillers, i.e., they rate products in addition to the target item. For the average attack, the filler items are rated as the average of the ratings, whereas, the ratings to the filler items are assigned randomly in random attacks.

We considered the number of attackers between 5 and 50, number of filler items between 20 and 120, and the probability of add back between 0.1 and 0.6. The number of connection requests per attackers is set to 100 for all the scenarios. As mentioned earlier, matrix factorization is used to estimate the rating behavior of users. The number of latent for matrix factorization is set to 2.

7.2.5 Simulation Results

Fig. 7.1 shows the precision with varying number of attackers for push (random and average) and nuke (random and average). Fig. 7.2 shows the recall with varying number of attackers for push

(random and average) and nuke (random and average). As more attacks are launched, the proposed method is able to identify the attackers with better accuracy. A small number of attackers usually do not disturb the normal activity of the system significantly; therefore it is relatively difficult to detect those small size attacks.

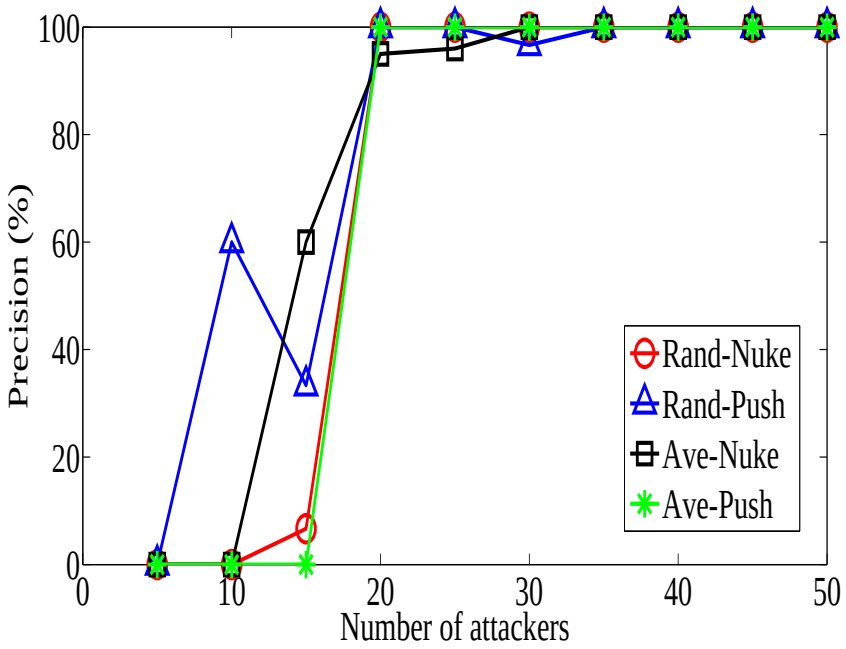

Figure 7.1 Effect of number of attackers on precision

Precision and recall are shown in Figs. 7.3 and 7.4 for increasing number of fillers. A higher number of fillers also increases the chance of detecting the attackers. This is because, attackers with high number of fillers are more likely to behave abnormally in terms of their rating behavior. In those cases, the attackers would have difficulty to produce ratings for fillers which are statistically consistent with the rest of the system. Intuitively, an average attack profile should be very similar to an authentic user profile than a random attack profile. As a result, for small filler sizes, it is difficult to differentiate the average attack from real users simply by checking the

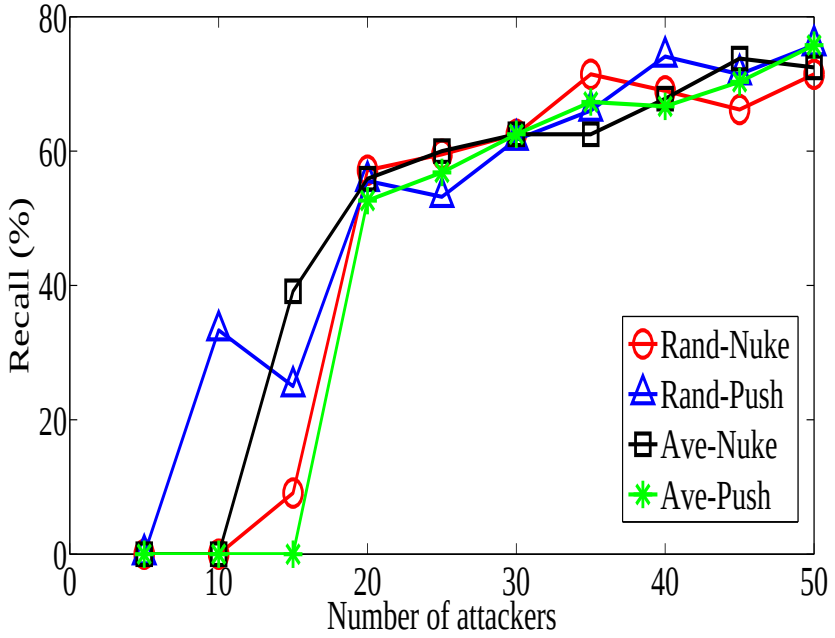

Figure 7.2 Effect of number of attackers on recall

difference in the average rating for that products. However, in random attacks with small filler sizes, random ratings would be more affected by the number of connections.

