EXPLICIT AND IMPLICIT TRUST MODELING COMPATIBILITY AND EXPLICIT TRUST LINK PREDICTION IN TRUST-BASED RECOMMENDATION SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET UTKU DEMIRCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2022

Approval of the thesis:

EXPLICIT AND IMPLICIT TRUST MODELING COMPATIBILITY AND EXPLICIT TRUST LINK PREDICTION IN TRUST-BASED RECOMMENDATION SYSTEMS

submitted by **MEHMET UTKU DEMIRCI** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _____

Prof. Dr. Pınar Karagöz
Supervisor, **Computer Engineering, METU** _____

Examining Committee Members:

Prof. Dr. İsmail Hakkı Toroslu
Computer Engineering, METU _____

Prof. Dr. Pınar Karagöz
Computer Engineering, METU _____

Prof. Dr. Suat Özdemir
Computer Engineering, Hacettepe University _____

Date: 31.01.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mehmet Utku Demirci

Signature :

ABSTRACT

EXPLICIT AND IMPLICIT TRUST MODELING COMPATIBILITY AND EXPLICIT TRUST LINK PREDICTION IN TRUST-BASED RECOMMENDATION SYSTEMS

Demirci, Mehmet Utku
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Pınar Karagöz

January 2022, 47 pages

In social networks, *trust* is a fundamental notion affecting the nature and the strength of ties between individuals. It is also a piece of useful auxiliary information for improving the performance of recommendation systems. The number of ratings given by a user is minimal compared to all items in popular, widely-used e-commerce sites. Therefore, the user-item matrix that is used in collaborative filtering suffers from *data sparsity*, resulting in poor recommendation quality. Another issue is the *cold start* problem, which occurs for the inclusion of new users and new items to the system. *Trust* notion helps alleviate the effect of these problems by providing additional relationships between the users and pointing out strong relationships. Information as to the trust between users can be *explicitly* available. However, such information is not widely available, and hence *implicit* trust models have been employed. This work analyzes two sub-problems under trust modeling for recommendation: (1) What is the relationship between explicit and implicit trust scores, are they replaceable? (2) Can the explicit trust in a trust network be modeled? An implicit trust model is presented for the first problem, and the compatibility of implicit and explicit trust scores is analyzed. For the second problem, two different explicit trust models are proposed:

Explicit trust modeling through users' rating behavior and explicit trust modeling as a link prediction problem. The performances of the generated prediction models are analyzed on a set of benchmark data sets.

Keywords: Trust modeling, Implicit trust, Explicit trust, Recommendation, Recommender systems, Supervised learning

ÖZ

GÜVENE DAYALI ÖNERİ SİSTEMLERİNDE AÇIK VE ÖRTÜK GÜVEN MODELLEME UYUMLULUĞU VE AÇIK GÜVEN BAĞLANTISI TAHMİNİ

Demirci, Mehmet Utku
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Pınar Karagöz

Ocak 2022 , 47 sayfa

Sosyal ağlarda *güven*, bireyler arasındaki bağların doğasını ve gücünü etkileyen temel bir kavramdır. Ayrıca, öneri sistemlerinin performansını iyileştirmek için faydalı bir yardımcı bilgi parçasıdır. Bir kullanıcı tarafından verilen puan sayısı, popüler, yaygın olarak kullanılan e-ticaret sitelerindeki tüm öğelere kıyasla minimum düzeydedir. Bu nedenle, işbirlikçi filtrelemede kullanılan kullanıcı-öğe matrisi, *veri seyrekliğinden* muzdariptir ve bu da düşük öneri kalitesine neden olur. Diğer bir konu ise yeni kullanıcıların ve yeni öğelerin sisteme dahil edilmesiyle ortaya çıkan *soğuk başlangıç sorunudur*. *Güven* kavramı, kullanıcılar arasında ek ilişkiler sağlayarak ve güçlü ilişkilere işaret ederek bu sorunların etkisini hafifletmeye yardımcı olur. Kullanıcılar arasındaki güvene ilişkin bilgiler *açıkça* mevcut olabilir. Bununla birlikte, bu tür bilgiler yaygın olarak mevcut değildir ve bu nedenle *örtük* güven modelleri kullanılmıştır. Bu çalışma, tavsiye için güven modellemesi altındaki iki alt problemi analiz etmektedir: (1) Açık ve örtük güven puanları arasındaki ilişki nedir, bunlar değiştirilebilir mi? (2) Bir güven ağında açık güveni modelleyebilir miyiz? İlk problem için bir örtük güven modeli sunulmakta ve örtük ve açık güven puanlarının uyumluluğu

analiz edilmektedir. İkinci problem için, iki farklı açık güven modeli önerilmiştir: Kullanıcıların derecelendirme davranışı yoluyla açık güven modellemesi ve bir bağlantı tahmin problemi olarak açık güven modellemesi. Üretilen tahmin modellerinin performansları, bir dizi kıyaslama veri seti üzerinde analiz edilir.

Anahtar Kelimeler: Güven modelleme, Örtülü güven, Açık güven, Tavsiye, Öneri sistemleri, Gözetimli öğrenme

To my family

ACKNOWLEDGMENTS

I would like to express my sincere thanks and gratitude to my supervisor, Prof. Dr. Pınar Karagöz, for her support and assistance during both my undergraduate and graduate studies. She always helped and guided me with her helpful, friendly and understanding attitude when I was stuck or unmotivated during my graduation project and thesis writing periods.

I would also like to express my gratitude and love to my family for supporting and believing in me throughout my education life.

I would like to thank my dear friends Ezelsu Şimşek Yılgın and Cem Küçük, who were with me throughout my university life and with whom we worked until late for exams and worked on projects.

I would also like to express my special thanks to my dear friend Erdem Yazar, who was my guarantor for the scholarship program. I would also like to thank my friends Damla Erkoç, Tuna Berk Kaya, Gizem Aslaner, Ezgi Özel and Oğuzhan Yıldız, who annoyed me and motivated me by asking me how my thesis was going on a regular basis.

I would also like to thank HAVELSAN, where I was a full-time employee when I was a graduate student, for providing flexibility in terms of time.

Finally, I also thank the Scientific and Technological Research Council of Turkey (TUBITAK) for providing financial means through this study. This work is supported by TUBITAK with grant no 118E356, and my graduate studies are supported by the 2210/A scholarship program.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

CF	Collaborative Filtering
SVM	Support Vector Machine
ITRA	Implicit Trust Recommendation Approach
SBPR	Social Bayesian Personalized Ranking
SREE	Social Recommendation Approach Based On Euclidean Embedding
TBPR	Bayesian Personalized Ranking with Strong and Weak Ties

CHAPTER 1

INTRODUCTION

Many e-commerce systems and online streaming applications, such as Spotify[1] and Netflix[2], offer item recommendations to their users. Once this was a new feature in the past, yet recommendation systems are now indispensable for such applications. The widespread use of recommendation systems is not without a reason. We live in the age of big data and there is a large number of content in the internet. The positive side is that users can easily access any information. On the other hand, among huge amount of options, finding what they really need is a challenging experience for users. Recommendation systems make it easier for users to access the relevant items in this regard. This also positively affects the perception of the users about the applications by improving the user experience.

User experience on items can be described in two categories. *Explicit rating* is a score that a user gives for an item based on the experience. The rating given by the user is the most direct way of interpreting a user's opinion about a product. On the other hand, *implicit rating* indirectly provides information about the relationships between users and items, based on activities of users such as clicking, searching some keywords, purchases etc.

