IMPROVING ACCURACY OF RECOMMENDER SYSTEMS THROUGH TRIADIC CLOSURE

Panagiota TSELENTI

Thesis submitted in partial fulfilment of the requirements of

Kingston University

for the degree of Doctor of Philosophy

March 2017

Abstract

The exponential growth of social media services led to the information overload problem which information filtering and recommender systems deal by exploiting various techniques. One popular technique for making recommendations is based on trust statements between users in a social network. Yet explicit trust statements are usually very sparse leading to the need for expanding the trust networks by inferring new trust relationships. Existing methods exploit the propagation property of trust to expand the existing trust networks; however, their performance is strongly affected by the density of the trust network. Nevertheless, the utilisation of existing trust networks can model the users' relationships, enabling the inference of new connections. The current study advances the existing methods and techniques on developing a trust-based recommender system proposing a novel method to infer trust relationships and to achieve a fully-expanded trust network. In other words, the current study proposes a novel, effective and efficient approach to deal with the information overload by expanding existing trust networks so as to increase accuracy in recommendation systems.

More specifically, this study proposes a novel method to infer trust relationships, called TriadicClosure. The method is based on the homophily phenomenon of social networks and, more specifically, on the triadic closure mechanism, which is a fundamental mechanism of link formation in social networks via which communities emerge naturally, especially when the network is very sparse. Additionally, a method called JaccardCoefficient is proposed to calculate the trust weight of the inferred relationships based on the Jaccard Cofficient similarity measure. Both the proposed methods exploit structural information of the trust graph to infer and calculate the trust value.

Experimental results on real-world datasets demonstrate that the TriadicClosure method outperforms the existing state-of-the-art methods by substantially improving prediction accuracy and coverage of recommendations. Moreover, the method improves the performance of the examined state-of-the-art methods in terms of accuracy and coverage when combined with them. On the other hand, the JaccardCoefficient method for calculating the weight of the inferred trust relationships did not produce stable results, with the majority showing negative impact on the performance, for both accuracy and coverage.

Acknowledgements

First of all, I want to thank my caring parents to which this study is dedicated, for being always supportive, patient and for giving me unlimited encouragement and love. Without them I wouldn’t be what I am.

I am grateful to my supervisor Professor Kostas Danas for his support all these years, his kindness, and his availability whenever I needed. His guidance on planning, designing projects but also on defining and tackle new problems was important during this study.

Then I want to thank my advisor Dimitris Kardaras for his critical guidance and insights when I was stuck. He was the person that inspired me to research recommender systems and helped me start and put me in touch with my supervisor.

I also want to thank Irene for helping when I stuck with the experiments but also Maria, Thanasis and Rallou for their comments on this thesis. I thank also the head of my office Maria for her kindness, understanding, and flexibility providing me the time I needed to finish this study.

Finally, I want to thank my best friend Olia for encouraging me all these years as also for her patience, her support, and her help during a critical period of my life.

Contents

Abstract .... **i**

Acknowledgements .... **ii**

Contents .... **iii**

List of figures .... **v**

List of tables .... **vi**

Glossary .... **viii**

Chapter 1 Introduction **1**

1.1 Motivation-Research problem *1*
1.2 Aim *4*
1.3 Objectives *4*
1.4 Research hypotheses *4*
1.5 Research questions *5*
1.6 Contributions *5*
1.7 Publications *5*
1.8 Thesis overview *6*

Chapter 2 Research methodology **8**

2.1 Problem investigation *10*
2.2 Solution design *10*
2.3 Design validation *11*
2.4 Choice of solution *11*
2.5 Implementation description *11*
2.5.1 Evaluation measures 12
2.5.2 Datasets 13
2.5.3 Experimental design 14
2.6 Implementation evaluation *16*
2.7 Tools and technologies *17*

Chapter 3 Recommender Systems **18**

3.1 How do they work? *20*
3.2 Recommendation approaches *22*
3.2.1 Similarity measures 24
3.3 Content-based recommendations *26*
3.4 Collaborative Filtering recommendations *29*
3.4.1 Memory-based collaborative filtering 32
3.4.2 Model-based collaborative filtering 34
3.5 Other methods *36*
3.5.1 Hybrid approaches 36
3.5.2 Community-based 38
3.6 Evaluating Recommender Systems *39*
3.6.1 Designing the evaluation experiments 39
3.6.1.1 Offline experiments 40
3.6.1.2 Live user experiments 42
3.6.2 Accuracy metrics 43
3.6.2.1 Predictive Accuracy Metrics 44
3.6.2.2 Classification Accuracy Metrics 46
3.6.2.3 Rank Accuracy Metrics 47
3.6.3 Other measures 48

3.7 *Conclusion* ... *51*

Chapter 4 Trust-based Recommender Systems ... 53
4.1 *Definition of Trust* ... *54*
4.2 *Computational properties of trust* ... *58*
4.3 *Trust inference* ... *60*
4.3.1 The small-world phenomenon ... 61
4.3.2 Trust prediction as a link prediction problem ... 63
4.3.3 Propagation ... 65
4.3.4 Aggregation ... 68
4.4 *Trust metrics* ... *70*
4.4.1 Local trust metrics ... 72
4.4.2 Global trust metrics ... 72
4.5 *Major algorithms for trust inference* ... *73*
4.5.1 Advogato ... 73
4.5.2 EigenTrust ... 76
4.5.3 Appleseed ... 76
4.5.4 TidalTrust ... 78
4.5.5 MoleTrust ... 83
4.6 *Conclusion* ... *85*

Chapter 5 The need for a new system ... 86
5.1 *Limitations of current systems* ... *86*
5.2 *Classification of trust models* ... *90*
5.3 *Comparison of Graph-Based Recommendation Models* ... *90*
5.3.1 Comparison criteria ... 92
5.4 *Conclusion* ... *94*

Chapter 6 A novel method to infer trust ... 96
6.1 *The homophily phenomenon* ... *97*
6.2 *Triadic closure* ... *98*
6.3 *Inferring trust with triadic closure* ... *100*
6.4 *Calculating trust with JaccardCoefficient* ... *102*
6.5 *Incorporating TriadicClosure in the recommendation process* ... *103*
6.5.1 An example with synthetic data ... 104
6.6 *Conclusion* ... *107*

Chapter 7 Experimental evaluation ... 109
7.1 *Experimental design* ... *109*
7.2 *Evaluation results and analysis* ... *110*
7.3 *Summary and conclusion* ... *130*

Chapter 8 Discussion, future work and conclusions ... 131
8.1 *Discussion* ... *131*
8.2 *Future work* ... *139*
8.3 *Conclusions* ... *140*

Appendix A Java code ... 143

References ... 147

List of figures

Figure 2-1 Steps followed to solve the research problem 9
Figure 3-1 Inputs and output of a recommender system 21
Figure 3-2 Evaluation metrics for recommender systems 50
Figure 3-3 Recommendation approaches 52
Figure 4-1 A typical trust network 57
Figure 4-2 Process to infer trust 61
Figure 4-3 An example of direct propagation 65
Figure 4-4 Atomic Propagation examples 67
Figure 4-5 Aggregation example 69
Figure 4-6 The three conceptual steps in Advogato trust graph conversion 74
Figure 4-7 An example of direct trust network 79
Figure 6-1 The triadic closure effect in an undirected network 99
Figure 6-2 The triadic closure effect on a directed trust network 101
Figure 6-3 Incorporating TriadicClosure in the recommendation process 103
Figure 6-4 Example trust network 105
Figure 6-5 Updated trust network of Figure 6-4 after TriadicClosure and propagation 106
Figure 7-1 Accuracy performance 112
Figure 7-2 Coverage performance 112

List of tables

Table 2-1 Statistics of the two datasets ... 14
Table 2-2 Statistics for different views in Filmtrust ... 16
Table 2-3 Statistics for different views in Epinions ... 16
Table 3-1: Real-life recommender systems ... 19
Table 3-2: An example of user-item ratings matrix ... 21
Table 3-3 An example of user-item ratings matrix with averages of ratings ... 33
Table 3-4 User-user correlations ... 34
Table 3-5 Possible results of the usage of a prediction ... 46
Table 3-6 Comparison of content-based and collaborative filtering methods ... 51
Table 4-1 Atomic propagations according to Guha et al. (2004) ... 67
Table 4-2 A sample user/item ratings table ... 81
Table 4-3 Major algorithms for trust inference ... 85
Table 5-1 Classification of trust models ... 91
Table 5-2 Comparison of main literature in graph-based trust recommender systems ... 93
Table 6-1 User-item ratings matrix ... 105
Table 6-2 User-user (adjacency) matrix of the trust network of Figure 6-4 ... 106
Table 6-3 Updated user-user trust matrix with inferred trust relationships ... 106
Table 6-4 Trust matrix with JaccardCoefficient values of the inferred relationships ... 106
Table 7-1 First-step evaluation results on Filmtrust dataset ... 111
Table 7-2 First-step evaluation results on Epinions dataset ... 111
Table 7-3 Evaluation results for different propagation lengths on Filmtrust dataset ... 113
Table 7-4 Evaluation results for different propagation lengths on Epinions dataset ... 114
Table 7-5 Impact of TriadicClosure on TidalTrust and MoleTrust ... 114
Table 7-6 Performance of JaccardCoefficient combined with other methods on Filmtrust dataset ... 116
Table 7-7 Performance of JaccardCoefficient combined with other methods on Epinions dataset ... 116
Table 7-8 Impact of JaccardCoefficient on all algorithms ... 117
Table 7-9 Performance of all methods for cold-start users on the two datasets ... 120
Table 7-10 Impact of TC and JC on all methods for cold-start users ... 121

Table 7-11 Performance of all methods for heavy raters on the two datasets 122
Table 7-12 Impact of TC and JC on all methods for heavy raters.. 123
Table 7-13 Performance of all methods for grey-sheep users on the two datasets............. 124
Table 7-14 Impact of TC and JC on all methods for grey-sheep users 125
Table 7-15 Performance of all methods for controversial items on the two datasets 126
Table 7-16 Impact of TC and JC on all methods for controversial items............................... 127
Table 7-17 Performance of all methods for niche items on the two datasets...................... 128
Table 7-18 Impact of TC and JC on all methods for niche items 129

Glossary

Accuracy. The closeness of the prediction of a recommender system to the needs of its users. It measures the effectiveness of the recommendations.

Collaborative filtering. Recommendations based on correlations between users formed by the similarities of the opinions and preferences of users.

Coverage. Prediction coverage refers to the percentage of items for which the system can provide predictions.

Homophily. Refers to the tendency of humans in a social environment to form more associations with similar parties than with dissimilar ones. Some of the definitions for homophily are: *"Similarity breeds connection"*, *"birds of a feather flock together"*, *"people love those who are like themselves".*

Information Overload. It is the inability of the internet user to cope with and manage all the available information in an efficient way.

Propagation. In Social Network Analysis, propagation phenomenon is used to examine how information, diseases or rumours and fads spread across a social network. Through propagation trust can be inferred in a social network, since trust information can be propagated and create trust chains

Recommender system. An information system that make predictions for its users by exploiting information about their tastes, preferences, and needs.

Sparsity. Users do not usually rate enough items or/and other users leading to incomplete modelling of the user due to insufficient information.

Triadic Closure. It is a fundamental mechanism of link formation in social networks which can be perceived as triads tending to close-up. It is based on the phenomenon in social communities that two strangers who possess a mutual friend will tend to become friends in the future.

Trust. In this study is defined as *"relationship between two agents namely the trustor and the trustee where the trustor trusts the trustee in a specific context".*

Trust inference. A mechanism via which a trust relation can be established in a social network between two nodes not being yet connected. This mechanism is implemented through a trust inference algorithm or in other words a 'trust metric', recommending an unknown trust value from one user to another

Trust metric. An approach to calculate and predict trust links between users.

Trust network. A social network on which users state explicitly their trust belief about another user.

Trust-based recommender systems. Recommender systems that use explicit trust relationships between users instead of other measures to calculate the similarity between users.

Chapter 1

Introduction

1.1 Motivation-Research problem

During recent years, the rapid evolution in ICT and mobile technology has brought tremendous change in all sectors of modern society. Since the Internet became widely known, information traffic has radically increased whereby knowledge was distributed. One step further than the first Web and its 'static' information was Web 2.0, which as soon as it emerged, saw an increasing interest in information sharing. The advent of Web 2.0 technologies, along with the hardware development for faster connections, boosted the user contribution on the Internet and content sharing became familiar to every user, not just experts. Web 2.0 applications allow millions of users to publish and edit content as well as to share and tag data in an uncontrolled way and, thus, Web 2.0 technologies have radically transformed not only the way that users interact with information, but also the information available and its volume.