We also investigate the impact of add-back probability on precision and recall. From Figs. 7.5 and 7.6, we see that the probability of add-back inversely influences the performance of the anomaly detection. Higher add-back probability indicates social acceptance of the attackers by the rest of the users in the system. This will make the detection of the attackers difficult since they blend in with the genuine profiles. This clearly indicates the willingness to accept unknown connection requests plays an adverse role in anomaly detection.

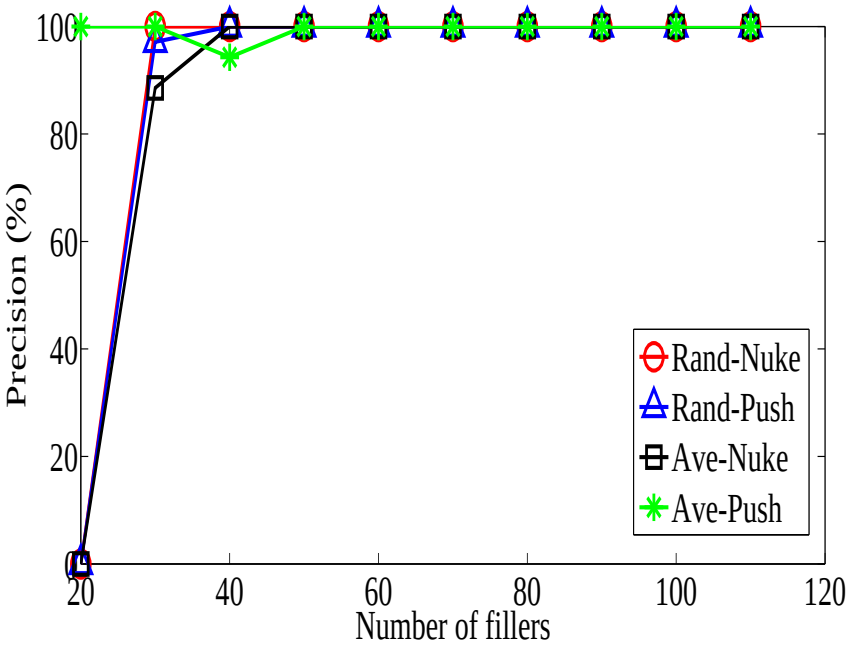

Figure 7.3 Effect of number of fillers on precision

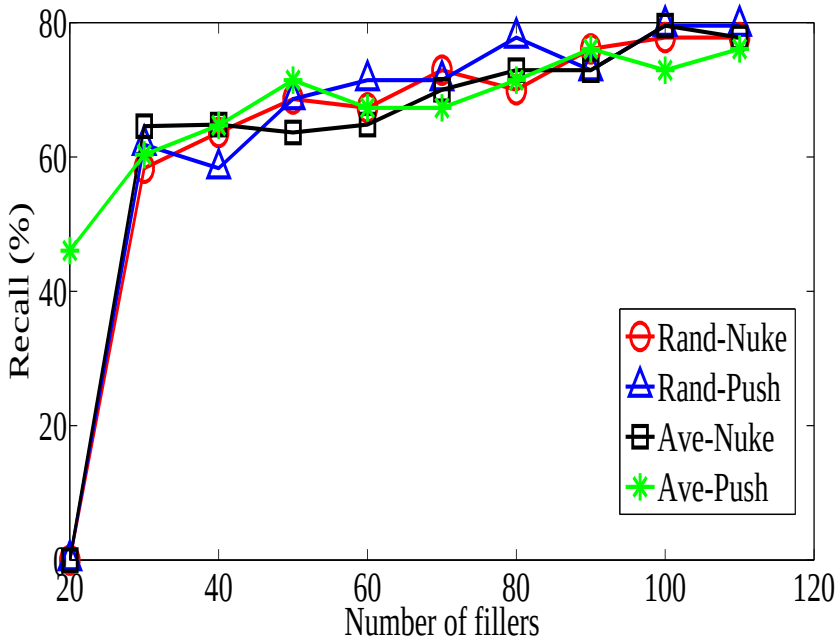

Figure 7.4 Effect of number of fillers on recall

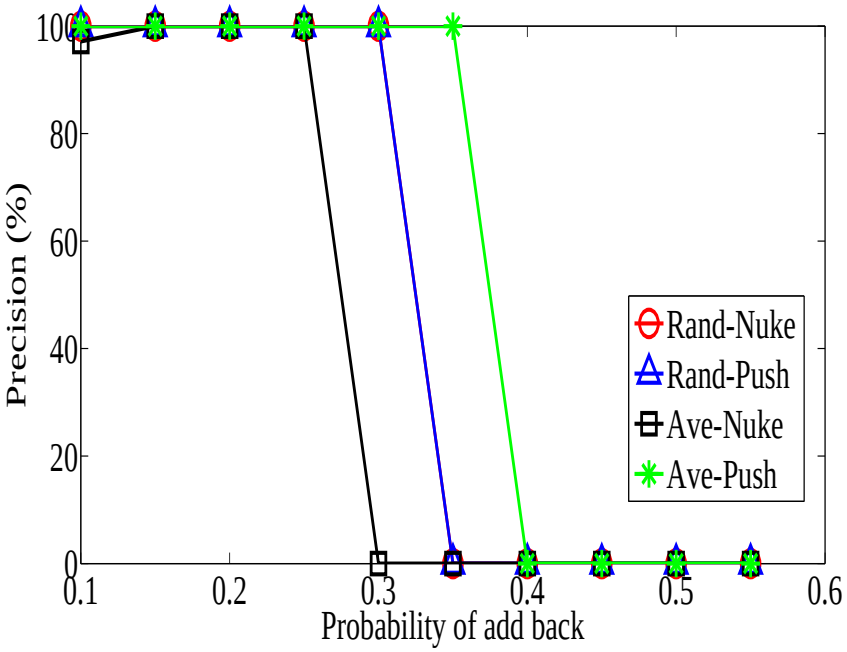

Figure 7.5 Effect of add-back probability on precision

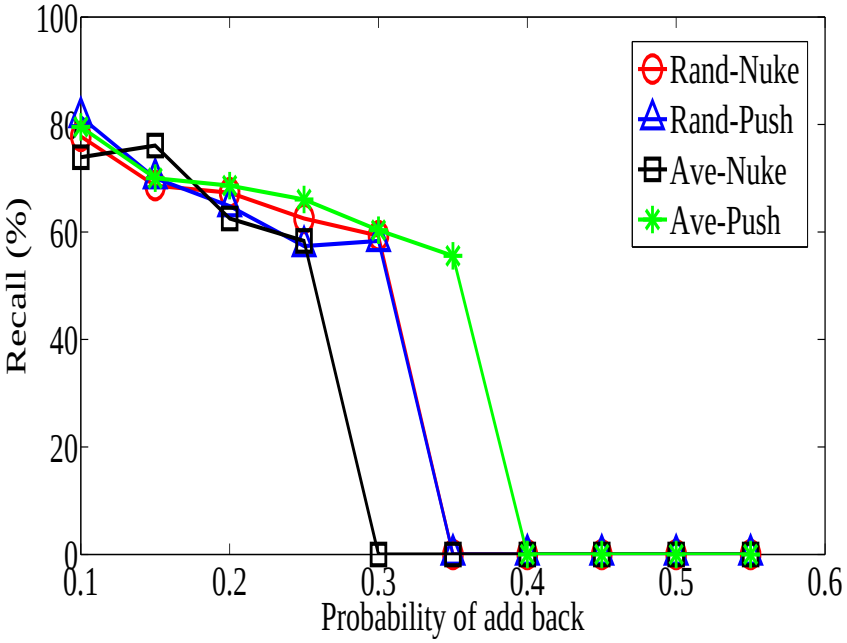

Figure 7.6 Effect of add-back probability on recall

7.3 Summary

Profile injection attacks threaten the trustworthiness of social recommender systems. Though there are techniques that try to identify such profiles, most of them are focused on individual profiles and ignore the social interactions between attackers and authentic users. In this chapter, we exploit the social connections to detect the anomalies of user profiles and the corresponding ratings using k-means clustering. We propose three attributes that capture the deviations of ratings, user similarities, and abnormal rating behavior. We use the Epinions dataset to evaluate the performance of our framework. we inject attacker profiles to the system to perform nuke and push attacks. These attackers randomly add multiple connections and assign biased ratings to the selected products. We use precision and recall as the performance metrics to show that k-means clustering algorithm that uses the three attributes can identify the attack profiles with high accuracy and low false positives.

CHAPTER 8: CONCLUSIONS