The most popular approach in recommendation systems is the Collaborative Filtering (CF) method [1]. On the basis of the assumption that the similar behaviors in the past reflect future behaviors, CF based recommendation systems use the similarity between users' past preferences. However CF suffers from well-known *data sparsity* and *cold start* problems. In order to overcome the performance degrading due to such

[1] spotify.com
[2] netflix.com

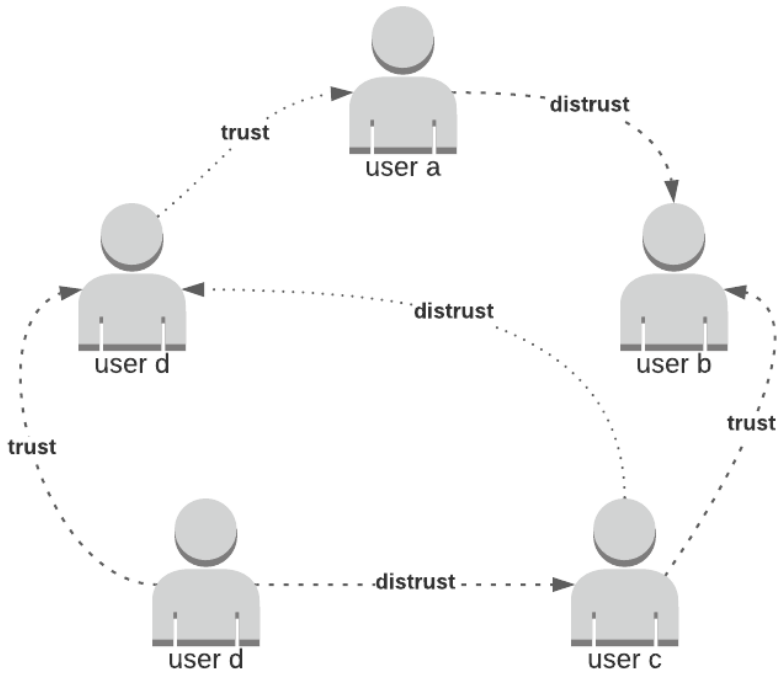

Figure 1.1: An example of a signed explicit trust network

problems, recommender systems employ a variety of auxiliary information including product details of the items, social network of users or external contextual information such as weather or currency rate [2].

Trust is an essential notion in social networks to determine the nature of the relationship between users and to quantify such relationship. It is valuable auxiliary information to improve recommendation performance, and hence it has been employed to overcome the issues of CF mentioned above. It helps to alleviate the data sparsity through enrichment with the ratings of trusted neighbors. Trust is also valid against the cold start problem as the preferences of trusted neighbors or trusted users, in general, can provide a basis for the recommendation.

In the literature, generally, two types of trust are described: *explicit* and *implicit trust*. Explicit trust is obtained through explicit feedback of users for other users. This feedback is the most direct way of interpreting a user's opinion about another user. Figure 1.1 shows an example of a signed explicit trust network where users evaluate

each other by declaring trust or distrust. A well-known example is epinions[3], which is a website of product reviews. It uses a trust system such that users can define their *web of Trust*, reviewers whose reviews and ratings are consistently found to be useful, and their *block list*, reviewers that they consistently find inaccurate or not useful[4]. The web of trust and the block list collection crawled from Epinions web site, namely *epinions data set* has been popularly used as explicit trust data in various studies [3] [4] [5] [6].

There are two types of explicit trust networks: unsigned (network with only positive trust links) and signed (trust network with negative and positive trust links) explicit trust network.

Implicit trust, on the other hand, indirectly provides information about the trust relationship between users, generally based on activities and behaviors of users [7], [3]. Explicit trust information is rarely available, and also it is sparse, just like in the user-item matrix case. Therefore, various studies focused on generating implicit trust by exploiting the resources such as the rating data and social connection of users [8]. For example, in [9], interest similarity is used for inferring trust between two users, whereas in [7], trust propagation over a social network is employed for constructing the trust network of a given user.

1.1 Research Questions, Approach and Contributions

In this work, the following two sub-problems on explicit and implicit trust in the recommendation setting are analyzed:

- What is the relationship between explicit and implicit trust scores, are they replaceable?
- Can the explicit trust in a trust network be modeled?

The first one is examining the compatibility between explicit and implicit trust scores.

[3] https://web.archive.org/web/20090420090156/http://www.epinions.com/help/faq/?show=faq$_w ot$

[4] http://www.trustlet.org/epinions.html

For this analysis, an implicit trust model is proposed. By using the proposed implicit trust model, the matching between implicit and explicit trust scores is analyzed and explicit trust is tried to model by using implicit trust model. This analysis is crucial for understanding the nature of implicit and explicit trust and using them in either complementary ways or as a replacement. The reason why this issue is important is that if we can create explicit trust data implicitly, where the explicit trust data is not available, we can still use its benefits.

The second sub-problem is to construct an explicit trust model to predict missing trust relationships in the trust network. In this way, the data sparsity in explicit trust information could be reduced. There are two types of explicit trust networks: *unsigned* network with only positive links and *signed* trust network with negative and positive links. In an unsigned trust matrix, trust information is explicitly expressed as 1 to denote trust. However, 0 as the trust value may indicate either a neutral or unknown trust relationship.

For this, two different explicit models are used. In the first model, users' rating behavior is exploited for explicit trust modeling. A trust graph is generated in the second approach, and the problem is specified as a link prediction problem. In the graph model, trust value 1 in the matrix denotes a link, whereas trust value 0 shows that there is no edge between the given nodes (i.e., users). It is aimed to predict the missing trust relationships in the user-user trust matrix by constructing an explicit trust model. The proposed approach tried to predict the data and the effect of augmenting the trust matrix is analyzed by using explicit trust inference through trust-based recommendation methods in the literature.

Part of the output of this study was published at The 13th International ACM Conference on Management of Digital EcoSystems (MEDES'21) [10]. The contributions of this study can be summarized as follows:

- To analyze the compatibility of implicit and explicit scores, an implicit trust model is devised, which is adapted from the consistency model for reputation scores of users in [11]. The trust scores of this model are compared with the explicitly presented trust scores.

- The nature of these two scores is not exactly compatible since the implicit trust model generates a single score per user. In contrast, the explicit trust model tells the trust relationship between two users. To overcome this incompatibility, a mapping schema is presented that generates an explicit trust score per user from the explicit trust score between two users.
- An explicit trust model is devised by exploiting the ratings that users give to model explicit trust data. It is considered as a classification problem. This method was used on both signed and unsigned data.
- Finding an explicit trust between two users is considered as an edge prediction problem and a supervised learning model is generated to predict unknown trust values. The effectiveness of an augmented trust network is analyzed through recommendation performance.

1.2 Structure of the Thesis

The rest of the thesis is organized as follows. In Chapter 2, some basic subjects that are necessary to understand the content of the thesis are mentioned. In Chapter 3, related studies in the literature are summarized. The methods employed in this study are presented in Section 4. The experiments and results are presented in Chapter 5. Finally, Chapter 6 concludes the thesis with an overview and future work.

CHAPTER 2

BACKGROUND

2.1 Recommendation Systems

Recommendation systems are the algorithms and methods designed to suggest relevant items to users. The main goal of the recommendation systems is to predict the most likely items that the user is most interested in. There are several critical applications of recommendations systems, such as product recommendations, movie recommendations, content recommendations. These systems use different technologies as given in Figure 2.1, and they can be classified into following groups [12].

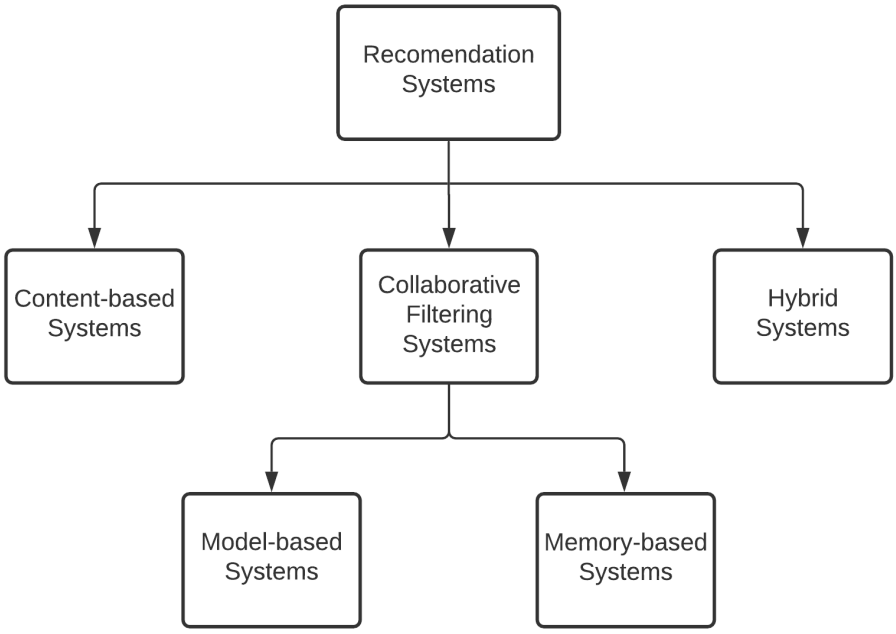

Figure 2.1: Recommendation Systems

- Content-based systems examine the features of the items. Similar items are determined by examining the similarity in their features. A profile is constructed for each user that describes the types of items that interest the user and represents the user's essential characteristics. These systems recommend an item to a user according to the item's features and a profile of the user's preferences [13]. For instance, if a user has watched many comedy movies, the system recommends a movie labeled as having the "comedy" genre.