Hence, the explosive growth of information has led to the 'information overload' problem - that is, the inability to cope with and manage all the available information in an efficient way. As Ricci *et al*. (2011) stated, this means that traditional information retrieval systems face tremendous difficulties in retrieving information from 'a mess'. One approach to deal with the information overload problem is to adapt the content of a web page according to the needs and special characteristics of the user; viz. by exploiting personalisation techniques.

At the same time, all this constantly growing information, as well as the advent of new businesses and services, led users to a labyrinth of choices, making the final decision difficult and at many times with limited confidence (Schwartz, 2004). Difficulties in the decision-making process are usually increased due to the user's limited knowledge about a topic, or the time spent on dealing with the volume of the available information. This process is even more complicated when there are too many alternatives, intensified by information overload, as it then becomes too time-consuming to acquire deep knowledge of all the different alternatives. Considering the case of a travel plan, the task is even more complicated as the user needs to find information on different topics, regarding accommodation, attractions, destinations, etc. A typical solution to this, in real life, is to seek advice and suggestions from friends and/or experts. In practice, users need information filtering and recommendations from experts to support their decisions and to avoid waste of time. Recommender systems are proved (Ricci *et al.*, 2011) to be a valuable means to support users in their information-seeking process and cope successfully with the information overload problem.

The purpose of a recommender system is to assist the user to deal with the vast amount of information which is available on the Internet and, moreover, to function as a support tool to the decision-making process. In fact, recommender systems are tools that deal with the information overload by filtering information through various techniques and make suggestions for information items of probable interest to the user. One of the first studies in recommender systems was Tapestry (Goldberg *et al.*, 1992), which also introduced the term 'collaborative filtering'. Approaches based on collaborative filtering take into account the preferences of a multitude of users. The main concept of this technique is that common preferences and choices between two or more users in the past tend to be the same in the future. The other main technique in recommender systems is content-based filtering whereby the items are recommended according to the similarity of their characteristics. All the other approaches are based on one or both of these filtering techniques.

Despite the increasing research effort and a variety of approaches for improving recommendations, recommender systems still face limitations and need further improvements to be more effective and applicable to a broader range of real-life applications (Gao, Liu and Wu, 2010). A common problem is that users typically do not provide sufficient ratings for items, leading to a very sparse ratings matrix for the recommender system. This sparsity causes problems to typical collaborative filtering algorithms, basing their recommendations on user neighbourhoods formed by users that rate common items. The same problem also exists for users not having yet rated any item (cold-start users). Another usual problem of typical recommender systems is that of malicious attacks in copying a user's rating profile for gaining similarity. Additional limitations and challenges also exist in recommender systems which will be discussed later in Section 5.1. However, the aforementioned led to the enhancement of trust statements between users, which are relations between them, such as the 'friendship' relation in a social network. These systems are called 'trust-based recommender systems'.

The emergence and widespread use of social networks have afforded opportunities to develop new approaches for recommender systems exploiting not only the comments and tags created by users, but also the relationships between these users. Friends in a social network form a kind of trusted network, which is exploited by the trust-based recommender systems. A recent survey (Nielsen, 2015) reports that more than eight-in-ten global respondents (83%) say they completely or somewhat trust the recommendations of friends and family. But trust isn't confined only to those in our inner circle. In fact, two-thirds (66%) say they trust consumer opinions posted online. Gretzel and Yoo (2008) also report that word-of-mouth plays a key role in reducing the risks and uncertainty involved in the consumer's decision-making process. However, Social Web provides valuable information, such as the relationships between users forming trusted neighbourhoods. Therefore, recommender systems can take advantage of the Social Web and exploit the networks that built the users, to provide more trusted recommendations from friends and family than the recommendations based on completely unknown users. As a consequence, the social recommender systems emerged and, along with the trust-based recommender systems, generated a rising interest in the research area to improve the accuracy and quality of the recommendations by manipulating the available social information. However, just as the typical recommender systems suffer from sparsity in ratings matrix, trust-based systems also suffer from an insufficient number of trust statements.

While research in recommender systems focuses on the improvement of accuracy while maintaining privacy and scalability, current methods, however, need further improvements in order to be more effective (Mehta *et al.*, 2011) and to overcome existing problems. The cold-start problem, the sparseness and malicious ratings cause user profiles to become the weakest link in the whole recommendation process. On the other hand, the explosive growth of social networks, the empirical observations of the impact of word-of-mouth on a community and on social behaviours, afford potential to overcome these problems. However, without sufficient knowledge about users, even the most sophisticated recommendation strategy will not be able to make satisfactory recommendations (Zhou *et al.*, 2012). Finally, novel methods are needed for modelling the user to overcome the existing limitations.

1.2 Aim

The aim of this study is to improve the accuracy of trust-based recommender systems through the inference of new connections between users, based on existing relationships within a trust network.

1.3 Objectives

The objectives of this study are to:

- Review the literature on existing methods and techniques on recommender systems. Furthermore, review, identify and evaluate contemporary trust-based approaches (Chapter 3, Chapter 4 and Chapter 5).
- Develop a new method for expanding existing trust networks (Section 6.3).
- Develop a new method to calculate trust for inferred trust relationships (Section 6.4).
- Investigate the influence of the proposed methods on the accuracy of recommendations (Sections 6.5.1 and 7.2).
- Integrate the proposed methods into other state-of-the-art methods and investigate the influence on the accuracy of recommendations (Chapter 7).
- Evaluate the proposed methods and compare them with existing state-of-the-art approaches (Chapter 7). Present and analyse the results and discuss the learning outcome (Sections 7.3 and 8.1). Finally, draw conclusions about the proposed system and make any suggestions for future work (Sections 8.3 and 8.2).

1.4 Research hypotheses

Due to the fact that typically not enough users are directly connected within trust networks, the research hypotheses of this study are that, by exploiting the existing trust relationships, we can model users and infer new connections. The inferred trust relationships can then contribute to improve the accuracy of recommender systems.

1.5 Research questions

Therefore, the purpose of this study is to address the following questions:

RQ1. How can the accuracy of recommender systems be improved? Is it possible to utilise trust data to improve accuracy?

RQ2. How to can the sparsity in the item and trust ratings matrices be dealt with?

RQ3. How can we expand an existing trust network?

RQ4. Is there any new way of utilising existing trust data to expand the trust network? Can we predict the new connections from knowledge of the existing trust network?

RQ5. Can we predict topical similarity from the trust network?

RQ6. What is the impact of expanding a trust network on the accuracy of recommender systems?

1.6 Contributions

The main contribution of the current study is the use of a novel trust inference technique that increases the range of the existing neighbours of a user and improves recommendation accuracy since it performs better than existing standard trust inference techniques. The novelty of the proposed approach is the way that relationships are handled, as described later in Section 6.3, inspired by real-life scenarios. The other contributions of this research are summarised as follows:

- Proposes a new algorithm to infer trust relationships based on the triadic closure property of social networks (Section 6.3).
- Proposes a new method to calculate trust (Section 6.4).
- Investigates the triadic closure influence on the accuracy of recommendations (Section 6.5).
- Evaluates extensively the proposed algorithms in comparison with other well-known recommendation algorithms (Chapter 7).
- Provides evidence that the proposed model substantially improves prediction accuracy and coverage with respect to previous methods (Section 7.2).

1.7 Publications

This study led to the publication of the following two papers presented in conferences:

1. Tselenti, P. and Danas, K., 2014. A Review of Trust-Aware Recommender Systems Based on Graph Theory. In *International Conference on Computer Science, Computer Engineering, and Social Media*. 12-14 Dec 2014, Thessaloniki, Greece, pp. 1–12.
2. Tselenti, P. and Danas, K., 2016. Trust-based recommendations through triadic closure. In *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*. Chalkidiki: 13-15 July, IEEE, pp. 1–6.

1.8 Thesis overview

An overview of the remainder of this study is presented below:

Chapter 2 presents the methodology followed for this study. All the stages of the engineering cycle followed to complete this study are analysed and are linked with the sections of the remainder of this study. A special focus is given on the datasets and the measures used during the experimental study along with a detailed description of the experimental design in order to be reproducible. Finally, it presents all the tools and technologies used during all the stages for completing this study.

Chapter 3 is a literature review presenting the various approaches for producing recommendations and analysing the two major methods of recommender systems with the way they work. Additionally, there is a Section (3.6) presenting the various methods for evaluating recommender systems.

Chapter 4 provides an extensive presentation of the trust-based recommender systems, including definitions and computational properties of trust. The investigation of the trust inference mechanism and the survey of the existing trust metrics is the necessary preparatory step for obtaining all the in-depth knowledge to fulfil the aim of this study.

Chapter 5 initially presents the limitations of the current recommender systems and then surveys and classifies existing trust models based on the techniques they use. Then, it presents a comparison of the main literature in graph-based models and concludes that there is need for a new system that considers also the ‘common friends’ and not only ‘the friend of my friend’ to propagate trust and incorporate it in recommender systems.

Chapter 6 proposes a novel method to infer trust, called TriadicClosure, based on the homophily phenomenon. Initially, it presents the mechanism of forming links in a social network and how it is measured. Next, it describes thoroughly the mechanism of inferring trust with triadic closure and presents the TriadicClosure algorithm, as well as a new method to calculate trust, based on the JaccardCoefficient algorithm. Finally, the chapter ends with a description of how to incorporate the proposed methods in the recommendation process and validates the proposed methods by evaluating them with synthetic data.

Chapter 7 presents and analyses the results of the experimental evaluation of the two proposed methods (TriadicClosure and JaccardCoefficient). Initially, the TriadicClosure algorithm is compared with basic trust-based approaches and then it is incorporated into the state-of-the-art trust-based approaches. In the next stage, the JaccardCoefficient is compared against all the above methods and, finally, the performance of both the proposed methods is evaluated for different views of datasets.

Finally, Chapter 8 discusses the results of experimental evaluation of the methods in comparison with other state-of-the-art methods presenting the usefulness and impact of the TriadicClosure method and how the research questions have been addressed. Moreover, it discusses the limitations of the two methods and also any challenges that arose during this study. Then, it discusses possible extensions and applications of the proposed methods for future work and, finally, it concludes, based on the literature review and the results of the experimental evaluation, by stating the contributions of this study.

Chapter 2

Research methodology

The current study follows the engineering cycle to solve a ‘world problem’ as described by Wieringa and Heerkens (2006). This ‘world problem’ here is described by the aim (Section 1.2) of the current study. The steps (Figure 2-1) followed to solve the research problem as described previously (Chapter 1), are listed next:

1. **Problem investigation**. Extensively review the literature and investigate in-depth the research problem.
2. **Solution design.** Think and examine several techniques that could offer a solution to the problem.
3. **Design validation.** Initially, validate the proposed method.
4. **Choice of solution**. Choose the best solution.
5. **Implementation description.** Describe the implementation of the chosen method.
6. **Implementation evaluation.** Evaluate the proposed method.