With the growing popularity of social networks, recommendation systems are becoming important due to their commercial, social, and political impacts. In this dissertation, we investigate various issues in online social recommender systems: information spreading, rating prediction, and malicious profile detection. We developed a probabilistic spreading model for information diffusion in online social networks and tested the proposed model using Facebook dataset. We also proposed a diffusion model to predict the same by considering the dynamic carrying capacity of the network. Our model is able to predict the influenced users at any time by minimizing the Mean Absolute Error (MAE) between the observed and predicted values. We used Genetic Algorithm with random initial guess for the error minimization. We validated our model using real data from Digg dataset.

For rating prediction of products in social recommender systems, we proposes a social trust model that incorporates the preference similarity, user's centrality, and social relation in order to predict the rating of a product. We capture the trust relationships between users considering users with similar profile and their importance. We argue that users with more similarity would trust each other more; also users with higher importance would be trusted more. Similarity is quantified by using rating-based approaches and connection-based centralities. The importance of users is modeled by degree, eigen-vector centrality, Katz, and PageRank centralities. We define trust as a linear combination of similarity and centrality using a weighting parameter. The proposed

framework is validated using real data from Epinions. Our result indicates that the proposed trust model produces better rating estimation in terms of the mean absolute error (MAE), the root mean squared error (RMSE) and error distribution, compared to the traditional binary trust model which is widely used in recommender systems. Trust enforced by degree centrality shows better performance compared to other centrality methods. The same conclusion is valid for connection-based similarity compared to rating-based. The trust relationships are also observed to be more dependent on the similarity rather than centrality. The proposed framework can thus be effectively applied to electronic retailers in promoting their products and services.