- Collaborative filtering systems use similarity measures between users or items to recommend items to users. CF is the process of filtering items based on the evaluations from users. These systems are based on the past activities recorded between users and items in order to find similarities. These activities are stored in a user-item matrix. Similar users prefer the items recommended to a user. Collaborative filtering systems bring together the preferences of enormous interconnected parties and filter significant amounts of data [14].

- Model-based systems concern building a model based on the ratings. A data model is built upfront in model-based systems as supervised or unsupervised machine learning methods. Consequently, the training phase and the prediction phase are separated. These systems deliver the advantages of speed and scalability. Such methods include decision trees, Bayes classifiers, regression models, support vector machines, and neural networks [15].

- Memory-based systems use the entire data set to generate a prediction. They try to find users that are similar to the active user. Every user is part of a cluster of users with matching interests. A prediction of preferences on new items for the active user can be generated by specifying the neighbors of the active user [16]. Since these systems use the entire data every time they make a prediction, they are slower than model-based systems.

- Hybrid systems merge different recommendation approaches to achieve better system performance to bypass some constraints and problems of recommendation systems [17]. The argument is that a mixture of algorithms will provide more precise and compelling recommendations than a single method, as another method can overcome the weaknesses of one algorithm [18].

2.2 Trust-Based Recommendation Systems

Trust is an instrumental concept that helps us understand and interpret the relationships between users in social networks, predominantly because it is determined explicitly by users.

The use of this specific notion has emerged as a new way of giving more accurate recommendations. The previous studies show that trust-based recommendation techniques outperform those that only use user similarity or item similarity. Those kinds of systems help users by providing extra information on how trustworthy the user is in an activity. The essential point behind this idea is that users rate each other as well as they rate items [19]. In other words, if a phenomenon in daily life is given as an example, the evaluations made by people whom everyone explicitly trusts are more important to people than the evaluations made by an ordinary person.

2.3 Machine Learning Methods

Several machine learning algorithms were used in the proposed approaches and conducted experiments in this thesis. The methods used are described below.

- Random Forest Classifier consists of a huge amount of individual decision trees. Each tree in the forest makes a class prediction, and the prediction with more votes becomes the classifier's prediction. Random Forest is suitable for large data sets, and it provides higher accuracy via cross-validation. [20]
- Extra Trees Classifier is an ensemble learning method that aggregates the results of multiple decision trees as a forest to classify the result. It can be used for feature selection. Conceptually, it is pretty similar to a Random Forest Classifier. The differences between them are the construction of the decision trees and the execution time. As Extra Trees Classifier randomly chooses the selection of cut points, Random Forest Classifier chooses optimum split. Also, Extra Trees Classifier is much faster. [21]
- Isolation Forest Classifier is a one-class classification algorithm suitable for

anomaly detection. It is one of the efficient ways of performing outlier detection. Isolation Tree explicitly isolates outliers by using binary trees without profiling all samples. It detects outliers by isolating samples by randomly selecting an attribute and then randomly selecting a split value between the minimum and maximum values. [22]

- SVMs are a set of supervised learning methods, and these methods are used for classification, regression, and anomaly detection. They are effective in high dimensional data and also memory efficient. However, if the number of features exceeds the sample size, the model may over-fit. SVMs are expensive computation-wise due to five-fold cross-validation.

- One-class SVM is an unsupervised novelty detection algorithm. It learns a decision function and classifies new samples as similar or different from the trained samples. [23]

- Gaussian Naive Bayes is one of the methods that are a set of supervised learning methods based on Bayes' theorem. The likelihood of the attributes is assumed to be Gaussian. [24]

2.4 Link Prediction in Social Networks

A social network represents the relationships between social actors such as users and groups. It can be visualized as a graph, where nodes represent social entities and edges represent social ties. The interactions and links among these social entities give us a large amount of valuable information to interpret. The retrieved information can be used for understanding the formed relationships better and designing recommendation systems.

As mentioned in [25], link prediction is used to predict missing links in current networks and new links in future networks. Social networks are highly dynamic, and because of that, the nodes and links might appear or disappear in the future. Thus, predicting the unobserved links in current networks is vital for completing the current networks. Link prediction can be used in many different applications. One of

them is recommendation systems. It can help users find new friends and improve the performance of recommending interesting items.

In this study, a social network is created using trust relationships between users. The link prediction method is used to predict the trust relationship between users whose trust relationship is unknown. In this manner, the trust data is strengthened by finding lost links or trust values to increase the accuracy of recommendations.

CHAPTER 3

LITERATURE REVIEW

In the literature, there are several research efforts that focus on the use of the trust model to improve the accuracy of recommendations.

Li et. al. [7] propose an implicit trust recommendation approach (ITRA) that utilizes implicit user information to overcome poor accuracy issues due to cold start and data sparsity. To generate a set of trusted neighbors of a given user, the authors exploit the social network and trust diffusion features in a trust network. After finding the trust neighbor set, trust values are determined by computing the shortest distance between a user and inferred trusted neighbor.

Yang et al. [26] propose TrustMF, a matrix factorization-based method that fuses rating and trust information. TrustMF defines two models: *truster* model which denotes how others will affect user u's preferences and *trustee* model which denotes how user u will affect others' preferences. The main idea of the truster mentioned above and trustee models is to link ratings and trust information.

Wang et al. [27] introduce TeCF, a trust enhanced collaborative filtering method that integrates user-based, item-based, and trust-based techniques to predict unrated items. The conducted experiments show that their approach significantly reduces the effects of data sparsity by making the rating matrix denser.

Guo et al. [28] propose Merge, a trust-based approach that uses explicitly specified social trust information while providing recommendations. The method merges the ratings of a user's trusted neighbors to represent the user's opinion to find similar users.

Htun and Tar [9] mention trust as a solution to cold-start problems in recommender systems by using explicit trust ratings given to users for neighbor formation. They propose a method to derive implicit trust relationships based on the similarity of user's interest due to reliable explicit trust data is rarely available. The authors worked on social bookmarking systems where users annotate the bookmark resources using their tags. Therefore, tags show the user's interest in the resources. They measure trust values between users according to the following similarity measures: tag usage similarity, resource item similarity, and interest similarity on resource items. The resulting trust metric is incorporated into the top-k recommender system even though there is no available explicit trust data. The performance of the proposed approach is reported to outperform traditional CF.

Hu et al. [3] propose the SSL-SVD method to solve cold-start and data sparsity problems in recommendation systems. The method incorporates social trust (explicit trust) and sparse trust (implicit trust) information to give better recommendations. In the experiments, they report that social trust is influenced by many social factors and has a limited effect in improving the accuracy of recommendations.

Chen et al. [29] propose a cold start recommendation method that integrates a user model with trust and distrust for each new user in order to identify other trustworthy users. With that approach, they can identify trustworthy users by analyzing the web of trust of experienced users. The aggregation of trusted users provides valuable suggestions for cold start new users. The proposed method is implemented in two stages. Firstly, a user model is constructed by using a clustering algorithm to group experienced users into clusters. Each cluster is formed with users that have similar item preferences. They construct a web of trust for each cluster and use the PageRank algorithm to find experienced users in the cluster. Similarly, they use distrust networks to find unreliable users. Secondly, the most closely related cluster is identified for a cold start new user to predict an unrated item's possible rating. Previously identified experienced users in the cluster are exploited to recommend new cold-start users. Also, the proposed method identifies implicit trust links between users by exploiting the given ratings and enriches the web of trust. Since many users may be unwilling to give explicit trust information due to privacy concerns.