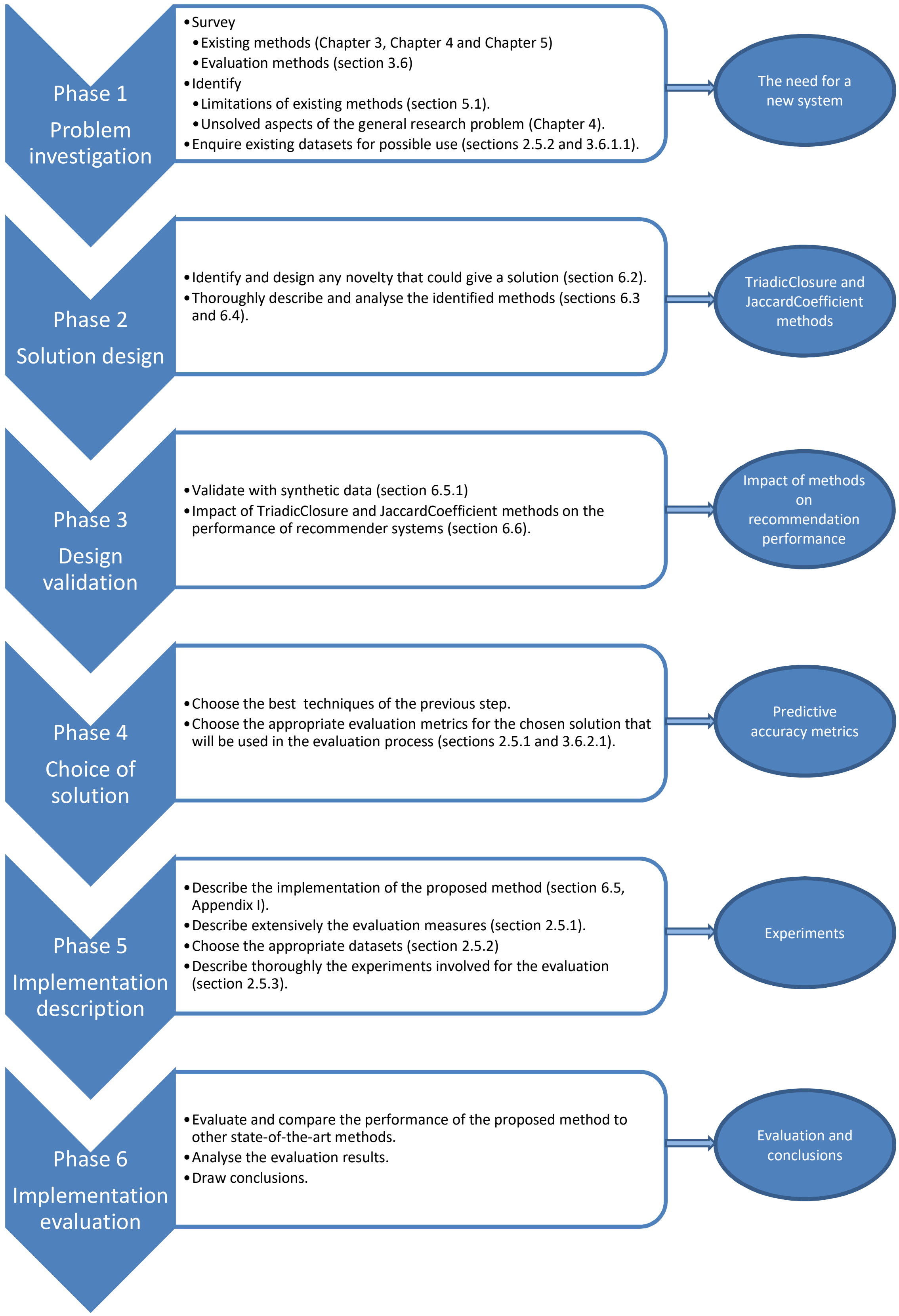


Figure 2-1 Steps followed to solve the research problem

2.1 Problem investigation

This step (Chapter 3, Chapter 4 and Chapter 5) involves the extensive literature review and the in-depth investigation of the research problem. More specifically, the tasks of this step are to:

a. Study the state-of-art solutions and identify possible limitations of each one of them (Chapter 3, Chapter 4 and Chapter 5).
b. Detect any unsolved (sub-problems) aspects of the general research problem (Section 5.1).
c. Except the methods to be surveyed, enquire also for any existing datasets that could be used (Sections 2.5.2 and 3.6.1.1).
d. Survey the evaluation methods followed in the specific research area (Section 3.6).

The outcome of this step is the need of a novel method to infer new connections between users in trust-networks considering not only 'the friend of my friend' but also the 'common friends'.

For the literature review and the in-depth investigation of the research problem, various tools and sources were used, such as hardcopy books, digital libraries and scholar search engines as well as also forums, membership to professional groups and networks, etc. Kingston University's online library catalogue (iCat) was useful for both printed and electronic resources searching, but also to view online the full text of electronic resources such as journals, conference proceedings and electronic books.

Membership to Springer and ACM as well as ResearchGate proved valuable for extending and updating the review on the state-of-the-art methods. Moreover, the Mendeley reference manager proved one of the most valuable tools during this research, since, except its primary role as a reference manager, it also offers functionality to search and import sources as well as provides personalised suggestions for articles based on the user's library.

2.2 Solution design

In this step (Chapter 6) several techniques are examined that could offer a solution to the problem. The tasks of this step are to:

a. Identify and design any novelty that could give a solution (Section 6.2).
b. Thoroughly describe and analyse the identified methods (Sections 6.3 and 6.4).

More specifically, existing methods of the research area of Social Network Analysis are examined as a possible solution to the problem. Finally, **TriadicClosure** is proposed as a solution to infer new trust relationships. Moreover, a novel method to calculate the trust weight of an inferred trust relationship is proposed, called **JaccardCoefficient**. The design and the description of the two proposed methods are thoroughly discussed in Chapter 6.

2.3 Design validation

In this step, the proposed methods of the previous step are validated as an initial proof of their validity. To achieve this, the proposed methods are initially evaluated with synthetic data (Section 6.5.1) in order to form an early conclusion about the performance and validate the design of the proposed methods. The results of this evaluation also indicate the impact of the proposed methods on the performance of the recommender system when recommending items (Section 6.6).

2.4 Choice of solution

In this step, the best solution is chosen (Chapter 7), involving two tasks:

a. Choose the best of the validated techniques of the previous step.
b. Choose the appropriate evaluation metrics for the chosen solution that will be used in the evaluation process. In the current study, the experiments for measuring and evaluating the quality of rating predictions of various methods, use the **predictive accuracy metrics** (Sections 2.5.1 and 3.6.2.1), since the proposed methods, and, also, the examined state-of-the-art methods, produce **item ratings** and, in this case, these are the most appropriate metrics.

2.5 Implementation description

The chosen method of the previous step that best serves the solution to the research problem is implemented (Chapter 6 and Chapter 7); therefore, the tasks are to:

a. Describe the implementation of the proposed method (Section 6.5, Appendix I).
b. Describe extensively the evaluation measures (Section 2.5.1).

c. Choose the appropriate datasets (Section 2.5.2)

d. Describe thoroughly the experiments involved for the evaluation (Sections 2.5.3, 7.1 and 7.2).

2.5.1 Evaluation measures

In the experiments, all the methods are evaluated for their performance in terms of prediction accuracy and coverage. The evaluation metrics used for measuring the prediction accuracy of all the methods are the Mean Absolute Error or else MAE (Eq. 3.16) and the Root Mean Squared Error or else RMSE (Eq. 3.18). Recall that RMSE emphasises large errors thus, for a broader view of the performance of each algorithm the two metrics (MAE and RMSE) are used together to diagnose the variation in the errors of prediction. For consistency, the two metrics are given below as in Section 3.6.2.1:

$$MAE = \frac{\sum_{(u,i)\in Test} \left| p_{u,i} - r_{u,i} \right|}{\left| Test \right|} \qquad \text{(Eq. 3.16)}$$

and

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in Test} \left(p_{u,i} - r_{u,i} \right)^2}{\left| Test \right|}} \qquad \text{(Eq. 3.18)}$$

where $r_{u,i}$ is the real rating while $p_{u,i}$ is the predicted rating for a pair of user-item u,i and $|Test|$ is the size of the testing set. The lower the MAE and RMSE are, the more accurately the recommendation engine predicts user ratings.

Moreover, as it will be analysed later (Section 3.6.3), coverage is a measure being always considered in conjunction with accuracy. Therefore, we define the **Ratings Coverage (RC)** which is similar to the predictions coverage identified by (Herlocker *et al.*, 2004) and measures *"the ratio of the number of predicted rating over the total number of the testing ratings"*, described by the formula:

$$RC = \frac{\left| R_{pred} \right|}{\left| R_{test} \right|} \qquad \text{(Eq. 1.1)}$$

Users Coverage (UC) is similar to 'user space coverage' described by (Shani and Gunawardana, 2011) and is the *"proportion of users for which the system can provide recommendations"*. It is formulated as:

$$UC = \frac{|U_{pred}|}{|U_{test}|} \qquad \text{(Eq. 1.2)}$$

were $|U_{pred}|$ is the number of users that the method provided predictions and $|U_{test}|$ is the number of users in the testing set.

Combining the RMSE and coverage RC into a single metric resulting in the $FMeasure$, which is the balance between accuracy and coverage. According to (Jamali and Ester, 2009), we can compute it as:

$$FMeasure = \frac{2 \times Precision \times RC}{Precision + RC} \qquad \text{(Eq. 1.3)}$$

where $Precision$ is the RMSE converted into a precision metric in the range [0,1] defined as follows:

$$Precision = 1 - \left(\frac{RMSE}{r_{max} - r_{min}} \right) \qquad \text{(Eq. 1.4)}$$

with r_{max} — r_{min} the maximum and the minimum rating value of the rating scale, respectively. Higher values of $FMeasure$ indicate better overall performance.

All the above metrics are used to measure the performance of the proposed methods and to compare it with the performance of other state-of-the-art methods.

2.5.2 Datasets

For this study, two real-world datasets are used in the experiment, namely Filmtrust and Epinions. The reason for choosing these two datasets between others (Section 3.6.1.1) is that both include not only user-item ratings but also explicit trust statements being necessary to conduct the experiments. Filmtrust is chosen as a first step of the offline performance test and Epinions is chosen to test the performance on a large dataset.

Table 2-1 Statistics of the two datasets

	Filmtrust	Epinions
# of Users	1,508	22,164
# of Items	2,071	296,274
# of Ratings	35,497	912,373
# of Trust Links	1,853	355,754

Filmtrust is a trust-based social site where users can rate and review movies. It was built as an experimental platform (Golbeck, 2006b). The dataset used in the experiments was crawled by Guo *et al*. (2013) in June 2011 and contains 35,494 ratings for 2,071 movies from 1,508 users. The ratings scale is between 0.5 and 4.0 with step 0.5 while the trust statements are bivalent with trust values 1 for expressing that the other user is trustworthy and 0 for expressing that the other user is untrustworthy. The trust network consists of 1,853 trust statements from 609 users.

Epinions has been extracted by Tang *et al*. (2012) from the Epinions.com which is an online product review site. Users can rate and/or review products in different categories such as movies, books, games etc. but they can also build their own trust network by expressing trust on other users. The dataset contains 912,373 ratings for 296,274 items in various categories from 22,164 users. The ratings scale is integer from 1 to 5. The trust statements are bivalent with 1 for expressing that the other user is trustworthy and -1 for expressing that the other user is untrustworthy.

2.5.3 Experimental design

Next, a series of experiments were conducted for testing and evaluating the effectiveness of the proposed methods. The empirical study was conducted with the two different datasets described previously, aiming to compare the performance of the proposed methods with different state-of-the-art trust-based methods. Specifically, the experimental evaluation intends to address the following general questions:

(Q1) How does the TriadicClosure algorithm perform on accuracy compared with different state-of-the-art trust-based methods?

(Q2) What is the impact of the TriadicClosure on coverage?

(Q3) How does the TriadicClosure algorithm perform on large datasets?

(Q4) How does the TriadicClosure algorithm perform on accuracy and coverage when integrated within other state-of-the-art trust-based methods?

(Q5) What is the impact of propagation on the TriadicClosure?

(Q6) How does the JaccardCoefficient algorithm perform on accuracy and coverage when implemented within the TriadicClosure algorithm and other state-of-the-art trust-based methods?

(Q7) What is the performance comparison of the two proposed methods on different views of users (cold-start, heavy-raters, grey-sheep, controversial items, niche items)?

The experiments can be divided into four conceptual stages. Each stage answers one or more of the questions set above.

Stage 1 TriadicClosure basic evaluation

This is the initial stage in which the TriadicClosure algorithm is compared with basic trust-based approaches. The experiments of this stage try to give an answer to questions (Q1), (Q2) and (Q3)

Stage 2 TriadicClosure total performance

In this stage, the TriadicClosure algorithm is incorporated into the state-of-the-art trust-based approaches. The experiments of this stage try to answer questions (Q4) and (Q5).

Stage 4 JaccardCoefficient performance

In this stage, the JaccardCoefficient is compared against all the above methods. The experiments try to give an answer to question (Q6).

Stage 5 Performance of TriadicClosure and JaccardCoefficient for different views

This stage answers the question (Q7) by comparing all the above methods for different views of users.

In order to test the performance of the proposed methods for different views the datasets were split, as defined in (Massa and Avesani, 2007):

- **All** represents the whole dataset.
- **Cold-start users** refer the users who rated less than 5 items.
- **Heavy raters** are those who rated more than 10 items.
- **Grey-sheep users** rated more than 4 items, and the average difference between their average rating and the mean rating of each item is greater than 1.

- **Controversial Items** are those which received ratings with a standard deviation greater than 1.5.
- **Niche Items** refers to items which received less than 5 ratings.

The statistics for these views for the two datasets are shown in Table 2-2 and Table 2-3.