As for identifying profile injection attacks that threaten the trustworthiness of social recommender systems, we exploit the social connections to detect the anomalies of user profiles and the corresponding ratings using clustering methods. We propose multiple attributes that capture the deviations of ratings, user similarities, and abnormal rating behavior. We use the Epinions dataset to evaluate the performance of our framework. We inject attacker profiles to the system to launch nuke and push attacks. These attacks randomly add multiple connections and assign biased ratings to the selected products. We use precision and recall as the performance metrics using the attributes to help identify the attack profiles with high accuracy and low false positives.

LIST OF REFERENCES

[1] A. Abdul-Rahman, S. Hailes, "Supporting trust in virtual communities", Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", IEEE Trans. on Knowledge and Data Engineering, vol. 17, pp. 734–749, 2005.

[3] Levien and Aiken, Advogatos trust metric, online at http://advogato.org/trust-metric.html, 2002.

[4] X. Amatriain, J. M. Pujol, and N. Oliver, "I Like It... I Like It Not: Evaluating User Ratings Noise in Recommender Systems", International Conference on User Modeling, Adaptation, and Personalization, pp. 247–258, 2009.

[5] W. An, "Models and methods to identify peer effects", The SAGE Handbook of Social Network Analysis, 2010.

[6] P. Avesani, P. Massa, and R. Tiella, "A trust-aware recommender system for ski mountaineering", International Journal for Infonomics, Vol. 20, pp. 1–10, 2005.

[7] P. Van Baalen, J. Bloemhof-Ruwaard, and E. van Heck, "Knowledge sharing in an emerging network of practice", European Management Journal, Vol. 23, pp. 300–314, 2005.

[8] M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation, Communication of ACM, vol. 40, no. 3, pp. 66–72, 1997.

[9] A-L. Barabasi, "Scale-Free Networks: A Decade and Beyond", Science (24) July 2009, Vol. 325 no. 5939 pp. 412–413.

[10] A.L. Barabasi and R. Albert, "Emergence of Scaling in Random Networks", Science (15), 1999, Vol. 286, no. 5439, pp. 509-512.

[11] R. Bhaumik, B. Mobasher, R. D. Burke, "A clustering approach to unsupervised attack detection in collaborative recommender systems", Proceedings of the 7th IEEE international conference on data mining, pp. 181–187, 2011.

[12] R. Bhaumik, C. A. Williams, B. Mobasher and R. D. Burke, "Securing collaborative filtering against malicious attacks through anomaly detection", Proceedings of the 4th workshop on intelligent techniques for web personalization, 2006.

[13] J. S. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering”, Uncertainty in Artificial Intelligence, pp. 43–52, 1998.

[14] J. Brown and P. Reinegen, “Social ties and word-of-mouth referral behavior”, Journal of Consumer Research, 14:3 (1987), pp. 350–362.

[15] A. Barrat, M. Barthlemy, A. Vespignani, “Dynamical Processes on Complex Networks”, Cambridge University Press, 2008.

[16] J. Buder and C. Schwind, “Learning with personalized recommender systems: a psychological view”, Computer Human Behavior, Vol. 28, pp. 207-216, 2012.

[17] R. D. Burke, M. P. OMahony and N. J. Hurley, “Robust collaborative recommendation”, Recommender systems handbook. Springer, pp. 805–835, 2011.

[18] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, “Classification features for attack detection in collaborative recommender systems”, In Proceedings of the 12th ACM SIGKDD, pp 542–547, 2006.

[19] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik, “Detecting profile injection attacks in collaborative recommender systems”, In Proceedings of the IEEE Joint Conference on E-Commerce Technology and Enterprise Computing, E-Commerce and E-Services (CEC/EEE 2006), 2006.

[20] R. D. Burke, B. Mobasher, R. Zabicki, “Identifying Attack Models for Secure Recommendation”, WebKDD Workshop on the Next Generation of Recommender Systems Research, pp. 19–25, 2005.

[21] J. Canny, “Collaborative filtering with privacy via factor analysis”, 25th international ACM SIGIR conference on Research and development in information retrieval, pp 238–245, 2002.

[22] R. Cantrell and C. Cosner, “Spatial Ecology via Reaction-Diffusion Equation”, Wiley, 2003.

[23] M. Cha, A. Mislove, B. Adams, K. Gummadi, “Characterizing Social Cascades in Flickr”, in Proceedings of ACM SIGCOMM Workshop on Social Networks (WOSN), 2008.

[24] Z. Cheng, N. J. Hurley, “Robust collaborative recommendation by least trimmed squares matrix factorization”, IEEE conference on tools with artificial intelligence, pp. 105–112, 2010.