Guo et al. [30] propose three factored similarity models with the use of social trust for a recommendation that is based on implicit user feedback. A matrix factorization technique is introduced to recover user preferences between rated and unrated items by exploiting similarities between user-to-user and item-to-item. They propose a trust-based recommendation approach for top-N item recommendations due to the importance of social trust relationships between users. As a result of the experiments, they verified the impact of their approach and demonstrated the performance-enhancing effect of using social trust.

As described in the studies, the type of trust information used in the recommendation and how it is incorporated vary; however, overall the use of trust information has a positive effect on recommendation performance. In this study, to further increase this positive effect, another aspect of the use of trust modeling in the recommendation is focused on. The nature of implicit and explicit trust modeling and its compatibility is analyzed.

CHAPTER 4

METHODOLOGY

In this section, the methods for the problems that are mentioned in Chapter 1, the compatibility analysis of explicit and implicit trust models, and generating explicit trust prediction model are described in detail. For convenience, the symbols used in the formulas are listed in Table 4.1.

Table 4.1: The list of symbols

Notation	Explanation
u	User
r	Rating
m	Item
R_u	Set of ratings by user *u*
r_m	Average rating of item *m*
r_u	Average of the ratings given by user *u*
$r_u m$	The rating given by user *u* to item *m*
O_u	Conformity of user *u*
$O_{r_{um}}$	Conformity of rating *r* by user *u* for item *m*
C_u	Consistency of user *u*
s_m	Standard deviation of ratings for item *m*

4.1 Method for Analyzing the Compatibility of Explicit and Implicit Trust Models

Previous studies in the literature have shown that using trust information can improve the quality of predictions of recommendation systems [31]. However, trust data is not available for most of the recommendation systems. This is a situation that limits the use of trust-based approaches, no matter how effective such approaches are. Thus, it is investigated that how compatible implicit and explicit trust scores are to enrich the explicit trust data with the implicit one. This analysis is crucial to be able to understand whether these two models have overlapping or complementary nature.

The overview of the proposed approach is shown in Figure 4.1. First, the implicit trust score model is built using the rating data, and implicit trust scores are generated for each user with this model. Then, explicit trust scores are generated for each user using explicit trust data. Finally, mapping is applied between these two results and compared.

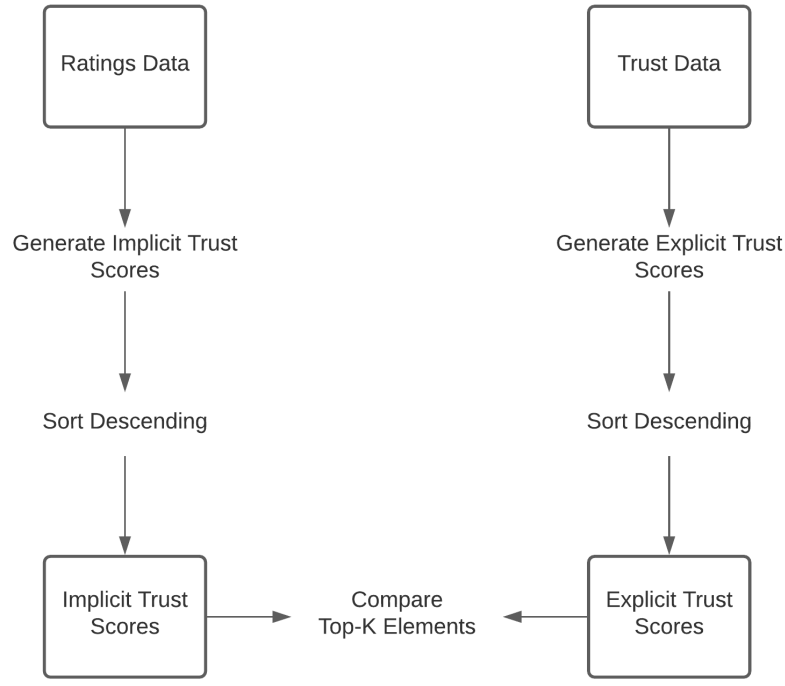

Figure 4.1: Overview of the process of analyzing the compatibility of the implicit and explicit scores

4.1.1 Implicit Trust Score Model

The study by Oh and Kim [11] introduces the concepts of activity, objectivity, and consistency for social media users and defines mathematical equations to determine scores for these features. These scores are used for calculating the reputation scores of users.

In this thesis, these features are adapted for generating the implicit trust scores of users[1] According to the adaptation, users with a high number of ratings above a given threshold are considered as *active* users. The conformity of a rating, $O_{r_{um}}$, is a measure of whether a rating *r* by user *u* on item *m* differs from the average rating on item *m* (denoted as $\overline{r}_m$). s_m denotes the standard deviation of the ratings on item *m*. Conformity of a given rating increases as $O_{r_{um}}$ approaches zero (Eq. 4.1).

$$O_{\text{r}_{\text{um}}} = \mid \frac{r_{\text{um}} - \overline{r}_m}{s_m} \mid \tag{4.1}$$

The conformity of user, O_u, is the average of ratings, $O_{r_{um}}$, by user u. As in the rating conformity, as the value gets closer to zero, the conformity of the user is considered to be higher (Eq. 4.2). Here R_u denotes the number of ratings by user u.

$$O_u = \frac{1}{\mid R_u \mid} \sum O_{r_{um}} \tag{4.2}$$

If the user u behaves similarly to other users in the system, it can be inferred that the user behaved consistently. The consistency of a user, C_u, is defined as the variation in conformity of her/his own evaluations (Eq. 4.3). In the proposed approach, C_u score of a user u is used as the *implicit trust score*.

$$C_u = \frac{1}{\mid R_u \mid} \sum_{r \in R_u} (O_r - O_u)^2 \tag{4.3}$$

[1] A similar adaptation of Oh and Kim's features in [11] for calculation of implicit trust scores for location-based social networks is also presented in Canturk D et al., Trust-aware location recommendation location-based social networks: A graph-based approach (Under review). In this study, these mathematical models are adapted for rating data.

4.1.2 Explicit Score Construction

The implicit trust model generates a trust score per user. On the other hand, explicit trust in the network indicates a trust relationship between two users. To provide compatibility between these two models, a mapping schema is defined that generates an explicit trust score per user from the explicit trust score between two users.

The mapping schema is as follows: Given a user in an *unsigned* trust network, the number of incoming trust edges is determined as the explicit trust score per user. For *signed* networks, the number of incoming edges with weight 1 denotes the trust score per user. Similarly, the number of incoming edges with weight -1 is the *distrust score* of the user. For example, in a given unsigned trust network, if the node of $usera$ has ten incoming edges, this denotes that ten users trust this user. Then the explicit trust score of $usera$ is set as 10.

As an alternative mapping schema, a well-known PageRank algorithm [32] is used. In order to generated trust score per user, PageRank algorithm is applied on the trust network. This scoring also gives the ranking of the users in the trust network, which is used for comparison with implicit trust score rankings. Although there are several other personalized node ranking algorithms proposed for signed networks in the literature [33], in this study, conventional PageRank is used for both unsigned and signed trust networks.

4.1.3 Mapping of Explicit and Implicit Trust Scores

After building the implicit score model and constructing the explicit trust scores, a mapping was applied between them. These scores are sorted separately among themselves. The compatibility of the implicit and explicit scores is analyzed as the overlapping on top-k% items between the sorted implicit and explicit trust scores for users. The analysis results conducted on three data sets are presented and discussed in Section 5.3.

4.2 Explicit Trust Modeling

The purpose of explicit trust modeling methods discussed below is to increase the accuracy of trust-based recommendations by estimating unknown values in the explicit trust matrix. It is done in two different ways. In the first proposed method, explicit trust data is modeled using users' rating behavior. In the other method, an explicit trust network is created, and missing links between users are aimed to be found with link prediction.

4.2.1 Explicit Trust Modeling through User's Rating Behaviour

In this section, a novel explicit modeling approach is proposed. It is aimed to model the explicit trust data by using the rating data in order to generate augmented trust data, as shown in Figure 4.2.