Table 2-2 Statistics for different views in Filmtrust

	# of Users	**# of Items**	**# of Ratings**
All	1,508	2,071	35,497
Cold-start users	281	156	608
Heavy raters	963	2,034	32,979
Grey-sheep users	93	214	2,893
Controversial Items	60	30	73
Niche Items	382	1,653	3,162

Table 2-3 Statistics for different views in Epinions

	# of Users	**# of Items**	**# of Ratings**
All	22,164	296,274	912,373
Cold-start users	22	63	64
Heavy raters	20,750	295,173	899,561
Grey-sheep users	1,535	14,842	31,718
Controversial Items	10,594	3,149	22,223
Niche Items	19,983	264,837	372,188

The experiments follow the standard *leave-one-out* (Section 3.6.1.1.1) as validation process. Thus, the true user-item rating is hidden and a predicted value is calculated for each method we want to evaluate. This process is iterative until all ratings in the dataset are tested.

2.6 Implementation evaluation

This final step verifies whether the proposed method solved the problem and to what extent (Chapter 7and Chapter 8) and involves three tasks:

a. Evaluate and compare the performance of the proposed method to other state-of-the-art methods.
b. Analyse the evaluation results.
c. Draw conclusions.

2.7 Tools and technologies

For the experimental study, the LibRec[1] Java library for recommender systems was used to implement the proposed methods and perform the evaluation. The LibRec library implements a suite of state-of-the-art recommendation algorithms as well as the traditional methods. In addition, a series of evaluation metrics are implemented including diversity-based metrics which are rarely enabled in other libraries. LibRec provides a platform for fair comparisons among different algorithms in multiple aspects, given the fact that the evaluative performance depends on data characteristic. It also provides a high flexibility for expansion with new algorithms.

Other popular open source recommendation frameworks are available such as Apache Mahout[2], Duine[3], LensKit[4], MyMediaLite[5], and PREA[6]. Lee *et al*. (2012) from their detailed comparison, report that Mahout, Duine provide only simple memory-based algorithms while recent state-of-the-art algorithms are often not supported. Moreover, LensKit provides only a few classic recommendation algorithms. Guibing *et al*. (2015) also provide a comparison between PREA, MyMediaLite and LibRec reporting that the first two libraries become less active for further development while regarding the evaluation performance of recommender algorithms, PREA only provides predictive error-based metrics while MyMediaLite does not provide novel measures beyond accuracy. In contrast, LibRec library provides novel measures such as coverage, as well as the traditional accuracy-based measures. The researchers finally demonstrate that LibRec runs much faster than PREA and MyMediaLite while achieving competitive recommendation performance. Hence, for this study, the LibRec framework seemed to be the most appropriate choice for implementing the proposed algorithms and perform the evaluation experiments comparing with recent state-of-the-art algorithms.

[1] http://www.librec.net/index.html
[2] https://mahout.apache.org
[3] http://www.duineframework.org/
[4] http://lenskit.org/
[5] http://www.mymedialite.net/
[6] http://prea.gatech.edu/

Chapter 3

Recommender Systems

From the early beginnings of the Internet, one of the main research challenges was to effectively manage and filter all available information. As already mentioned, the aim of a recommender system is to help the user cope with the vast amount of information which is available on the Internet and, even more, to perform as a supporting tool to the decision process. Actually, a recommender system is a tool dealing with the information overload by filtering information and, through various techniques, makes suggestions for information items being of probable interest to the user. The suggestions can be based on the popularity of an item, the purchase history of the user, demographic information about the user, or similarities of the current user with other users or even with a community of users.

Initially, recommender systems were based on cognitive science (Rich, 1979) and approximation (Powell, 1981), while applied techniques and methodologies of information retrieval (Salton, 1989) and algorithms of Data Mining (Ricci *et al.*, 2011). In the early 1990s, they became an independent research area by taking advantage of user ratings. Since then, the interest for this research area has constantly increased with many applications taking advantage of the different existing methods.

Both information retrieval systems and recommender systems use similar techniques for filtering and ranking information, but recommender systems have the fundamental difference of taking into account the user's preferences. On the other hand, information filtering and retrieval techniques are mainly based on item description without considering user preferences or context. Recently, web search engines, such as Google, have taken advantage of recommendation techniques to filter their results as well as to implement advanced search features like user's location and language.

Real-life applications of recommender systems exist for various topics, such as e-commerce, news, dating, jobs citations, websites, recipes, etc. A list of some very popular real-life applications is given in Table 3-1, in which two of the best known and oldest recommender systems are included (Amazon and Google News). At this point, we will consider how these two popular websites produce their recommendations. Suppose a user is interested in purchasing a specific book from Amazon. As soon as the user searches for the book, Amazon recommends some other books also having similar characteristics, such as the topic or the writer and/or books or even other products that other users bought together with the specific book. On the other hand, Google uses a very different approach to recommend web pages. At the heart of Google's method is the PageRank algorithm (Brin and Page, 1998) which ranks the web pages based on the analysis of the hyperlink network of the websites.

Table 3-1: Real-life recommender systems

Application/site	**Topic**
Netflix	Movies
Pandora	Music
Amazon	Books and other products
Facebook	Friends
MovieLens	Movies
StumbleUpon	Websites
Perfectmatch	Dating
CareerBuilder	Jobs
Google News	News
Last.fm	music
eBay	Various products

Another very popular website is Netflix, a movie recommender site which, in October 2006, announced a competition of $1 million Prize[7]. This was a challenge to improve the accuracy of the company's prediction algorithm (Cinematch) by more than 10% in terms of the root mean square error (RMSE). On September 2009, the BellKor's Pragmatic Chaos team won the grand prize, out of 20,000 registered teams from which there were 2,000 submitted prediction sets with a total of 13,000 submissions. The Netflix Prize reflects the importance and the economic impact of recommender systems on e-commerce. Several studies examine this impact on sales (Fleder and Hosanagar,

[7] http://www.netflixprize.com/

2007; Pathak *et al.*, 2010; Sivapalan *et al.*, 2014) and have proved (Pathak *et al.*, 2010) that the strength of recommendations has a positive effect on sales as well as the cross-selling efforts of sellers.

3.1 How do they work?

The goal of a recommender system is to suggest new items to a target user. This can be achieved by utilising information about items and/or users. Predictions can be made based on users with similar tastes or by finding the most popular item.

The input values for a recommender system are subject to the needs of the algorithm. The usual values that can be used are:

- User data; which can be demographic data (age, gender, education, language), location, browsing history, social network, trust relationships, content data written by the user (comments, stories, etc).
- Item data; information about the item, like title-name, alternative names, item characteristics like genre, etc.
- Ratings data; the ratings about the items given by the users. The rating scale is usually numerical and can be from 1 to 5 (Likert scale), or from 0 to10 or from 0 to 1 or even from 1 to 100.

To examine how a recommender system works, let $U=\{u_1,u_2,...,u_m\}$ be the set of users with *m* the number of them and $I=\{i_1,i_2,...,i_n\}$ the set of items with n the number of available items. Every user u_i with $i=1,2,\dots,m$ has rated a list of items $I_{u,i}\subseteq I$. Let f_u be the utility function (Adomavicius and Tuzhilin, 2005) measuring the usefulness of item i to user u, i.e., $f_u\colon U\times I\rightarrow R$, where R is a totally ordered set that, in fact, represents the ratings set. Then, for each user $u\in U$, we want to choose such an item $i\in I$ that maximises the user's utility. More formally:

$$\forall u\in U \quad and\ i\in I,\ i_u=max(f_u(u,i)) \qquad (Eq.\ 3.1)$$

The rating $r_{a,j}\in R$ that the target user u_a has given for an item i_j is a number according to the scale being used in the system. This number is usually between 1 and 5 or 0 and 10 or even between 0 and 1.

The task of the recommender system is to make a prediction for an item $i_j \notin I_{u,i}$ that the active user $u_a \in U$ has not yet rated. The output of the recommender system can be either:

- Prediction, which refers to the rating prediction $p_{a,j}$ of a user u_a to a specific item i_j that has not yet been rated. It is a numerical value, within the scale that the recommender system uses.
- Recommendation, which is a list of N items of interest for the active user, where $N \leq n$. Note that this top-N list refers to items that the active user may like more and, of course, has not yet purchased and/or rated.

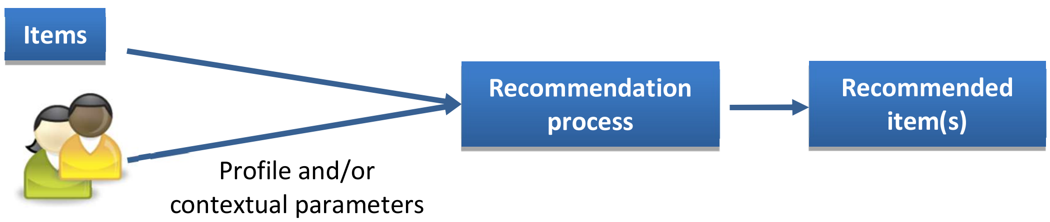


Figure 3-1 Inputs and output of a recommender system

To give an example, suppose we have the ratings matrix of Table 3-2, with users in rows and items in columns. The value in each cell represents the rating score for a movie from each user. Essentially, the ultimate goal for a recommender engine is to predict values for the empty cells, viz., ratings for the not-yet-rated movies.

Consequently, in a typical collaborative filtering system, the input is the ratings matrix and its goal is to fill, as much as possible, the empty cells of the matrix according to the similarity of ratings between users.

Table 3-2: An example of user-item ratings matrix

		Movie				
		Memento	Pulp Fiction	The Godfather	Titanic	The Room
User	**Alice**	1	4	4	3	5
	Bob	4	5	4	1	
	John		2	4		3
	Emily	3	5			
	Frank		3	3		4

3.2 Recommendation approaches

The two major methods for producing recommendations are the content-based filtering technique, which relies on item attributes and/or the historical data of the user, and the collaborative filtering technique, which is based on the opinions and preferences of other users. In everyday life, we usually seek advice from friends or colleagues or someone we trust to recommend us a movie, a book, a restaurant or other things of interest. In fact, collaborative filtering recommender systems follow the assumption that, if a user has common preferences with other users in the past, they are more likely to also like other things that these users liked. Whilst there are hybrid techniques combining both of the above, collaborative filtering is the most successful and widely-used technique (Ray and Mahanti, 2010). One of the first commercial recommender systems, as already mentioned, was Tapestry (Goldberg *et al.*, 1992), which was purely based on collaborative filtering. Two years later, in 1994, researchers (Resnick *et al.*, 1994) introduced a k-Nearest-Neighbour algorithm. Since then, several optimisations of these algorithms or even entirely new approaches have been proposed.

Based on the information filtering techniques that are used to make their recommendations, recommender systems can be classified (Nageswara Rao and Talwar G., 2008) into the following categories.

- Content-based filtering systems: recommendations are based on items and information that users preferred in the past.
- Collaborative filtering systems: recommendations are based on items that people with similar tastes and preferences liked in the past.
- Demographic filtering systems: user data such as age, gender, education, etc., are considered to form the recommendations.
- Knowledge-based filtering systems: recommendations are based on specific domain knowledge about how a specific item meets the user's interests.
- Utility-based recommender systems: suggestions are based on the computation of the utility of each object for the user.
- Community-based filtering systems: preferences of user's friends are considered for item recommendations.
- Hybrid filtering systems: combination of two or more filtering techniques to minimise each one's deficiencies and/or benefit each one's advantages.

However, the emergence and the wide spread of social networks has given opportunities in developing new approaches for recommender systems exploiting, not only the comments and tags created by users, but also the relationships, bringing to the forefront of the research the Social recommender systems and the Trust-based recommender systems. Although both are based on Social Web, the fundamental difference is that, while trust-based systems are based on social relations to produce suggestions for items, the Social recommender system provides proposals for connecting with new friends within the social network.

As a general rule, recommender systems build and exploit the user model to generate recommendations, by modelling the profile that contains information about users' tastes, preferences and needs (Adomavicius and Tuzhilin, 2005).

In demographic recommender systems, for example, the user model contains demographic information like age, gender, education, etc., whereas, in collaborative filtering, the user is modelled by his ratings to the items (Ricci *et al.*, 2011). Two methods exist for collecting information about the user:

- Explicit, where the information is provided explicitly by the user through the completion of a checking list of interests or by collecting previous ranking information given by the user.
- Implicit, where information is extracted from the user's browsing history through an automated reasoning mechanism.

The aforementioned approaches and techniques are static, whereas user preferences and attributes change over time. Thus, there is a need (Nanas, De Roeck and Vavalis, 2009) for constant updates of the user profile for producing better recommendations. Recently, research has focused on the semantic description of the user model enriching user profiles with metadata, moving on from the conventional vector representation of the user model. Many approaches describe the user model semantically (Heckmann *et al.*, 2005a; Heckmann, *et al.*, 2005b; Felden and Linden, 2007; Kim *et al.*, 2007) some of which deal with information interchange between user modelling systems. Some others analyse general user models, such as:

- UserML (Heckmann and Krueger, 2003), a mark-up language for communication about partial user models in a ubiquitous computing environment, where all different kinds of systems work together to satisfy the user's needs and

- GUMO (Heckmann *et al.*, 2005b) for the uniform interpretation of distributed user models in intelligent semantic web enriched environments.