[25] P. A. Chirita, W. Nejdl, and C Zamfir, “Preventing shilling attacks in online recommender systems’, Proceedings of the 7th annual ACM international workshop on Web information and data management, pp. 67-74, 2005.

[26] K. Choi, D. Yoo, G. Kim, Y. Suh, “A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis”, Electronic Commerce Research and Applications, vol. 11, pp. 309-317, 2012.

[27] M. Chowdhury, A. Thomo, B. Wadge, Trust-based in finitesimals for enhanced collaborative filtering, Proceedings of the 15th International Conference on Management of Data (COMAD), 2009.

[28] C. Y. Chung, P. Y. Hsu and S. H. Huang, "A novel approach to filter out malicious rating profiles from recommender systems", Decision Support Systems, vol. 55, No. 1, pp. 314–325, 2013.

[29] A. Davoudi, and M. Chatterjee, "Probabilistic Spreading of Recommendations in Social Networks", IEEE Military Communication Conference, IEEE MILCOM, pp. 1–6, 2015.

[30] A. Davoudi, M. Chatterjee, "Product Rating Prediction Using Centrality Measures in Social Networks", 36th IEEE Sarnoff Symposium, 2015.

[31] A. Davoudi and M. Chatterjee, "Prediction of Information Diffusion in Social Networks using Dynamic Carrying Capacity ", In Proceedings of IEEE International Conference on Big Data (BigData), pp. 2466–2469, 2016.

[32] A. Davoudi, and M. Chatterjee, "Detection of Profile Injection Attacks in Social Recommender Systems Using Outlier Analysis", In Proceedings of IEEE International Conference on Big Data, IEEE BigData, 2017.

[33] A. Davoudi, and M. Chatterjee. "Product rating prediction using trust relationships in social networks", IEEE Consumer Communications and Networking Conference (CCNC), pp. 115–118, 2016.

[34] A. Davoudi, M. Chatterjee, "Modeling Trust for Rating Prediction in Recommender Systems", SIAM data mining, International Workshop on Machine Learning Methods for Recommender Systems, pp. 1–8, 2016.

[35] A. Davoudi, and M. Chatterjee, "Social trust model for rating prediction in recommender systems: Effects of similarity, centrality, and social ties", Online Social Networks and Media, Vol. 7, pp. 1–11, 2018.

[36] P. DeMeo, A. Nocera, G. Terracina, D. Ursino, "Recommendation of similar users, resources and social networks in a social internetworking scenario", Information Sciences, Vol. 7, pp. 1285-1305, 2011.

[37] M. Deshpande and G. Karypis, "Item-Based Top N-Recommendation, ACM Transactions on Information Systems, Vol. 22, pp. 143–177, 2004.

[38] Y. Dong, Ch. Zhao, W. Cheng, L. Li, and L. Liu, "A Personalized Recommendation Algorithm with User Trust in Social Network", International Conference of Young Computer Scientists, Engineers and Educators (ICYCSEE), pp. 6376, 2016.

[39] P. Fife, "Spatial Ecology via Reaction-Diffusion Equation", Mathematical Aspects of Reacting and Diffusing Systems, 1979.

[40] Z. Fuguo, “Analysis of Profile Injection Attacks against Recommendation Algorithms on Bipartite Networks”, International Conference on Management of e-Commerce and e-Government, pp. 1–5, 2014.

[41] R. Ghosh and K. Lerman, “A framework for quantitative analysis of cascades on networks”, ACM International Conference on Web search and data mining, 2011.

[42] W. Goffman and V. A. Newwill, “Generalization of epidemic theory: An application to the transmission of ideas”, Nature, pp. 225–228, 1964.

[43] J. Golbeck, “Computing and Applying Trust in Web-based Social Networks”, PhD thesis, University of Maryland College Park, 2005.

[44] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, “Information diffusion through blogspace”, in Proceedings of International Conference on World Wide Web (WWW), 2004.

[45] A. Guille, H. Hacid, C. Favre and D. Zighed, “Information Diffusion in Online Social Networks: a Survey”, ACM SIGMOD Record, Vol. 42, pp. 17–28, 2013.

[46] I. Gunes, C. Kaleli, A. Bilge and H. Polat, “Shilling attacks against recommender systems: A comprehensive survey”, Artificial Intelligence Review, vol. 11, pp. 767–799, 2014.

[47] G. Guo, J. Zhang, D. Thalmann, “A simple but effective method to incorporate trusted neighbors in recommender systems”, Proceeding of the 20th Conference on User Modeling, Adaptation, and Personalization (UMAP), pp. 114–125, 2012.

[48] U. Hanani, B. Shapira, P. Shoval, “Information filtering: overview of issues, research and systems”, User Modeling and User-Adapted Interaction, Vol. 11, pp. 203-259, 2001.

[49] S. J. Hardiman and L. Katzir, “Estimating Clustering Coefficients and Size of Social Networks via Random Walk”, Proceedings of the 22nd international conference on World Wide Web, 2013, pp. 539–550.

[50] J. Herlocker, J. Konstan J., A. Borchers, and J. Riedl, “An Algorithmic Framework for Performing Collaborative Filtering”, ACM SIGIR Conference, pp. 230–237 1999.