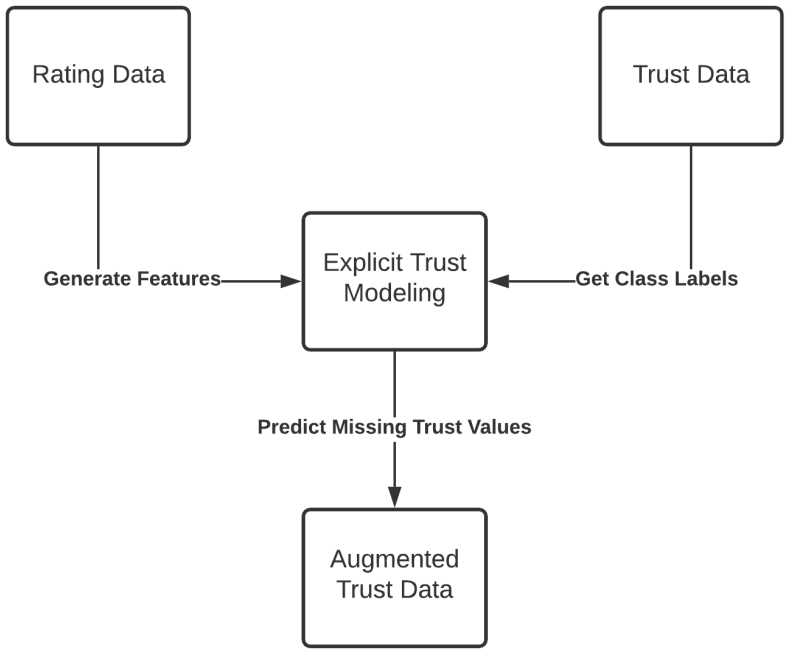

Figure 4.2: Overview of the process of explicit trust modeling through user's rating behaviour

$$isItemLiked = \begin{cases} True, & \text{if } r_{um} \geq r_m \\ False, & \text{otherwise} \end{cases} \tag{4.4}$$

This issue is treated as a classification problem. Therefore, various features are generated for each explicit trust relationship by exploiting the ratings given by the users. While generating those features, liked and disliked items notion is presented. One can see the calculation method as given in Eq. 4.4:

An average rating score is created for each user by looking at the ratings given by that user. Then, by looking at the items that each user gave a rating, the rating given to that item and the average rating score of the user is compared. If the rating given to that item is greater than or equal to the average rating score of the user, it becomes a liked item; otherwise, it becomes a disliked item. The procedure is explained in Algorithm 1.

For instance, suppose a user gives a rating of 1.0, 2.0, 5.0 to movies a, b, and c,

Algorithm 1 Discovering Liked and Disliked Items for Each User

```
U: a set of users
R: a set of ratings
procedure DISCOVERLIKEDANDDISLIKEDITEMS(U, R)
    Build R_u, a list of ratings given by each user u using U and R
    for each user u in R do
        Calculate average rating a_u for u
        for each rating r in R_u do
            if r ≥ a_u then
                rated item is a liked item for user u
            else
                rated item is a disliked item for user u
            end if
        end for
    end for
end procedure
```

respectively. In this case, since the average rating of this user will be 3.0, it can be said that this user disliked movies a and b and liked the movie c. The reason for looking at each user's average here is that each user's rating behavior is not the same. While 4.0 means a good rating for some users, it may not mean the same for others.

Before modeling, seven different features are created for each trust relationship. These created features are listed below.

- The number of the intersection of rated items,
- The number of the intersection of liked items,
- The number of the intersection of disliked items,
- The average of the ratings given by the trustor,
- The number of the ratings given by the trustor,
- The average of the ratings given by the trustee,
- The number of the ratings given by the trustee,

For each user, separate lists are created for the items she rated, liked, and disliked. By observing those lists, the intersection of rated, liked, and disliked items between two users can be found easily, and it can be seen in which features the users differ.

The corresponding explicit trust value is the class label in each trust relationship. However, the value of the class label varies depending on whether the trust network is signed or unsigned. This approach can be divided into two separate subheadings as follows:

- Modeling unsigned explicit trust data by using one-class classification,
- Modeling signed explicit trust data by using binary classification,

For the signed trust network, the values that can be taken are 1 and -1. In this case, the problem can be considered a binary classification problem. However, this situation is slightly different for data with unsigned trust networks. Because there is no distrust

information in such data, every relationship in the network is expressed with 1. In this case, handling the problem had to be changed. The problem is an outlier/novelty detection or a one-class classification problem.

Isolation Forest and One-class SVM algorithms are used for modeling unsigned explicit trust data. For modeling signed explicit trust data, in addition to the previously mentioned algorithms, SVM, Random Forest, and Naive Bayes classifier algorithms are used.

Modeling signed explicit trust data using binary classification is conducted with SVM, Random Forest, and Naive Bayes classifiers. Unlike the technique mentioned above, these binary classifications used 1 for positive trusts and -1 for negative trust values as the class label in the data set. After the modeling procedure is finished, the new trust relationships are predicted with the aid of these models, and the explicit trust network is updated. More successful trust-based recommendations are targeted with the updated explicit trust network. This updated trust data has been given to various trust-based recommendation algorithms. The experiments are detailed in Chapter 5.4.

4.2.2 Trust Prediction Modeling as a Link Prediction Problem

In explicit trust prediction, the problem is considered as an edge prediction task on the directed trust network. More specifically, a supervised learning model is built for the *inference of explicit trust* between users. The process is visualized in Figure 4.3.

Here, in this classification task, the edges correspond to class labels. For unsigned trust networks, an edge denotes a trust relationship, and it is represented with *class label 1*. The rest of the (non-existing) edges in the graph are assumed to correspond to *class label 0*. For signed trust networks, the setting has a slight difference such that the edges are labeled (signed) as either 1 or -1, denoting trust or distrust, respectively. Then we have *class label -1* for edges representing distrust relationship.

To determine a balanced set of training instances, links with zero value as many as the number of trust links are randomly included. On the unsigned trust graph, for each edge, the following features are:

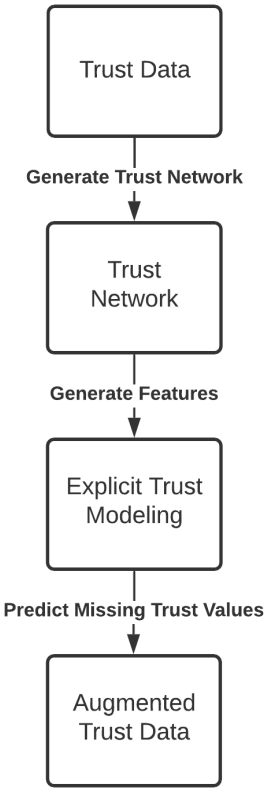

Figure 4.3: Overview of the process of trust prediction modeling as a link prediction problem

- Jaccard similarity for destination and source nodes,
- Cosine similarity for destination and source nodes,
- Katz centrality for destination and source nodes,
- Adar Index [34] for destination and source nodes,
- Number of nodes that trust the source node,
- Number of nodes that trust the destination node,
- Number of nodes that the source node trusts,
- Number of nodes that the destination node trusts,
- Intersection of the nodes that trust source and destination nodes,

- Intersection of the nodes that both source and destination nodes trust,
- Trust back,
- The shortest trust path between nodes.

Hence, on the total, 12 features are extracted from the trust graph. Among the features given above, *Adar Index* is a measure to predict links in a network by using the shared links between two nodes. *Trust back* is a binary field that denotes whether the destination node trusts the source node back or not. For calculating the shortest trust path between nodes, firstly, if they have been already connected, the link between them is deleted. Then, the shortest path between the nodes is computed.

Before constructing the supervised learning model, feature elimination is applied by using Extra-Trees Classifier [35] and filtered out Jaccard similarity, Cosine similarity, Katz centrality, and Adar Index features. As the supervised learning algorithms, Random Forest Classifier and SVM Classifier are employed [35] to construct the explicit trust model.

CHAPTER 5

EXPERIMENTS

5.1 Data Sets and Experiment Environment

The experiments are conducted on MacOS Catalina, Intel(R) Core(TM) i5 CPU @1.4GHz, 16 GB of RAM. The proposed methods are programmed in Python programming language by using scikit-learn [35], and RecQ [36] frameworks.