In a recent study (Zhang, Song and Song, 2007), the user model is based on the semantic representation of the user's activity, taking also into account the structure of visited websites. Lately, research in Social Network Analysis (McGrath, 2008) and Natural Language Processing (Wilks and Brewster, 2009) has offered a new perspective and solutions in the semantic description of user model. Methods based on Social Network Analysis model the users through their relationships and their interactions with other users. Social Network Analysis is a multidisciplinary research area based on social sciences, mathematics, computer science and physics and which attempts to quantify the interactions among the users of a social network in order to profile the users and the network's structure by investigating social group dynamics. Specifically, Social Network Analysis offers measures for the reputation and the importance of a user in a social network, but also provides the way to divide the network into subgroups (cliques). The current study is based on Social Network Analysis, exploiting methods to predict link connections such as propagation and homophily.

3.2.1 Similarity measures

Similarity measures or metrics compute the similarity between two objects. Some of the most successful recommendation approaches are based on similarity measures, either between users for collaborative filtering or between items for content-based. In the first case of collaborative filtering, similarity is computed between two users and is based on all rated items by the two users. Likewise, for content-based systems, similarity is calculated for items and is usually based on their features. The following subsection presents some of the most common similarity metrics, used in recommender systems, as referenced also by Ricci *et al*. (2011).

The quantified similarity between two objects is usually defined as the inverse of distance metrics. Similarity $w_{a,u}$ can be computed by measuring the distance between two objects x, y.

The simplest measure is the **Euclidean distance**

$$w_{x,y} = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2} \quad \text{(Eq. 3.2)}$$

where n is the number of attributes and x_k and y_k are the k^{th} attributes of objects x and y, respectively.

Minkowski distance is a generalization of Euclidean Distance

$$w_{x,y} = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{\frac{1}{r}} \quad \text{(Eq. 3.3)}$$

where r is the degree of the distance. When $r=1$ we get the **Manhattan distance**

$$w_{x,y} = \sum_{k=1}^{n} |x_k - y_k| \quad \text{(Eq. 3.4)}$$

Although these functions work well with numeric values for computing similarity, when the attributes are not numerical like the genre of a movie, there is a need for another approach to computing similarity.

When items are not numerical, the angle between the two items should be calculated. In **cosine based** similarity the items or users are represented as vectors and the similarity is the cosine of the angle between these vectors.

$$w_{x,y} = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|^2 \times \|\vec{y}\|^2} = \frac{\sum_{i \in R_{x,y}} r_{x,i} r_{y,i}}{\sqrt{\sum_{i \in R_{x,y}} r_{x,i}^2} \sqrt{\sum_{i \in R_{x,y}} r_{y,i}^2}} \quad \text{(Eq. 3.5)}$$

where $\vec{x} \cdot \vec{y}$ is the dot-product of these two vectors and $R_{x,y}$ is the set of commonly rated items by both users x and y. The result of this function is between -1 for full dissimilarity and 1 for full similarity.

Pearson correlation coefficient (PCC) is another way for computing similarity by measuring the linear correlation between two vectors of ratings

$$w_{x,y} = \frac{\sum_{i \in R_{x,y}} \left(r_{x,i} - \overline{r_x}\right)\left(r_{y,i} - \overline{r_y}\right)}{\sqrt{\sum_{i \in R_{x,y}} \left(r_{x,i} - \overline{r_x}\right)^2 \sum_{i \in R_{x,y}} \left(r_{y,i} - \overline{r_y}\right)^2}} \quad \text{(Eq. 3.6)}$$

where $R_{x,y}$ is the set of commonly rated items by both users x and y. The $\overline{r}$ denotes the user's average rating and reflects the different way that every user considers the rating scale. If a user tends to give higher rates than another, the inclusion of the average rating for each user normalises the results. Again, the result is between -1 for full dissimilarity and 1 for full similarity. However, this method may give misleading results when for instance two users may happen to rate a few items identically, but this does not imply that they have similar overall preferences. Yet, Pearson Correlation Coefficient is one of the most popular similarity methods as it is proved (Breese, Heckerman and Kadie, 1998) to perform better than cosine similarity and other similarity measures.

3.3 Content-based recommendations

Content-based recommender systems have their roots in information filtering and retrieval and rely on data about the available items. These data are attributes or features which describe an item. For example, a song can be described by attributes like the title, the creator, the composer, the singer, the genre, etc. The main idea behind this technique is that users tend to like items with similar characteristics. If a user has already rated or purchased or liked an item, the system generates recommendations for 'similar items' by making item-to-item correlation. Thus, the system suggests items similar to these that the user liked in the past. In fact, the content-based algorithms compare the user's profile with terms or keywords representing the attributes of items to produce recommendations. User profiles can be built, either explicitly or implicitly, by gathering historical information from positive reviews, ratings, purchases or even searching keywords.

Both items and user preferences are stored as features vectors. The features can be extracted by text processing techniques or by manual labelling using the Tagging function of Web 2.0 technologies.

A popular technique used in content-based recommenders is to find the most important words in a document and build the document profile. The importance of words in a document can be calculated based on their frequency in the document. This technique is called TF-IDF, which stands for Term Frequency/Inverse Term Frequency and has its roots in information retrieval (Salton, 1989). Considering that the length of the document may increase the relative document weight, the term frequency is normalised by calculating it over the maximum frequency of the other keywords in the document The whole procedure to build the profile of a user is presented below, as described by Adomavicius and Tuzhilin (2005).

So the term frequency $TF_{i,j}$ of keyword k_i in a document d_j is defined as:

$$TF_{i,j} = \frac{f_{i,j}}{\max_z f_{z,j}} \quad \text{(Eq. 3.7)}$$

where $f_{i,j}$ is the frequency of keyword k_i in the text and $max_z f_{z,j}$ is the maximum frequency of all the other keywords in the document.

The second measure that should be calculated for the TF-IDF technique is the inverse document frequency which is defined as:

$$IDF_i = \log \frac{N}{n_i} \quad \text{(Eq. 3.8)}$$

with N the number of all recommendable documents and n_i the number of documents in which the keyword k_i appears

Finally, the weight $w_{i,j}$ for keyword k_i in document d_j is calculated as the product of the term frequency and the inverse document frequency

$$w_{i,j} = TF_{i,j} \times IDF_i \quad \text{(Eq. 3.9)}$$

The document profile is built from the weights of all the terms and is defined as:

$$Content(d_j)=(w_{1,j}, \ldots, w_{k,j})$$

The next step is to build the user profile. Let $ContentBasedProfile(c)$ be the profile of a user c which represents the preference of the user. The profile is a vector, containing keywords analysed from the previously rated and seen items. Finally, the two vectors, of document and user profiles are compared using a similarity measure and the score being extracted is used to recommend the documents with the higher scores.

In addition to the memory-based approaches that have their roots in information retrieval there are also model-based approaches. These methods address the recommendation of a document as a classification problem wherein each user is associated with a classifier instead of a profile. This means that supervised machine learning techniques can be applied, such as Bayesian classifiers (Pazzani and Billsus, 1997; Mooney, Bennett and Roy, 1998), decision trees (Pazzani and Billsus, 1997), clustering, linear regression, support vector machine (SVM) (Wu, Qi and Feng, 2007) and artificial neural networks (Park, Seo and Jang, 2005).

However, as opposed to several studies (Adomavicius and Tuzhilin, 2005; Nageswara and Talwar , 2008; Jannach *et al.*, 2010; Ricci *et al.*, 2011; Bhuiyan, 2013; Bobadilla *et al.*, 2013), content-based recommender systems suffer some limitations over other recommendation techniques.

- **Cold-start problem** for new users. Namely, there is no information for the user that has not yet rated any item. The system provides recommendation according to the user's preferences. But the user's profile is not complete until there are enough ratings from the user. Hence, the recommendations will not be very accurate until the user profile is sufficiently complete.
- **Attribute extraction** problem. Some domains cannot exploit this technique as they cannot use attributes to describe their items due to limited content analysis. The attributes of the items sometimes have to be inserted manually, as in multimedia content, which needs considerable human effort.
- **Overspecialisation** problem. The system lacks serendipity and cannot recommend any item being different to those that the user has previously rated. This means that, if a user likes a horror movie, the system will recommend only horror movies, although the user may like also action movies.
- **Objectivity** problem. The recommendations are not based on qualitative information, but only on objective information, since the items are described only by their attributes.

On the other hand, there are, of course, some advantages, making the content-based filtering valuable for the recommendation process.

- There is no **new-item** problem. This means that, if a new item is added to the system, it is not necessary to be rated or to be popular in order to be in the recommending items list.
- No need of **data about other users**; therefore, content-based methods do not suffer from sparsity problems.
- Enough **explanations** are provided by displaying the attributes on which recommendations are based.
- **Privacy** is preserved, as the recommendations are based only on personal information, which is not used for providing recommendations to other users. Moreover, the personal profile can be maintained locally for security reasons.

From all the above advantages and disadvantages of the content-based method, it is clear why this method is usually applied as a part of the hybrid systems. Combined with other methods, the drawbacks of each method can be eliminated, while, at the same time, their strengths can improve the overall recommendation performance.

3.4 Collaborative Filtering recommendations

As already mentioned, the first commercial recommender system was Tapestry (Goldberg *et al.*, 1992), which also introduced the term 'collaborative filtering'. The system took advantage of newsgroups to recommend documents to a collection of users so as to prevent information overflow.

Recommender systems based on collaborative filtering consider the preferences of a multitude of users. The main concept of this technique is that common preferences and choices between two or more users in the past tend to be the same in the future. Consequently, recommendations are based on the opinions of other users, thus, items from different categories/domains can be recommended.

To produce recommendations, a collaborative filtering system, analyses users' opinions and ratings and, thereafter, predictions are produced through a correlation engine for matching user preferences. In the classic user-based approach, the system builds a neighbourhood of users with similar tastes, so recommendations can be produced for items not necessarily similar to those previously rated by the user, but based on the ratings of users belonging in the user's neighbourhood. The target user is correlated to other users through similarity metrics on their profiles which forms a neighbourhood consisting of the nearest-neighbour users. Items rated by these nearest-neighbour users can be recommended to the target user. Accordingly, in the item-based approach, recommendations are built upon the similarities between items. This approach is based on the idea that a user that liked or purchased a specific item is possible to like or purchase a similar item in the future. A major difference with the user-based approach is that item-based has usually an offline phase for data pre-processing, which results to improved computation time in the prediction phase.

The input for a collaborative filtering recommender is the same as in content-based filtering and is a user-item ratings matrix. The typical steps for predicting items to a target user are as follows:

1. Calculate the similarity between users so as to form the neighbourhood
2. Form the neighbourhood and select a subset on which predictions will be based
3. Predict a rating for not yet rated items from the active user

For example, in a movie collaborative-filtering recommender, in order to recommend a movie to a target user, u_a, the system finds users with similar tastes (peers) and forms a neighbourhood of, let's say, 5-nearest neighbours. Then the system recommends the most liked movies by these five neighbours.

Depending on the technique that they exploit, collaborative filtering algorithms are classified into two general classes (Breese, Heckerman and Kadie, 1998) depending on the way that the input matrix is utilised:

- Memory-based: which are heuristics by utilising all the ratings of the user to produce a rating prediction. They mainly exploit statistical techniques to find a set of users with similar tastes and form the neighbourhood. These methods are analysed in Section 3.4.1
- Model-based: which uses the user's previous ratings to model the user and then makes a prediction. They exploit probabilistic approaches to build the user model, such as Bayesian network, clustering or machine learning techniques like neural networks and several other techniques. These methods are analysed in Section 3.4.2

Generally, collaborative-filtering approaches have advantages and disadvantages, as listed below:

Advantages

- **No need for item attributes**. Recommendations are not based on content and, so, there is no need for the systems to know anything about the items except the ratings being received from the users.
- **Improved accuracy** of the recommendations **over time** as the ratings numbers increase.
- **Serendipity**. Recommendations are not based on item attributes, but on their popularity, so, it is more likely for a user to be recommended a novel item that could not be expected as this item may be liked by like-minded users.
- **No need of human effort** for tagging items with no attributes and no need for the user to have any domain knowledge.