[51] T. Hofmann, and J. Puzicha, “Latent Class Models for Collaborative Filtering”, In Proceedings of the IJCAI, Vol. 99, pp. 688–693, 1999.

[52] T. Hogg and K. Lerman, “Social Dynamics of Digg”, EPJ Data Science, Vol. 1, 2012.

[53] N. J. Hurley, Z. Cheng and M. Zhang, “Statistical attack detection”, Proceedings of the 3rd ACM RecSys, pp. 149–156, 2009.

[54] M. Jamali and M. Ester, “Trustwalker: a random walk model for combining trust-based and item-based recommendation”, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406, 2009.

[55] M. Jamali and M. Ester, “A matrix factorization technique with trust propagation for recommendation in social networks”, In Proceeding of the ACM RecSys Conference, pp. 135–142, 2010.

[56] A. Josang, R. Ismail and Colin Boyd, “Survey of trust and reputation systems for online service provision”, Decision Support Systems, pp. 618-644, 2005.

[57] C. Jensen, J. Davis, Sh. Farnham, “Finding others online: reputation systems for social online spaces”, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 447–454, 2002.

[58] A. Josang, R. Ismail, C. Boyd, “A survey of trust and reputation systems for online service provision”, Decision Support Systems, vol. 43, pp. 618–644, 2007.

[59] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina, “The Eigentrust Algorithm for Reputation Management in P2P Networks”, Proceedings of the 12th International Conference on World Wide Web, pp. 640–651, 2003.

[60] M. J. Keeling and K. T.D Eames, “Networks and epidemic models”, Journal of The Royal Society Interface, 2005.

[61] D. Kempe, J. Kleinberg and E. Tardos, “Maximizing the spread of influence through a social network”, ACM SIGKDD 2003, pp. 137-146.

[62] Y. A. Kim and J. Srivastava, “Impact of social influence in e-commerce decision making”, Proceedings of the ninth international conference on Electronic commerce, ACM, pp. 293-302, 2007.

[63] A. Kohrs and B. Merialdo, “Clustering for Collaborative Filtering Applications”, In proc. of the International conference on Computational Intelligence for Modeling Control and Automation, 1999.

[64] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques For Recommender Systems”, Computer vol. 8, pp. 30–37, 2009.

[65] S. Lam and J. Reidl, “Shilling recommender systems for fun and profit”, In Proceedings of the 13th International WWW Conference, 2004.

[66] K. O. Lee, N. Shi, M.K. Cheung, H. Lim and C.L. Sia, “Consumer’s decision to shop online: the moderating role of positive informational social influence”, Information Management, vol. 48, pp. 185-191, 2001.

[67] K. Lerman and R. Ghosh, “Information contagion: An empirical study of the spread of news on digg and twitter social networks”, in Proc. of Intl. Conference on Weblogs and Social Media (ICWSM), 2010.

[68] J. Leskovec, “Tutorial: Analytics & predictive models for social media”, in Proceedings of International Conference on World Wide Web (WWW), 2011.

[69] J. Leskovec, A. Singh, and J. Kleinberg, “Patterns of influence in a recommendation network, In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2006, pp. 380–389.

[70] Sh. Liang, L. Ma, and F. Yuan, “Reconstructing Trust Matrix to Improve Prediction Accuracy and Solve Cold User Problem in Recommender Systems”, UMAP, CEUR workshop, 2016.

[71] G. Linden, B. Smith, and J. York,“Amazon.com recommendations: Item-to-item collaborative filtering”, IEEE Internet Computing, pp.76–80, 2003.

[72] Z.B. Liu, W.Y. Qu, H.T. Li, and C.S. Xie, “A hybrid collaborative filtering recommendation mechanism for P2P networks”, Future Generation Computer Systems, vol. 26, pp. 1409-1417, 2010.

[73] L. Y. Lu, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang and T. Zhou, “Recommender System”, Physics Reports, vol. 519, No. 1, pp. 1–49, 2012.

[74] H. Ma, “On measuring social friend interest similarities in recommender systems”, In Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, pp. 465–474, 2014.

[75] H. Ma, D. Zhou, C. Liu, M. R. Lyu and I. King, “Recommender systems with social regularization”, In proc. of ACM WSDM, pp. 287–296, 2011.

[76] H. Ma, I. King and M. R. Lyu, “Learning to Recommend with Explicit and Implicit Social Relations”, ACM Transaction Intelligent Systems Technology, Vol. 2, 2011.

[77] H. Ma, M.R. Lyu, I. King, Learning to recommend with trust and distrust relationships, Proceedings of the third ACM Conference on Recommender Systems, pp. 189–196, 2009.

[78] H. Ma, H. Yang, M. R. Lyu and I. King, “SoRec: Social Recommendation Using Probabilistic Matrix Factorization”, In proc. of ACM CIKM, pp. 931–940, 2008.

[79] K. Madani, M. Hooshyar, S. Khatami, A. Alaeipour, and A. Moeini, “Nash-Reinforcement Learning (N-RL) for Developing Coordination Strategies in Non-Transferable Utility Games”, IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 2705–2710, 2014.