Table 5.1: Statistics on the datasets

	# of Users	# of Items	# of Ratings
FilmTrust	1,508	2,071	35,497
Epinions (Unsigned)	75,888	29,000	68,1213
Epinions (Signed)	132,492	755,760	13,668,320
Ciao	7,375	105,114	284,086

For the analysis, Epinions (Unsigned) [37], Epinions (Signed) [38], FilmTrust [39], and Ciao [40] data sets are used. All of those data sets are frequently used for recommendation systems analysis, especially in trust-based systems.

The statistical details about the data sets are given in Table 5.1. FilmTrust is a platform that allows its users to evaluate the movies they watch. Epinions is a social networking site where users can share their opinions about various products and express their trust network. Ciao is a product review and online shopping portal that contains trust relationships between users.

5.2 Evaluation Metrics

In this study, the following metrics are used for measuring the accuracy of the predictions.

- Accuracy measures the proportion of correct predictions among the total number of predictions

$$Accuracy = \frac{TruePositive + TrueNegative}{AllPredictions} \tag{5.1}$$

- Precision measures the number of positive class predictions that actually belong to the positive class.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5.2}$$

- Recall measures the number of positive class predictions made from all positive samples.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5.3}$$

- F1-score provides a single score which balances both precision and recall in one metric.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5.4}$$

- Outlier Ratio shows how many samples in the test data are determined as outliers

$$OutlierRatio = \frac{NumberOfOutliers}{NumberOfTestSamples} \tag{5.5}$$

5.3 Implicit and Explicit Trust Score Compatibility Analysis Results

To compare explicit and implicit trust scores, the implicit trust scores are generated as described in Section 4.1 in various configurations by filtering users according to

the number of activities. The users are filtered with respect to the number of ratings given and the values 3, 5 and 10 are used as the minimum rating count threshold. As a result, two rankings of users are obtained with respect to implicit and explicit trust scores, respectively, and measure how well the top k-percent elements match. In the experiments, 10 and 20 values are used as the k value. The results are given in Table 5.2, Table 5.3 and Table 5.4 on the data sets FilmTrust, Epinions and Ciao, respectively.

Table 5.2: Implicit vs. explicit trust model comparison results (FilmTrust)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.003	0.003	0.004
Precision@10%	0.025	0.025	0.038
Recall@20%	0.023	0.023	0.027
Precision@20%	0.106	0.106	0.125

Table 5.3: Implicit vs. explicit trust model comparison results (Epinions)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.000	0.000	0.000
Precision@10%	0.002	0.004	0.020
Recall@20%	0.000	0.000	0.001
Precision@20%	0.006	0.015	0.041

Table 5.4: Implicit vs. explicit trust model comparison results (Ciao)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.006	0.006	0.008
Precision@10%	0.064	0.064	0.080
Recall@20%	0.035	0.035	0.042
Precision@20%	0.173	0.173	0.214

The same comparison is applied between the implicit trust scores and PageRank scores on the explicit trust network. Similarly, the results are given in Table 5.5, Table 5.6 and Table 5.7 on the data sets FilmTrust, Epinions and Ciao, respectively.

Table 5.5: Implicit vs. PageRank model comparison results (FilmTrust)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.003	0.005	0.006
Precision@10%	0.038	0.50	0.062
Recall@20%	0.025	0.024	0.027
Precision@20%	0.138	0.131	0.150

Table 5.6: Implicit vs. PageRank model comparison results (Epinions)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.000	0.000	0.001
Precision@10%	0.004	0.008	0.022
Recall@20%	0.000	0.000	0.001
Precision@20%	0.009	0.019	0.042

As given in the tables, the matching between two rankings is very scarce. However, as the implicit modeling is performed among more active users, it is seen that the precision and recall scores in the amount of match between implicit and explicit scores increased. For the results of the proposed matching schema and the PageRank based scoring, although there are slight differences in the matching scores, both display very similar behavior on the overall for matching with the implicit trust sores. The reason for this difference may be that the trust relationship between users cannot be explained only by evaluating the rating behaviors of the users as in the method suggested here.

Table 5.7: Implicit vs. PageRank model comparison results (Ciao)

	min. 3 ratings	min. 5 ratings	min. 10 rating
Recall@10%	0.006	0.006	0.007
Precision@10%	0.057	0.057	0.070
Recall@20%	0.035	0.035	0.040
Precision@20%	0.174	0.174	0.203

In [3], it is reported that the combination of explicit and implicit trust models increases the accuracy of estimates compared to using them separately. In the same study, it is also noted that the explicit trust relationship is also related to the social ties between users, so this cannot be entirely determined only by ratings made by users. The results are compatible with the findings given in [3].

5.4 Explicit Trust Modeling Results

Experiments on explicit trust modeling methods proposed in Section 4.2 and the results of these experiments are described in Section 5.4.1 and Section 5.4.2.

5.4.1 Explicit Trust Modeling through User's Rating Behaviour Results

Before creating the features to model the explicit trust data implicitly, data exploration is performed to see which features needed to be created. Firstly, a signed trust network is used to understand best the concept of liked and disliked items; since there is no distrust in the unsigned data set, the contrast here cannot be fully seen in the data. When we look at Table 5.8, the values in positive and negative trust relationships are calculated separately for each feature to be created.

Table 5.8: The Mean of the Generated Features (Signed Epinions)

	Positive Trust	**Negative Trust**
# of the intersection of rated items	98.807	61.179
# of the intersection of liked items	83.241	49.783
# of the intersection of disliked items	5.607	3.819
avg of the ratings given by the trustor	4.637	4.417
# of the ratings given by the trustor	1310.380	6482.471
avg of the ratings given by the trustee	4.605	4.367
# of the ratings given by the trustee	5947.152	2434.528

From here, significant findings of users in positive trust and negative trust relationships can be obtained. Users in a positive trust relationship rated more common items

on average than those in a negative relationship. In addition, the positive trust relationship correlates with the number of common favorite items by looking at these results. Although the negative trust items appear to be less than the positive trust items in the number of common disliked items, after dividing it by the number of items with a common rating and normalizing the value, the number of common disliked items is also correlated with a negative trust relationship.

In addition, when looking at Table 5.8, it is seen that the total number of ratings given by people who give negative trust is higher than the total number of ratings given by people who give positive trust. It can be interpreted that more active users, who give higher ratings, are also more selective and evaluate other users according to them.

Table 5.9: The Mean of the Generated Features (Unsigned Data Sets)

	Filmtrust	**Epinions**	**Ciao**
# of the intersection of rated items	9.079	1.194	2.018
# of the intersection of liked items	3.287	0.481	0.761
# of the intersection of disliked items	1.818	0.292	0.397
avg of the ratings given by the trustor	3.033	4.064	4.174
# of the ratings given by the trustor	39.865	69.332	150.835
avg of the ratings given by the trustee	3.041	4.015	4.209
# of the ratings given by the trustee	38.670	108.083	83.771

Later, similar data exploration is also done in the unsigned data sets. The summary of that is shown in Table 5.9. According to these results, the number of common liked items in trust relationships established in all three unsigned trust data sets is higher than that of common disliked items. It shows that liked items and established trust relationships correlate in unsigned trust datasets as well as in signed trust datasets.

The proposed method is first tested on unsigned data. For this, a feature set is created for each trust relationship. The data created with this feature set is modeled differently with Isolation Forest and One-class SVM. Afterward, user pairs were selected randomly, and related features were generated again. While creating these user pairs, care was taken that this pair does not exist in the existing trust network. The models created later were fed with these test data, and outlier ratio scores were checked.

Table 5.10: Outlier Ratio of the Unsigned Data Sets

	Isolation Forest	**One-class SVM**
FilmTrust	0.134	0.883
Epinions	0.098	0.512
Ciao	0.111	0.538

The scores that appear here show how much the test data are included in the created classes or how outlier they are. Table 5.10 shows the results.

According to the results obtained, the Isolation Forest model shows that most randomly generated trust relationships are identified in the relevant class. It is not an expected result. On the other hand, it is shown that the one-class SVM model outliers nearly 50 percent of the Epinions and Ciao data sets. Here, too, we cannot talk about a very successful model. Only FilmTrust data shows an outlier ratio close to expected, but it could not provide these results in every data set. This difference may be due to the difference in user relations in the data sets. Figure 5.9 shows that the number of the intersection of rated items by users in the Filmtrust data set is significantly higher. Considering that the Filmtrust data set is smaller than the others, it can be said that the relations between users seem more potent than the others.