Disadvantages

- **Cold-start problem.** When a new item is added or there are not yet any ratings for an item, then the item is not included in any recommendation list as it is not also included in the user-item ratings matrix. Similarly, if a user has not yet rated any item, it is impossible to find users with similar tastes, because this calculation is based on existing ratings.
- **Data sparsity problem.** In practice, the users rate a small portion of items so the user-item ratings matrix is sparse. As a result, it is difficult to find users with similar tastes.
- **Unusual user problem (grey-sheep).** Sometimes there are users whose opinions are not consistent with any group of users. As a result, these users rarely receive accurate recommendations.
- **Unique tastes problem.** A user having unusual tastes compared to the majority of users would be difficult to correlate with other users and find similar users, which leads to less accurate recommendations.
- **Scalability problem.** Usually, there are millions of users and items, but traditional collaborative filtering algorithms are usually implemented as a centralised website. The computational complexity slows down the prediction for a real-time process.

- **Critical mass of users.** The effectiveness of collaborative filtering algorithms is based on the number of users that rated an item. If an item is not rated by an adequate number of users, the system may not recommend the item.

3.4.1 Memory-based collaborative filtering

Memory-based (heuristics-based) collaborative filtering is based on all the previously rated items by the users. In order to make predictions for items not currently rated, memory-based collaborative filtering exploits statistical techniques to find correlation between users and form a neighbourhood of users by calculating user to user similarity.

Once the neighbourhood is formed, different algorithms are exploited to predict an item or a list of top-N items. Such an algorithm is the weighted sum (Eq. 3.12) where the unknown rating $p_{a,i}$ for a target item i and target user u_a can be predicted from the ratings of the nearest-neighbours (most similar users) of the target user who rated the target item.

The unknown rating $p_{a,i}$ for an item i and a target user u_a is usually predicted as an aggregate (Adomavicius and Tuzhilin, 2005) of the ratings of the N most similar users for the specific item:

$$p_{a,i} = \underset{u \in U^N}{aggr}\, r_{u,i} \qquad \text{(Eq. 3.10)}$$

where U^n is the set of users forming the neighbourhood of target user which rated the item i. Three typical approaches (Adomavicius and Tuzhilin, 2005) that use aggregation are:

The simple average: $$p_{a,i} = \frac{1}{n} \sum_{u \in U^N} r_{u,i} \qquad \text{(Eq. 3.11)}$$

Weighted sum: $$p_{a,i} = k \sum_{u \in U^N} w_{a,u} r_{u,i} \qquad \text{(Eq. 3.12)}$$

The adjusted weighted sum: $$p_{a,i} = \overline{r_a} + k \sum_{u \in U^N} w_{a,u} (r_{u,i} - \overline{r_u}) \qquad \text{(Eq. 3.13)}$$

where k is a normalisation factor and $w_{a,u}$ is the similarity measure between users a and u and can be calculated based on one of the functions of Section 3.2.1. The normalisation factor is used for adapting the predicted rating to the rating scale of the system and is usually defined as: $k = \frac{1}{\sum_{u \in U^N} w_{a,u}}$

Thus the equation (Eq. 3.12) becomes: $p_{a,i} = \frac{\sum_{u \in U^N} w_{a,u} r_{u,i}}{\sum_{u \in U^N} w_{a,u}}$ (Eq. 3.14)

where U^N is the neighbourhood of target user that has rated the target item, $r_{u,i}$ is the rating of user u to item i and $w_{a,u}$ is the similarity measure between users a and u.

Yet this method disregards the personalised way that each user rates. This means that some users tend to give higher ratings while others are stricter. To overcome this problem, the **classic collaborative filtering** algorithm includes the mean ratings of users a and u

$$p_{a,i} = \overline{r}_a + \frac{\sum_{u \in U^N} w_{a,u}(r_{u,i} - \overline{r}_u)}{\sum_{u \in U^N} w_{a,u}} \quad \text{(Eq. 3.15)}$$

where U^N is the neighbourhood of the target user (set of users) that has rated the target item, $r_{u,i}$ is the rating of user u to the item, $w_{a,u}$ is the similarity measure between users a and u and $\overline{r}_a$ and r are the mean ratings of users a and u respectively.

Example 1. Assume a movie recommender system with integer ratings on a scale from 1 (for fully dislike) and 10 (for fully like). Table 3-3 depicts a cut off from the total ratings table. More specifically, the table shows the ratings of five users to six items. An empty cell means that the user has not rated the item. The task of the recommender system is to predict values for the empty cells.

Table 3-3 An example of user-item ratings matrix with averages of ratings

	Item1	Item2	Item3	Item4	Item5	Item6	Average
Alice	8	6	?	5	8	6	6.6
Bob		9		7	8		8
John	5	7	8		6		7
Emily	8	6	9	3		8	6.5
Frank	4	2	7		4	4	4.2

Suppose we want to predict the rating that could give Alice to item3. The first step for the collaborative filtering recommender is to calculate the similarities between the users. Choosing the Pearson Correlation Coefficient (Eq. 3.6), we can calculate the similarity e.g. between Alice (A) and Bob (B):

$$w_{A,B} = \frac{\sum_{i \in R_{A,B}} \left(r_{A,i} - \overline{r_A}\right) \cdot \left(r_{B,i} - \overline{r_B}\right)}{\sqrt{\sum_{i \in R_{A,B}} \left(r_{A,i} - \overline{r_x}\right)^2 \sum_{i \in R_{A,B}} \left(r_{B,i} - \overline{r_B}\right)^2}} =$$

$$= \frac{(6-6.6)\cdot(9-8) + (5-6.6)\cdot(7-8) + (8-6.6)\cdot(8-8)}{\sqrt{(6-6.6)^2 + (5-6.6)^2 + (8-6.6)^2}\sqrt{(9-8)^2 + (7-8)^2 + (8-8)^2}} = 0.3273$$

In the same way, we can calculate the similarities between all the users (Table 3-4).

Table 3-4 User-user correlations

	Alice	Bob	John	Emily	Frank
Alice	1	0.327327	-0.86603	0.953821	0.57735
Bob	0.327327	1	1	1	-1
John	-0.86603	1	1	0.142857	0.438357
Emily	0.953821	1	0.142857	1	0.953821
Frank	0.57735	-1	0.438357	0.953821	1

As a second step, the recommender forms the neighbourhood of Alice as the set of users that rated the item3 that is $U^N = \{John,\ Emily,\ Frank\}$. Therefore, the rating prediction for target user Alice and item3, is calculated as follows:

$$p_{a,i} = \overline{r_A} + \frac{\sum_{u \in U^N} w_{A,u}(r_{u,iten3} - \overline{r_u})}{\sum_{u \in U^N} w_{a,u}} =$$

$$= 6.6 + \frac{(-0.866)(8-7) + (0.9538)(6-6.5) + (0.577)(7-4.2)}{-0.866 + 0.9538 + 0.577} = 7.0$$

This formula (Eq. 3.15) is the most popular technique for predicting recommendations also known as ‘Resnick’s formula’ since it was proposed by Resnick *et al*. (1994). It is a baseline algorithm in collaborative filtering recommender systems and is widely used in both academia and industry.

3.4.2 Model-based collaborative filtering

Model-based collaborative filtering approaches utilise the collection of ratings to learn a model. This model, with its parameters, is then stored and used, instead of the ratings matrix, to produce item predictions. Thus, these techniques have a learning phase which may be time-consuming;, however, the prediction phase is faster than that of memory-based techniques, as they use only the model to compute the predicted rating and not the entire ratings matrix, which is usually too heavy. Various approaches exist to model the user and/or the item, based on probabilistic or statistical techniques:

Bayesian belief networks. A Bayesian network is a directed acyclic graph in which nodes represent attributes and arcs represent dependencies. Each item in the network has a set of parent items, being the best predictors of its votes. In fact, the model represents graphically previous knowledge in a domain. This model is very fast with analogous accuracy with nearest neighbours methods (Breese, Heckerman and Kadie, 1998) and is more suitable when user preferences change slowly in comparison with the time needed for building the model. The great advantage of this approach is that it handles incomplete data well while is quite robust to model overfitting (Ricci *et al.*, 2011).

Clustering. In this method, users are classified into segments, with each segment consisting of like-minded users. In one such study Chee, Han and Wang (2001) introduce a k-means-like algorithm to partition the users in clusters consisting of similar ones and then perform subsequent clustering based on smaller, partitioned databases. Other approaches that cluster users with k-means are those of Ungar and Foster (1998), Sarwar *et al*. (2002) and Xue *et al*. (2005). The main advantage of k-means clustering is their ease of implementation. Generally, clustering methods perform better (Sarwar *et al.*, 2000; Chee, Han and Wang, 2001; Xue *et al.*, 2005) in terms of scalability than common collaborative filtering methods, as their predictions are calculated within a smaller (clustered) amount of data. However, the smaller the group partitions of users the worse it becomes the quality of the recommendations.

Decision trees. This method follows the structure of a tree where branches (edges) are connected with nodes. Internal nodes represent questions and edges represent the answers. The leafs at the end of an edge represent the final decision. Decision trees are learned by recursively splitting the training data into subsets based on an attribute value until these subsets belong to a single class. Typical examples of decision trees can be found in Quinlan (1984), Pazzani *et al*. (1996) and Pazzani and Billsus (1997).

Association rule. This is a usual technique for performing shopping basket analysis. A common case is to detect items being purchased together. A typical rule may be, ‘A mobile phone case is often (82%) bought along with a screen protector.' Such a rule may contribute in capturing item relationships in large-scale sales transactions. Association rules are usually combined with collaborative filtering, when ratings are not on a gradual scale, but are binary ('like', 'dislike').

Matrix factorisation. Dimensionality reduction techniques are applied for minimising the sparsity and scalability problems. These methods extract a set of latent (hidden) factors from rating patterns and capture latent relationships between users and items. Each user and each item have a K-dimension latent factor vector. One popular approach based on matrix factorisation is the **Singular Value Decomposition** (SVD) with its roots in information filtering (Deerwester *et al.*, 1990), but is also extensively applied in collaborative filtering (Sarwar *et al.*, 2000; Rennie and Srebro, 2005; Salakhutdinov and Mnih, 2007; 2008; Koren, Bell and Volinsky, 2009; Yu *et al.*, 2009). The winner of the Netflix Prize was based on matrix factorisation and proved that it is a method very valuable to improving recommendations accuracy. However, it is shown (Sarwar *et al.*, 2000) that, in some cases, the prediction quality was worse than that of memory-based techniques. Eventually, the quality of recommendations seems to depend on the right choice of the amount of data reduction.

3.5 Other methods

3.5.1 Hybrid approaches

Content-based and collaborative filtering are the two basic methods in recommender systems, but, as already pointed out, although each of them has its advantages, they also face a number of limitations. Combining these two methods can alleviate some of their problems, while, at the same time, can take advantage of their strengths, thus, improving recommendation performance. Such systems are called hybrid and one of the earliest ones was Fab (Balabanović and Shoham, 1997), which combined content-based with collaborative filtering approaches to exploit the advantages of both. A survey on hybrid recommender systems can be found in Burke (2002), where the researcher classifies them in the following categories:

- **Weighted**: The predicted rating for an item is computed from the results of all the recommendation techniques that participate in the system. The score of the various techniques of the system is combined, e.g. in a linear way to produce a single recommendation.

- **Switching**: The system uses certain criteria to switch between recommendation techniques. The switching criterion is an additional level of parameterisation and this introduces more complexity in the system.
- **Mixed**: Recommendations from different techniques are presented simultaneously when this is practically possible, e.g. a recommended viewing of a television programme. These systems do not suffer the new-item problem as the combination with the collaborative-filtering technique overcomes the specific problem.
- **Feature combination**: These systems use information from collaborative-filtering as additional feature data, associated with each item and produce recommendations over this augmented data through content-based techniques. Basu *et al.* (1998) report that this technique improves recall this was not also observed for precision.
- **Cascade**: This method follows a staged process where one recommendation technique refines the results of another one. With this method, poorly-rated items, from the second in order technique, are not recommended as the second step involves only on refining the recommendations from the technique of the first step.
- **Feature augmentation**: The result about an item of one technique is incorporated as item feature into the processing of another recommendation technique. This method improves the performance of the core system and makes a significant contribution to the quality of recommendations (Burke, 2002).
- **Meta-level**: In this method when two recommendation techniques are combined, the models of the two techniques are combined in a way that the entire model of the first technique is used as an input feature for the second technique. The advantage of this method is that the model is a compressed representation of a user's interest and this reduces the processing effort for the next model compared to that needed when raw rating data has to be processed.

Apart from the combination of content-based and collaborative filtering, there are also hybrid systems that combine other techniques like item-based collaborative filtering with semantic technologies (Mobasher, Jin and Zhou, 2004; Cantador, Bellogín and Castells, 2008), or knowledge-based techniques (Burke, 2000) and other techniques.