[80] K. Madani, and M. Hooshyar, “A game theory–reinforcement learning (GT–RL) method to develop optimal operation policies for multi-operator reservoir systems”, Journal of Hydrology, Vol. 519, pp. 732–742, 2014.

[81] P. Marsden and N. Friedkin, “Network studies of social influence”, Sociological Methods and Research, Vol. 22, pp. 127–151, 1993.

[82] S. Marsh, “Formalizing Trust as a Computational Concept”, PhD thesis, Department of Mathematics and Computer Science, University of Stirling, Stirling, UK, 1994.

[83] P. Massa and P. Avesani, “Trust-aware Collaborative Filtering for Recommender Systems”, ACM conference on Recommender Systems, RecSys, pp. 17–24, 2007.

[84] P. Massa, and P. Avessani, “Trust-aware Collaborative Filtering for Recommender Systems”, In Proceedings of Federated International Conference on the Move to Meaningful Internet, pp. 492–508, 2004.

[85] J. McAuley and J. Leskovec. (NIPS, 2012). Learning to Discover Social Circles in Ego Networks. [online]. Available: http://snap.stanford.edu/data/egonets-Facebook.html

[86] M. McPherson, L. Smith-Lovin, and J. Cook, “Birds of a feather: Homophily in social networks”, Annual review of sociology, pp. 415–444, 2001.

[87] B. Mehta, “Unsupervised shilling detection for collaborative filtering”, Association for the Advancement of Artificial Intelligence, 2007.

[88] B. Mehta, T. Hofmann, “A survey of attack-resistant collaborative filtering algorithms”, IEEE Data Engineering Bulletin, Vol. 31, pp. 14-22, 2008.

[89] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations”, In Proceedings of the 18th National Conference on Artificial Intelligence, pp. 187–192, 2002.

[90] S. L. Miller and D. G. Childers, Probability and Random Processes: With Applications to Signal Processing and Communications, 2nd ed., Academic Press, 2004.

[91] B. Mobasher, R. D. Burke, R. Bhaumik and J. J. Sandvig, “Attacks and remedies in collaborative recommendation”, IEEE Intelligent Systems, vol. 22, No. 3, pp. 56–63, 2007.

[92] B. Mobasher, R. D. Burke, R. Bhaumik, C. A. Williams, “Towards Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness”, ACM Transaction on Internet Technology, Vol. 7, pp. 23–60, 2007.

[93] J.D. Murray, “Mathematical Biology I. An Introduction”, Springer-Verlag, New York, 1989.

[94] M. Newman, “The Structure and Function of Complex Networks”, SIAM REVIEW, Vol. 45), pp. 167–256, 2003.

[95] M. Newman, “Networks: an introduction”, Oxford University Press, 2010.

[96] J. O’Donovan and B. Smyth, “Trust in Recommender Systems”, In Proceeding of IUI, pp. 167–174, 2005.

[97] M. O'Mahony, N. Hurley, G. Silvestre, "Attacking recommender systems: The cost of promotion", workshop on recommender systems, European conference on artificial intelligence, pp. 24–28, 2006.

[98] M. OMahony, N.J. Hurley, and G. Silvestre, "Utility-based neighborhood formation for efficient and robust collaborative filtering", In Proceedings of the 5th ACM Conference on Electronic Commerce, pp. 260–261, 2004.

[99] M. O'Mahony, N. Hurley, N. Kushmerick, G. Silvestre, "Collaborative recommendation: A robustness analysis", ACM Transactions on Internet Technology, vol. 4, pp. 344–377, 2004.

[100] M. OMahony, N. Hurley, G. Silvestre, "Promoting recommendations: an attack on collaborative filtering", In Proceedings of the 13th international conference on database and expert systems applications, pp. 494–503, 2002.

[101] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach", Uncertainty in Artificial Intelligence, pp. 473–480, 2000.

[102] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web", Technical report, Stanford, 1998.

[103] M. J. Pazzan and D. Billsus, "Content-based recommendation systems", The adaptive web, Springer, pp.325–341, 2007.

[104] M. J. Pazzan and D. Billsus, "Content-based recommendation systems", Adaptive web, Springer, pp.325–341, 2007.

[105] D. M. Pennock, E. Horvitz, S. Lawrence, and C. L. Giles, "Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach", Uncertainty in Artificial Intelligence, pp. 473–480, 2000.

[106] l. A. Rahman and S. Hailes, "Supporting trust in virtual communities", 33rd Hawaii International Conference on System Sciences, 2000.

[107] S. Ray, A. Mahanti, "Improving prediction accuracy in trust-aware recommender systems", Proceedings of the 43rd Hawaii International Conference on System Sciences (HICSS), pp. 1-9, 2010.

[108] S. Ray, A. Mahanti, "Strategies for Effective Shilling Attacks Against Recommender Systems", Lecture Notes Computer Science, Vol. 5456, pp. 111–125, 2009.

[109] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture collaborative filtering of netnews", ACM CSCW, pp. 175–186, 1994.