Table 5.11: Outlier Ratio of the Signed Epinions Data Set

	Isolation Forest	**One-class SVM**
Trained with positive, fed with positive	0.076	0.495
Trained with negative, fed with negative	0.077	0.501
Trained with positive, fed with negative	0.267	0.679
Trained with negative, fed with positive	0.114	0.526

The outlier/novelty detection methods mentioned above have also been tested on Epinions data, a signed trust network. Here, after the features are created for each trust relationship, both the Isolation Forest classifier and One-class SVM classifier are created separately. For both methods, estimations were made with the following

models:

- Making predictions by feeding the model created by training with positive trust data with trust data known to be positive
- Making predictions by feeding the model created by training with positive trust data with trust data known to be negative
- Making predictions by feeding the model created by training with negative trust data with trust data known to be negative
- Making predictions by feeding the model created by training with negative trust data with trust data known to be positive

The aim here is to see how much positive and negative trust relationships differ using the created features and outlier/novelty detection methods or whether positive and negative trust relationships can be modeled consistently within themselves. Table 5.11 shows the results of the relevant experiment. Based on the results here, one can say that the One-class SVM model does not perform well in distinguishing between both negative and positive trusts. It can be said that the Isolation Forest model successfully models both positive and negative trust data within itself. However, it cannot be said that it has the same success when distinguishing between positive and negative trust. It may be because the features created before modeling are not very suitable for the outlier/novelty detection method. Because when looking at Table 5.8, it is seen that although there are points where positive and negative trust differ, the users who have established these two relationships are also active users who have interacted with each other. For this reason, outlier/novelty detection models could not provide a satisfying prediction.

Table 5.12: The Performance of Supervised Learning Models (Signed Epinions)

	Precision	**Recall**	**F1-Score**
SVM	0.780	0.743	0.747
Random Forest	0.873	0.865	0.868
Naive Bayes	0.744	0.723	0.726

Signed Epinions data is modeled using multi-class classification methods besides outlier/novelty detection methods. SVM, Random Forest, and Naive Bayes classifiers are implemented here. As class labels, 1 for positive trusts and -1 for negative trust values were used. After the features were created, the data was separated as train and test data at a ratio of 0.8 and 0.2. Table 5.12 shows the performances of the models. When these results are examined, multi-class classification methods give more successful results than outlier/novelty detection methods. The reason is that both negative and positive trust relationships are used in the training phase. In addition, it can be seen that the Random Forest method gives the best results among these three methods.

Table 5.13: The effect of Modeled Explicit Trust Inference with SBPR, SREE and TBPR algorithms (Signed Epinions)

	w/o Trust Inference		**with Trust Inference**	
	Precision	**Recall**	**Precision**	**Recall**
SBPR [41]	0.005	0.016	0.009	0.027
SREE [42]	0.002	0.002	0.003	0.001
TBPR [43]	0.001	0.004	0.002	0.008

Trust data was strengthened by predicting new trust relationships after multi-class classification modeling. While selecting the new trust relationships to be predicted here, care was taken not to select the same relationships that were previously found in the trust network. In addition, to eliminate the possibility of users who have no relationship with each other, users who gave at least 1 rating to the same item were selected when choosing new trust relationships to be predicted. SBPR, SREE, and TBPR algorithms were used to compare the performance results of the trust data fed with the results and the old trust data. The resulting results are shown in Table 5.13. Precision and recall values were calculated by looking at the top-10 item rankings in each algorithm. Judging by the results, performance gains have been observed in almost every case where augmented trust data is used. It can be said that the trust inference method is successful.

5.4.2 Trust Prediction Modeling as a Link Prediction Problem Results

In explicit trust modeling analysis, the basic idea is to construct a trust prediction model and to reduce data sparsity by filling in the trust matrix by using the predictions of the explicit trust model. In other words, a prediction is generated for the edge weights, which are 0 in the original network. A supervised model was created using the features specified in Section 4.2.2. The accuracy performance of the models generated with Random Forest and SVM classifiers for explicit trust prediction is given in Table 5.14, and Table 5.15, respectively.

It can be said that the resulting explicit trust models can predict classes at a satisfactory rate based on the results. From here, it can be interpreted that an augmented matrix can be created by filling unknown trust links between users with this modeling technique. It is also observed that the highest prediction accuracy is obtained on Ciao data set, whereas the performance of the prediction on unsigned Epinions data set is better than those on FilmTrust. Also, by looking at the results, the Random Forest model performs better than the SVM model. For this reason, the output of the Random Forest model is used when creating the augmented trust.

Table 5.14: Accuracy results for explicit trust prediction (Random Forest)

Data Sets	**Accuracy**	**Precision**	**Recall**	**F1-Score**
FilmTrust	0.675	0.952	0.639	0.765
Epinions	0.930	0.979	0.879	0.926
Ciao	0.940	0.969	0.904	0.935

Table 5.15: Accuracy results for explicit trust prediction (SVM)

Data Sets	**Accuracy**	**Precision**	**Recall**	**F1-Score**
FilmTrust	0.819	0.891	0.728	0.801
Epinions	0.961	0.870	0.916	0.892
Ciao	0.901	0.955	0.843	0.895

With the aforementioned supervised learning model, some unknown trust links were inferred, and these were added to the existing trust network as new trust links. To

Table 5.16: The effect of Explicit Trust Inference on Recommendation with SBPR, SREE and TBPR algorithms (FilmTrust)

	w/o Trust Inference		**with Trust Inference**	
	Precision	**Recall**	**Precision**	**Recall**
SBPR [41]	0.301	0.537	0.303	0.549
SREE [42]	0.310	0.402	0.306	0.397
TBPR [43]	0.294	0.472	0.287	0.471

Table 5.17: The effect of Explicit Trust Inference on Recommendation with SBPR, SREE and TBPR algorithms (Epinions)

	w/o Trust Inference		**with Trust Inference**	
	Precision	**Recall**	**Precision**	**Recall**
SBPR [41]	0.007	0.017	0.008	0.018
SREE [42]	0.007	0.013	0.007	0.013
TBPR [43]	0.001	0.003	0.002	0.004

Table 5.18: The effect of Explicit Trust Inference on Recommendation with SBPR, SREE and TBPR algorithms (Ciao)

	w/o Trust Inference		**with Trust Inference**	
	Precision	**Recall**	**Precision**	**Recall**
SBPR [41]	0.015	0.022	0.016	0.023
SREE [42]	0.004	0.003	0.005	0.004
TBPR [43]	0.003	0.004	0.004	0.005

analyze the effect of explicit trust inference, the performance of the augmented trust matrix is compared against the original one by using a set of trust-based recommendation algorithms, SBPR, SREE, and TBPR on FilmTrust, unsigned Epinions, and Ciao data sets, given in Table 5.16, Table 5.17 and Table 5.18, respectively. The purpose of doing this is to see if newly added trust links will improve the performance of the following trust-based recommendation algorithms.

- SBPR [41] is a ranking-based model that exploits social connections between users to build better prediction models. The model is based on the idea that users tend to give higher rankings to items that their connections prefer.
- SREE [42] is a social recommendation approach based on Euclidean Space. The idea behind this algorithm is to place users and items in a unified Euclidean space where users are close to both the items they want and their social friends.
- TBPR [43] classifies strong and weak ties in a social network and learns latent feature vectors for all users and items. It is an extension of the Bayesian Personalized Ranking model.

The results indicate a minor increase in the recommendation performance with the inclusion of explicit trust inference. It may be due to that the trust values to be predicted do not have a significant change. Hence, the results hint for possibility for improvement by carefully selecting the trust relationships to be predicted and updated.

CHAPTER 6

CONCLUSION

In this work, the trust modeling is worked within recommendation context. More specifically, two sub-problems have been identified:

- Inferring the implicit trust information by examining the past user behaviors and analyzing the compatibility of implicit and explicit trust scores
- Building an explicit trust model and predicting the missing explicit trust information

For the first sub-problem, an implicit model is created. The implicit trust information is inferred by defining notions of conformity and consistency. After extracting implicit trust information, the compatibility of implicit and explicit trust scores are analyzed. The analysis of the approach reveals that there is no clear correlation between the implicit and explicit scores. The experiments analyzed how well the implicit and explicit scores matched at the top-20% and top-10%. In experiments with various parameters, precision and recall scores are generally below 0.1. In addition, as the modeling is performed among more active users, it is seen that the precision score in the amount of match between implicit and explicit scores increased above 0.1. The results hint at the effect of social ties in the trust relationship, and hence implicit trust model cannot replace explicit trust but is somewhat helpful as a piece of complementary information.