Additionally, researchers report (Balabanović and Shoham, 1997; Pazzani, 1999; Soboroff and Nicholas, 1999; Lucas *et al.*, 2013) that generally hybrid systems provide more accurate recommendations than pure methods.

3.5.2 Community-based

Traditional recommender systems assume that all the users are independent and identically distributed and do not take into account relations between them. Typical approaches in recommender systems ignore the connections between the users, though in real-life we always turn to our trusted friends for recommendations (Ma,King and Lyu, 2011). Moreover, trust in information source plays a key role in making decisions and following a recommendation or not. The emergence of social networks has afforded opportunities for developing new approaches for recommender systems. Recommender systems can take advantage of the existing relationships in the Social Web and build a trusted network. Recently, research has turned to the exploitation of user connections, such as trust relations, and built the so-called trust-based (also trust-aware or social-based or community-based) recommender systems, to simulate the real-world fact that it is common to turn to our trusted friends for advice and recommendations.

These approaches are based on the adage *"Tell me who your friends are, and I will tell you who you are"* and are grounded in social theory about social influence (Marsden and Friedkin, 1993). Community-based recommender systems acquire information about the social relations of the users and model users based on their friends' preferences. Recommendations are based on the ratings provided by the user's friends. These systems are, in fact, a kind of collaborative filtering forming the user's neighbourhood on existing social networks. Obviously, these approaches can be combined with other collaborative filtering and content-based techniques.

The adoption of social information in recommender systems is proved (Massa and Avesani, 2005; Ray and Mahanti, 2010) to be more effective than typical collaborative filtering approaches, thus, several approaches exist in the literature, although more research has to be done as these systems suffer from insufficient number of trust statements, which leads to sparsity of the trust matrix. Trust-based recommender systems are extensively presented in Section 3.6, since the proposed approaches of this study are based on trust connections between users.

3.6 Evaluating Recommender Systems

The ultimate goal of a recommender system is to provide qualitative recommendations for its users. But how can someone actually measure the 'quality' of a recommendation? What are the properties of a recommendation that can determine its 'quality'? Various methods exist for evaluating a recommender system, some of them having their roots in information retrieval and machine learning research areas. Especially for recommender systems, it is crucial to evaluate the system by measuring the user satisfaction regarding the recommended items or the effectiveness of the recommendation engine by increasing the revenue of an e-commerce platform. A recommendation algorithm that claims to produce better results than other algorithms must prove it by adopting common measures and techniques during the comparison process. In the literature, it is observed that recommendation methods can be evaluated either with quantitative measures or with qualitative techniques; however, Herlocker *et al.* (2004) highlight the fact that, despite the numerous and diverse published metrics, there is a lack of standardisation, leading to inability of comparing evaluation results from different publications, affecting the progress of knowledge in recommendation algorithms. However, the majority of researchers (Jannach *et al.*, 2010) evaluate their systems with a small fraction of the available metrics focusing on accuracy (see Section 3.6.2).

3.6.1 Designing the evaluation experiments

Evaluation of a recommender system can be carried out by two general experimental methods: a) offline, which is the most popular and is, in fact, a simulation of the online process, and b) online, in which evaluation is performed on live user interaction sessions. Nevertheless, whatever experimental method is followed, it is important to stick to some basic guidelines followed similarly in general experimental studies (Shani and Gunawardana, 2011):

- **Hypothesis**. Forming the hypothesis is the very first basic step before performing the experiment. In fact, the experiment should be based on this hypothesis, proving its truthfulness or not. So, the measures and the experimental methods to be used are fully dependent on what the hypothesis indicates to be proved.

- **Controlling variables**. When different algorithms are compared on a certain hypothesis, it is crucial that all the controlling variables not being tested remain fixed, otherwise conclusions can be biased.
- **Generalisation power.** When developing an algorithm for a real application, the goal is to generalise the results of the experiments and the conclusions drawn, beyond the specific application or context of the dataset used. Thus, it is important to experiment with different datasets or applications with diverse properties so as to be able to safely generalise the results.

3.6.1.1 Offline experiments

This is the most popular method for evaluating recommendation algorithms whereby no actual users are involved, but only a simulation of the online process. The experiments are conducted with a dataset that is split into a training set and a test set. The recommendation algorithm is firstly trained with the training set and then is validated with the test set. Using the same dataset, experiments can be conducted with various algorithms. In fact, the offline analysis is an opportunity to extensively examine the behaviour of a recommendation algorithm compared to other existing methods at a very low cost and over a short term. Other advantages of offline experiments are the ease to be conducted, not only on various datasets, but also on large datasets, as well as to provide reproducible evaluation results when using the same parameters and datasets and, moreover, better control by allowing tuning of the system parameters. On the other hand, the evaluation in offline experiments is based on predictions for items being already rated, so the typical sparsity of the real-world datasets limits the item predictions that can be evaluated. Another drawback is that 'real' user satisfaction cannot be measured with metrics unless the user him/herself provides actual feedback. To perform an offline experiment, the dataset must be historical data of user interaction from an existing rating system or created from the scratch as synthetic data.

Synthetic datasets

Sometimes it is useful to synthesise datasets for evaluating a recommendation algorithm when no real-world dataset is available. The advantage of synthetic datasets is that parameters of the dataset are fully controlled and this is helpful when there is a need for testing the behaviour of an algorithm against one or more properties. However, it is very difficult to model the user behaviour and, thus, the synthetic dataset can lead to misleading conclusions or may fit better to some algorithms than others and produce biased results. Consequently, synthetic datasets are preferred as an initial step before getting a real-world dataset or when evaluation regards measurements of computational time or even the effect of a dataset property change on an algorithm.

As already mentioned (Section 2.5.3) in this study, the proposed methods are initially evaluated with synthetic data (Section 6.5.1) in order to form an early conclusion about the performance of the proposed methods. The results of this evaluation will indicate the impact of the proposed methods on the performance of recommending items.

Natural or real-world datasets

The most common practice in recommender system evaluation is the offline experiments on a real-world dataset. These datasets are crawled from existing rating systems and, after a pre-process, they are sometimes publicly available for research purposes. A typical example is the MovieLens[8] dataset, one of the most popular among the datasets, pioneered by GroupLens[9] research lab of the University of Minnesota and collected from a non-commercial movie recommendation website also called MovieLens[10] project. The MovieLens dataset is released in three versions of 100K, 1M and 10M ratings. Other popular datasets include:

- Netflix- a large-scale dataset with movie ratings, released for the Netflix Prize competition (Bennett and Lanning, 2007), but no longer publicly available following publicity about research to de-anonymise anonymised datasets.
- Jester-a very dense dataset with only a few items from a joke recommender,

as also Book-crossing, EachMovie and Douban. However, these datasets do not include any social information. Consequently, for evaluating a trust-based recommender system, it is obviously necessary to have a dataset with social information, such as trust-distrust, or friendship relationships. Such datasets are:

- Epinions- from the online product review site Epinions.com, available in various versions (Richardson, Agrawal and Domingos, 2003; Meyffret *et al.*, 2012; Tang *et al.*, 2012) and also available as an extended version with distrust statements and timestamps.
- Flixster- available in various versions (Zafarani and Liu, 2009; Jamali, 2010).

[8] http://grouplens.org/datasets/movielens/
[9] http://grouplens.org/
[10] https://movielens.org/

- Filmtrust- from a trust-based social site, built as an experimental platform (Golbeck, 2006b) whereby users can rate and review movies, also building their trust network.

In the current study, the proposed methods are evaluated on two of the above real-world datasets: Filmtrust and Epinions. These two datasets are described thoroughly in Section 2.5.2.

3.6.1.1.1 Validation process

In order to evaluate a recommendation algorithm with offline experiments, it is necessary, as a first step, to train the algorithm and then validate it. Thus, the dataset is split into the training set, which is used as input for building the model, and the test set, which is used to validate the algorithm by measuring its performance. The split is random and can be of various proportions with the most common being 80/20 for training/testing (Ricci *et al.*, 2011). Then, the random sampling is repeated k times to ensure that measurements are not biased by some user profiles. Finally, the performance results of all the k learned models are aggregated to produce the overall performance of the recommendation algorithm. The process in which the users of the dataset are partitioned in k equal non-overlapping sets (folds) is called *k-fold cross-validation*. More specifically, in *k-fold cross-validation,* when the dataset is partitioned into k sets, one is used for testing and the remaining k-1 are used for training.

In the extreme case that k is equal to the total number of user profiles in the dataset, the validation process is called *leave-one-out*. In this method, each rating of the dataset is hidden and, then, the predicted value is compared with the real rating. Although computationally, this method is quite costly, it allows the algorithm to exploit the maximum amount of data for learning.

As already mentioned (Section 2.5.3), the experiments (Chapter 7) of the current study follow the standard *leave-one-out* as validation process. Thus, the true user-item rating is hidden and a predicted value is calculated for each method we want to evaluate. This process is iterative until all ratings in the dataset are tested.

3.6.1.2 Live user experiments

Online evaluations may be of various forms. A commonly used form is that of user studies. It is a popular method in the human-computer interaction research area for evaluating the usability of an information system (Virzi, 1992; Ivory and Hearst, 2001; Bowman, Gabbard and Hix, 2002). User studies can be carried out in-lab or virtually. Users who participate in the live-experiment are invited to interact with the system. Their behaviour can be observed and recorded via cameras, logs and/or questionnaires for evaluating any quantitative and/or qualitative characteristic of the system that is of interest. This type of experiment, although offering the opportunity for testing well-defined hypotheses under controlled conditions (Herlocker *et al.*, 2004), can still be very expensive to conduct while the number of participants is usually a very small portion and not necessarily a representative sample of the population of the real system. Additionally, there are considerations about ethical issues and also an increased possibility of biased actions of the user who is aware of being observed and recorded. A thorough analysis of online evaluation is out of scope here, but can be found in Shani and Gunawardana (2011).

3.6.2 Accuracy metrics

Recommendations can be either rating predictions or a set of recommendations or an ordered list of recommendations. For this reason, since accuracy metrics have to measure the quality of the produced recommendations, they are divided into three classes, as identified by Herlocker *et al*. (2004), depending on the form of the recommendation: a) **predictive accuracy metrics** for measuring the quality of rating predictions, b) **classification accuracy metrics** for measuring the quality of a set of recommendations and c) **rank accuracy metrics** for measuring the quality of an ordered list of recommendations.

An older study (Sarwar *et al.*, 1998) classified the accuracy metrics according to the method they use, as: a) statistical recommendation accuracy for measuring the closeness between the numerical recommendations provided by the system and the numerical ratings entered by the user for the same items with representative metrics the MAE, RMSE and correlation; and b) decision-support accuracy for measuring how effectively recommendations help a user select high-quality items with the representative metrics being the reversal rate, the ROC sensitivity (Receiver Operating Characteristic curve) and the RPC sensitivity (Precision-Recall Curve).

Various studies can be found (Herlocker *et al.*, 2004; De Wit, 2005; Shani and Gunawardana, 2011) to extensively analyse evaluation methods for recommender systems, but the following sections will focus on the most popular metrics used in recommender systems literature.

3.6.2.1 Predictive Accuracy Metrics

Prediction accuracy metrics are by far the most frequently used measures (Bobadilla *et al.*, 2013) in literature for evaluating a recommendation engine. This is also supported by a short survey (Jannach *et al.*, 2010) that revealed the popularity of accuracy metrics, since the majority of the surveyed studies adopted these measures for evaluating their recommendation methods by conducting offline experiments on historical (real-world) datasets. These metrics measure the quality of the ratings predictions by measuring the accuracy, aka the percentage, of correctly recommended items. Such metrics are the Mean Absolute Error (MAE), the Root Mean Square Error (RMSE) and the normalised values of them.

Mean Absolute Error (MAE)

Mean Absolute Error (MAE) measures the average deviation of the predicted rating values from the actual rating values of all the users and items within a test set. More specifically MAE is defined by the equation:

$$MAE = \frac{\sum_{(u,i)\in Test} \left| p_{u,i} - r_{u,i} \right|}{\left| Test \right|} \tag{Eq. 3.16}$$

where $r_{u,i}$ is the real rating while $p_{u,i}$ is the predicted rating for a pair of user-item u,i and $|Test|$ is the size of the testing set. The lower the MAE is, the more accurately the recommendation engine predicts user ratings.

Herlocker *et al*. (2004) state that besides that the mechanics, of the computing MAE, are simple and easy, it is also a well stated statistical method widely used offering the chance to test two different systems. However, MAE is less appropriate when the granularity of true preference is small.