[110] P. Resnick and H.R. Varian, "Recommender systems", Communications of the ACM, 40 (3), pp. 56-58.

[111] M. Richardson, R. Agarwal and P. Domingos, “Trust Management for Semantic Web”, Proceedings of the Second International Semantic Web Conference, 2003.

[112] M. Richardson and P. Domingos, “Mining knowledge-sharing sites for viral marketing”, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2002, pp. 61–70.

[113] K. Saito, M. Kimura, K. Ohara, and H. Motoda, “Efficient Discovery of Influential Nodes for SIS Models in Social Networks”, Knowledge and Information Systems, 2011.

[114] R. Salakhutdinov and A. Mnih,“Probabilistic matrix factorization”, Advances in Neural Information Processing Systems, Vol. 20, 2008.

[115] R. Salakhutdinov and A. Mnih,“Bayesian Probabilistic Matrix Factorization Using Markov Chain Monte Carlo” In Proceedings of International Conference on Machine Learning, 2008.

[116] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms”, World Wide Web conference, pp. 285–295, 2001.

[117] J. B. Schafer, D. Frankowski, and J. Herlocker, “Collaborative filtering recommender systems”, Adaptive Web, Springer, pp. 291–324, 2007.

[118] J.B. Schafer, J.A. Konstan, J. Riedl, “E-commerce recommendation applications”, Data Mining and Knowledge Discovery, vol. 5, pp. 115–153, 2001.

[119] R. Sinha and K. Swearingen, “Comparing recommendations made by online systems and friends”, DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries, pp. 73–78, June 2001.

[120] X. Song, Y. Chi, K. Hino, and B. L. Tseng, “Information flow modeling based on diffusion rate for prediction and ranking”, in Proceedings of International Conference on World Wide Web, 2007.

[121] G. Steeg, R. Ghosh, and K. Lerman, “What stops social epidemics?”, arXiv preprint arXiv:1102.1985, 2011.

[122] J. Tang. [online]. Available: www.public.asu.edu/ jtang20/datasetcode/truststudy.htm

[123] J. Tang, X. Hu and H. Liu, “Social recommendation: a review”, Social Network Analysis and Mining December 2013, Vol. 3, pp. 1113–1133.

[124] S. Tang, N. Blenn, C. Doerr, and P. V. Mieghem, “Digging in the Digg Social News Website”, IEEE Transactions on Multimedia, vol. 13, no. 5, pp. 1163–1175, 2011.

[125] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, “Propagation of trust and distrust”, World Wide Web, pp. 403–412, 2004.

[126] P. Victor, C. Cornelis, “Trust Networks for Recommender Systems”, Springer Science and Business Media, Vol. 4, 2011.

[127] D.J. Watts and S.H. Strogatz, “Collective dynamics of ‘small-world’ networks”, Nature (393), June 1998, pp. 440–442.

[128] F. Wang, H. Wang, and K. Xu, “Diffusive logistic model towards predicting information diffusion in online social networks”, in Proceedings of IEEE ICDCS Workshop on Peer-to-Peer Computing and Online Social Networking (HOTPOST), 2012.

[129] C. Williams, B. Mobasher, R. Burke, “Defending recommender systems: detection of profile injection attacks”, Service Oriented Computer Applications, Vol. 1, pp. 157-170, 2007.

[130] C. A. Williams, B. Mobasher, R. D. Burke, R. Bhaumik, J. J. Sandvig, “Detection of obfuscated attacks in collaborative recommender systems”, Workshop on recommender systems, European conference on artificial intelligence, pp. 19–23, 2006.

[131] J. Yang and J. Leskovec, “Modeling information diffusion in implicit networks”, In ICDM, pp. 599–608, 2010.

[132] L. Ye, Ch. Wu, and M. Li, “Collaborative Filtering Recommendation Based on Trust Model with Fused Similar Factor”, MATEC Web of Conferences, ICMITE, 2017.

[133] B. Yu, and H. Fei, “Modeling Social Cascade in the Flickr Social Network”, in Proceedings of International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2009.

[134] F. G. Zhang, “Reverse Bandwagon Profile Injection Attack Against recommender systems”, symposium on computational intelligence and design, pp. 15–18, 2009.

[135] S. Zhang, and J. Ford, “Analysis of a low dimensional linear model under recommendation attacks”, ACM SIGIR, pp. 119–128, 2006.

[136] L. Zhen, Z. Jiang, H. Song, “Distributed recommender for peer-to-peer knowledge sharing”, Information Sciences, Vol. 18, pp. 3546–561, 2010.

[137] X. Zheng, Y. Wang, M. A. Orgun, Y. Zhong and G. Liu, “Trust Prediction with Propagation and Similarity Regularization”, Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 237–243, 2014.

[138] W. Zhou, J. Wen , Y. S. Koh, Q. Xiong, M. Gao, G. Dobbie, and Sh. Alam, “Shilling Attacks Detection in Recommender Systems Based on Target Item Analysis”, PLOS ONE journal, 2015.

[139] C. N. Ziegler, G. Lausen, “Spreading Activation Models for Trust Propagation”, Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service, 2004.