For the second sub-problem, two different explicit models exhibit two different approaches. In the first approach, explicit trusts are modeled by generating a set of new features containing liked and disliked items. While creating these features, users'

rating behavior is used. Here, separate experiments are performed with signed and unsigned trust data sets. By feeding opposite value trusts to models created with negative and positive trust, the calculated outlier ratio would be expected to be considerably higher than 0.5. Expected performance could not be achieved in models created with one-class classification. However, in the experiments made with the augmented trust data created with the multi-class classification model, the precision and recall values in the SBPR and TBPR algorithms are boosted approximately two times. Here, it is seen that the trust-based recommendation accuracy can be increased by modeling the ratings and explicit trusts given by the users together.

An explicit trust network is formed in the other approach, and missing links are predicted. After generating the augmented trust data, the accuracy of the newly estimated trust data is tested using various trust-based algorithms. The results show that the augmented trust matrix leads to improvement in performance, but the effect is not very high (Section 5.4.2). This can be because the trust values to be predicted selected randomly, and the predictions do not significantly change the edge labels. Hence, with a more detailed mechanism for selecting the unknown trust relationships to be predicted, the performance could be further improved.

For future work, the explicit trust model created using users' rating behavior, can be modified to be used for data that does not contain explicit trust data. Since the trust data is primarily unavailable in data sets, by doing so, we might get the advantage of trust-based recommendations on most of the available data sets. The positive effects of strengthening trust relationships are seen in the experiments.

In addition, if the links accepted as -1 in the explicit link prediction model are selected by using a particular method instead of random, the performance of that model may also be improved. It has been observed that the effect of generating a trust and creating a new augmented matrix for users with no ties to each other is limited.

The scope of the topics in this thesis can be expanded by conducting studies to detect and prevent attacks that can manipulate trust-based systems and affect users' trust scores. This issue may be similar to manipulations with fake users currently created on social media platforms.

Another future work direction is generating different implicit trust modeling and elaborating on their compatibility with explicit trust scores. Studies can also be done on an implicit model that produces a distrust score.

REFERENCES

[1] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, p. 5–53, Jan. 2004.

[2] D.-K. Chae, J. Kim, D. H. Chau, and S.-W. Kim, *AR-CF: Augmenting Virtual Users and Items in Collaborative Filtering for Addressing Cold-Start Problems*, p. 1251–1260. New York, NY, USA: Association for Computing Machinery, 2020.

[3] Z. Hu, G. Xu, X. Zheng, J. Liu, Z. Li, Q. Z. Sheng, W. Lian, and H. Xian, "Ssl-svd: Semi-supervised learning–based sparse trust recommendation," *ACM Trans. Internet Technol.*, vol. 20, Jan. 2020.

[4] J. Khan and S. Lee, "Implicit user trust modeling based on user attributes and behavior in online social networks," *IEEE Access*, vol. 7, pp. 142826–142842, 2019.

[5] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, (New York, NY, USA), p. 135–142, Association for Computing Machinery, 2010.

[6] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, *Collaborative User Network Embedding for Social Recommender Systems*, pp. 381–389.

[7] Y. Li, J. Liu, J. Ren, and Y. Chang, "A novel implicit trust recommendation approach for rating prediction," *IEEE Access*, vol. 8, pp. 98305–98315, 2020.

[8] G. Guo, J. Zhang, D. Thalmann, and N. Yorke-Smith, "Etaf: An extended trust antecedents framework for trust prediction," in *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014)*, pp. 540–547, 2014.

[9] Z. Htun and P. P. Tar, "A trust-aware recommender system based on implicit trust extraction," *International Journal of Innovations in Engineering and Technology (IJIET) Technology(IJIET)*, vol. 2, no. 1, pp. 271–276, 2013.

[10] M. U. Demirci and P. Karagoz, *Trust Modeling in Recommendation: Explicit and Implicit Trust Model Compatibility and Explicit Trust Prediction*, p. 8–14. New York, NY, USA: Association for Computing Machinery, 2021.

[11] H.-K. Oh and S.-W. Kim, "Identifying and exploiting trustable users with robust features in online rating systems.," *TIIS*, vol. 11, no. 4, pp. 2171–2195, 2017.

[12] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*. USA: Cambridge University Press, 2011.

[13] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, 2007.

[14] J. B. Schafer, D. Frankowski, J. L. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The Adaptive Web*, 2007.

[15] C. C. Aggarwal, "Recommender systems: The textbook," 2016.

[16] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. in Artif. Intell.*, vol. 2009, jan 2009.

[17] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modeling and User-Adapted Interaction*, vol. 12, 11 2002.

[18] F. Isinkaye, Y. Folajimi, and B. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, 2015.

[19] M. G. Ozsoy and F. Polat, "Trust based recommendation systems," in *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pp. 1267–1274, 2013.

[20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 10 2001.

[21] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, pp. 3–42, 04 2006.

[22] F. T. Liu, K. Ting, and Z.-H. Zhou, "Isolation forest," pp. 413 – 422, 01 2009.

[23] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 07 2001.

[24] H. Zhang, "The optimality of naive bayes," in *FLAIRS Conference*, 2004.

[25] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Science China Information Sciences*, vol. 58, 11 2014.

[26] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[27] F. Wang, W. Zhong, X. Xu, W. Rafique, Z. Zhou, and L. Qi, "Privacy-aware cold-start recommendation based on collaborative filtering and enhanced trust," in *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 655–662, 2020.

[28] G. Guo, J. Zhang, and D. Thalmann, "Merging trust in collaborative filtering to alleviate data sparsity and cold start," *Knowledge-Based Systems*, vol. 57, pp. 57–68, 2014.

[29] C. C. Chen, Y.-H. Wan, M.-C. Chung, and Y.-C. Sun, "An effective recommendation method for cold start new users using trust and distrust networks," *Information Sciences*, vol. 224, pp. 19–36, 2013.

[30] G. Guo, J. Zhang, F. Zhu, and X. Wang, "Factored similarity models with social trust for top-n item recommendation," *Knowledge-Based Systems*, vol. 122, pp. 17–25, 2017.

[31] A. Zahir, Y. Yuan, and K. Moniz, "Agreereltrust—a simple implicit trust inference model for memory-based collaborative filtering recommendation systems," *Electronics*, vol. 8, no. 4, 2019.

[32] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[33] W. Lee, Y.-C. Lee, D. Lee, and S.-W. Kim, *Look Before You Leap: Confirming Edge Signs in Random Walk with Restart for Personalized Node Ranking in Signed Networks*, p. 143–152. New York, NY, USA: Association for Computing Machinery, 2021.

[34] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social networks*, vol. 25, no. 3, pp. 211–230, 2003.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[36] J. Yu, M. Gao, H. Yin, J. Li, C. Gao, and Q. Wang, "Generating reliable friends via adversarial training to improve social recommendation," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 768–777, 2019.

[37] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in *International semantic Web conference*, pp. 351–368, 2003.

[38] M. R. Hamedani, I. Ali, J. Hong, and S.-W. Kim, "Trustrec: An effective approach to exploit implicit trust and distrust relationships along with explicitones for accurate recommendations," *Comput. Sci. Inf. Syst.*, vol. 18, pp. 93–114, 2021.

[39] J. Golbeck, J. Hendler, *et al.*, "Filmtrust: Movie recommendations using trust in web-based social networks," in *Proceedings of the IEEE Consumer communications and networking conference*, vol. 96, pp. 282–286, 2006.

[40] H. L. Jiliang Tang, Huiji Gao and A. D. Sarma, "etrust: Understanding trust evolution in an online world," in *the Eighteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012.

[41] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pp. 261–270, 2014.

[42] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, and T. Dou, "Social recommendation using euclidean embedding," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 589–595, 2017.

[43] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 5–14, 2016.