Evaluating a system with MAE, produces results within the rating scale of the system. Nevertheless, different systems use different rating scales. When comparing two systems that use different rating scales the produced MAEs are not directly comparable. Consider for example the effect and implication of an error of 1.8 in a system with rating scale [1, 5] and the effect on a system with rating scale [-5, 5]. For addressing this deficiency, Goldberg *et al*. (2001) normalised MAE to the rating scale of each system:

$$nMAE = \frac{MAE}{r_{\max} - r_{\min}} \quad \text{(Eq. 3.17)}$$

where r_{max} and r_{min} stand for the highest and lowest rating values of the system respectively. The result of nMAE takes values between 0 and 1 and thus they are comparable between different application scenarios and contexts while the ranking of algorithms remains the same as the ranking given by the MAE.

Root Mean Square Error (RMSE)

Root Mean Squared Error (RMSE) is a measure similar to MAE which emphasises large errors by squaring each individual error and is defined (Herlocker *et al.*, 2004) as:

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in Test} \left(p_{u,i} - r_{u,i}\right)^2}{|Test|}} \quad \text{(Eq. 3.18)}$$

where $r_{u,i}$ is the real rating while $p_{u,i}$ is the predicted rating for a pair of user-item u, i and $|Test|$ is the size of the testing set.

If the predicted ratings of an algorithm do not deviate very far from the real rating, will give lower RMSE values. Lower values of RMSE indicate better accuracy. RMSE was also used in Netflix prize (Bennett and Lanning, 2007) competition as the accuracy measure which was supposed to be improved by 10%.

Like MAE, RMSE must be normalised for a comprehensive comparison of systems with different rating scales. The normalised RMSE (nRMSE) is defined (Ekstrand, Riedl and Konstan, 2007) as:

$$nRMSE = \frac{RMSE}{r_{\max} - r_{\min}} \quad \text{(Eq. 3.19)}$$

where r_{max} and r_{min} represent the highest and lowest rating values of the system respectively.

3.6.2.2 *Classification Accuracy Metrics*

When the goal of a recommender is to identify and produce a set of the n most relevant items for a user, then the accuracy metric has to measure the usage prediction. Classification accuracy metrics measure the frequency of the correct or incorrect decisions about whether an item is good. Considering a recommender system, four possibilities exist for the usefulness of its predictions for a user, as depicted in Table 3-5. Precision and Recall are the two most common metrics with their roots in information retrieval. Additional to these two metrics, PRC and ROC curves are also common in the literature.

Table 3-5 Possible results of the usage of a prediction

	Recommended	**Not recommended**
Used	True-Positive (TP)	False-Negative (FN)
Not used	False-Positive (FP)	True-Negative (TN)

Precision

Precision is defined as the ratio of the relevant items recommended to the user's need. In other words, it measures the effectiveness of the recommendation represented by the probability that a recommended item is relevant. Given the possibilities of Table 3-5 *precision* (blue dashed rectangle) is defined (Ricci *et al.*, 2011) by the following equation (Eq. 3.20) as the fraction of the number of the successful recommendations *(TP)* to the total number of recommended items *(TP+FP)*.

$$precision = \frac{|TP|}{|TP| + |FP|} \qquad \text{(Eq. 3.20)}$$

Recall

Recall is defined as the ratio of the actual set of relevant items which have been correctly classified as relevant. Given the possibilities of Table 3-5 *recall* (red dotted rectangle) is defined (Ricci *et al.*, 2011) by the equation (Eq. 3.21) as the fraction of the number of the successful recommendations *(TP)* to the total number of useful items *(TP+FN)*.

$$recall = \frac{|TP|}{|TP| + |FN|} \qquad \text{(Eq. 3.21)}$$

Precision and Recall combined

Precision and Recall do not give enough information about accuracy when used separately. Moreover, it is a general rule that precision and recall are inversely related; increasing the number of recommended items, recall is increased while precision is decreased. Therefore, several approaches combine the two above metrics into one single metric. One of them is the $Fmeasure$ used for evaluating recommender systems (Sarwar *et al.*, 2000) given by the equation (Eq. 3.22). This measure equally weights the two metrics and produces results between 0 and 1.

$$Fmeasure = \frac{2 \times precision \times recall}{precision + recall} \qquad \text{(Eq. 3.22)}$$

Another method combining the two metrics is to use the Precision-Recall Curve (Sarwar *et al.*, 1998) also called PRC sensitivity, which is a plot depicting the proportion of the recommended items being preferred. Similarly, another curve, with its roots in signal detection theory, called Receiver Operating Characteristic (ROC) is a curve depicting the *Recall (sensitivity)* against the complement of *specificity (FP rate)* as described by Herlocker *et al*. (2004) given by (Eq. 3.23)

$$FP\ rate = 1 - Specificity = \frac{|FP|}{|FP| + |TN|} \qquad \text{(Eq. 3.23)}$$

Although ROC curves are good for evaluating the performance of certain algorithms it is, however, difficult to produce clear results when comparing large numbers of algorithms.

3.6.2.3 Rank Accuracy Metrics

It is common in recommender systems to produce ordered lists of recommended items, also known as top-N recommendations. In this case, recommendation algorithms do not predict a particular rating for each item, but, rather, they order the items according to the user's preferences. Rank accuracy metrics measure the closeness of these ranks to the way that user would order the same items. Such metrics are: a) the Prediction-Rating correlation, by using Pearson or Spearman rank correlation coefficients; b) the half-life utility metric (Breese, Heckerman and Kadie, 1998) measuring the expected utility of a rank list to the user based on the observation that users tend to look only at highly ranked items; and c) the Normalised Distance-based Performance Measure

(NDPM) metric (Balabanović and Shoham, 1997) used to compare two weakly ordered lists. Analysing these metrics here is out of scope since the hypothesis of our approach refers to the rating prediction accuracy; however, more details on these measures can be found in the related literature (Herlocker *et al.*, 2004; Shani and Gunawardana, 2011).

3.6.3 Other measures

Beyond the accuracy metrics, there are also other quality characteristics of a recommender system that can be measured. One of the characteristics usually mentioned in the literature is coverage, which is the percentage of items for which the system can make a prediction. Herlocker *et al.* (2004) identified two types of coverage, *prediction coverage* and *catalogue coverage*. Prediction coverage refers to the percentage of items for which the system can provide predictions, while catalogue coverage is the percentage of available items being ever recommended to users. Similarly, Shani and Gunawardana (2011) refer to three types of coverage, specifically, *'item space coverage',* which is the same as the prediction coverage; *'user space coverage',* which is the percentage of users or user interactions for which the system can provide recommendations; and *'cold-start*', which is, in fact, a subproblem of coverage and is the performance of the system on new items or users. However, coverage is a measure that, much like precision and recall, cannot be considered separately. Consider for example the implications of a system that achieves very high coverage by providing predictions for all available items and/or users, but against accuracy. Therefore, coverage is a measure being always considered in conjunction with accuracy. Apart from coverage, Herlocker *et al*. (2004) also denote some other characteristics that can be measured for evaluating a recommender system, specifically:

- Confidence: the system's trust in its recommendations or predictions reflecting how helpful the system is for the users to make more effective decisions.
- Learning rate: how quickly an algorithm can produce good recommendations.
- Novelty: whether a recommendation is a novel possibility for a user.
- Serendipity: how surprising for the user are the successful recommendations and can be measured as the amount of relevant information in a recommendation being new to the user.

Furthermore, Shani and Gunawardana (2011) identified the following as additional metrics:

- Trust: the user's trust in the system recommendation.
- Diversity: the opposite of similarity. It is useful when we are interested in presenting the user with a diverse set of recommendations.
- Utility: the value that either the system or the user gains from a recommendation. However, diversity and serendipity can be seen also as different types of utility functions.
- Risk: the potential risk for a user of selecting the recommendation.
- Robustness: the stability of the recommendation in the presence of fake information.
- Adaptivity: when the item collection in the system changes rapidly, or where trends in interest over items may shift, it is important for the system to be adapted rapidly.
- Scalability: the system must scale up to real datasets, which usually consist of millions of items, without affecting accuracy or coverage.

Figure 3-2 depicts all the measures presented in this chapter.

As already mentioned (Section 2.5.3), in the current study the experiments for measuring and evaluating the quality of rating predictions of various methods use the **predictive accuracy metrics**, since the proposed methods produce item ratings. Moreover, coverage is another measure being used in the experiments, which is also considered in conjunction with RMSE, forming a new measurement, the FMeasure, as described in Section 2.5.3.

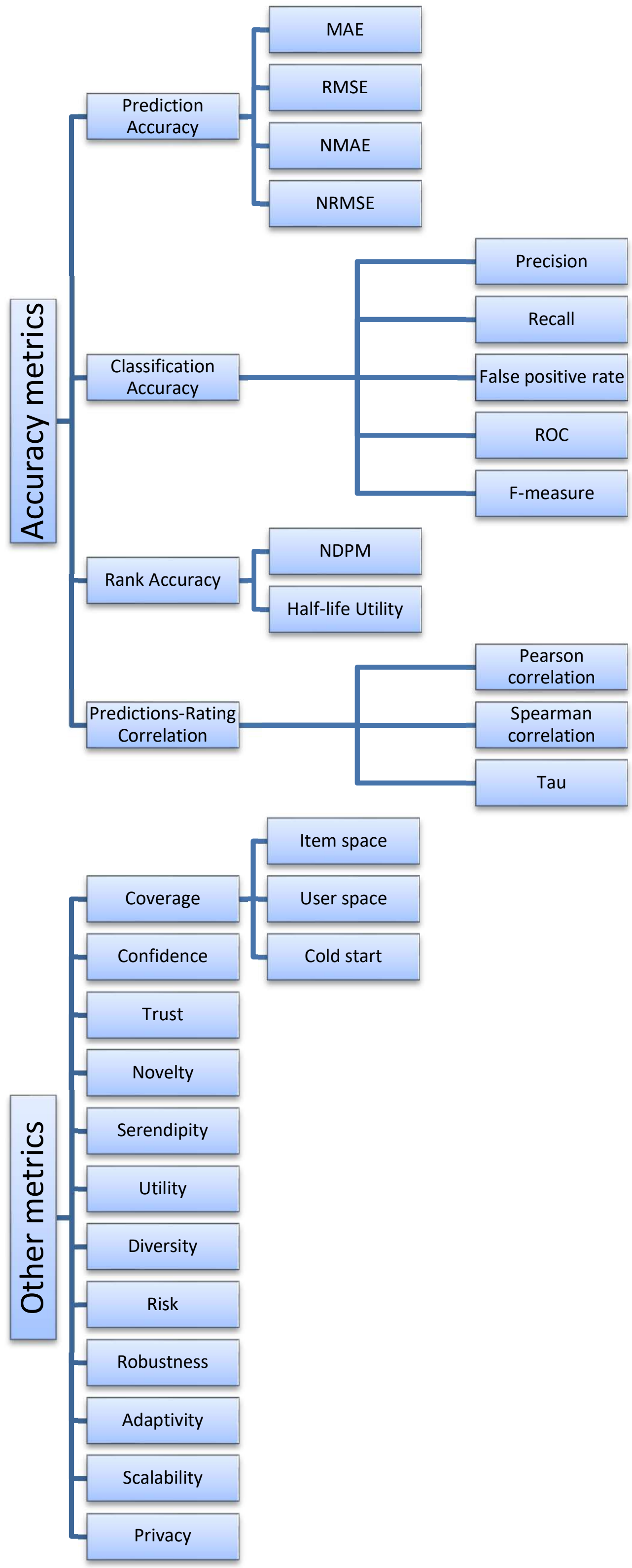


Figure 3-2 Evaluation metrics for recommender systems

3.7 Conclusion

As an evolution of various methods from different disciplines, to deal with the information overload problem, recommender systems adopt various techniques to filter information and produce suggestions for users. This chapter presented the various approaches for producing recommendations and analysed the two major methods of recommender systems (content-based and collaborative filtering) demonstrating the way they work with the basic algorithms, their advantages, but also their limitations (Table 3-6). Additionally, this chapter examined the various methods for evaluating recommender systems, pointing out the methods that this study uses in Chapter 7 and Section 6.5.1. More specifically, this study uses offline experiments for evaluating the proposed approaches with both synthetic and real-world datasets with leave-one-out validation and predictive accuracy metrics, as described also in Section 2.5.

Table 3-6 Comparison of content-based and collaborative filtering methods

	Advantages	Disadvantages
Content-based	• No new-item problem • No need of data about other users • Enough explanations • Privacy preservation	• Cold-start for new users • Attribute extraction • Overspecialisation • Objectivity
Collaborative filtering	• No need for item attributes • Improved accuracy • Serendipity • No need of human effort	• Cold-start for new users and new items • Data sparsity • Unusual user (grey-sheep) • Unique tastes • Scalability • Critical mass of users

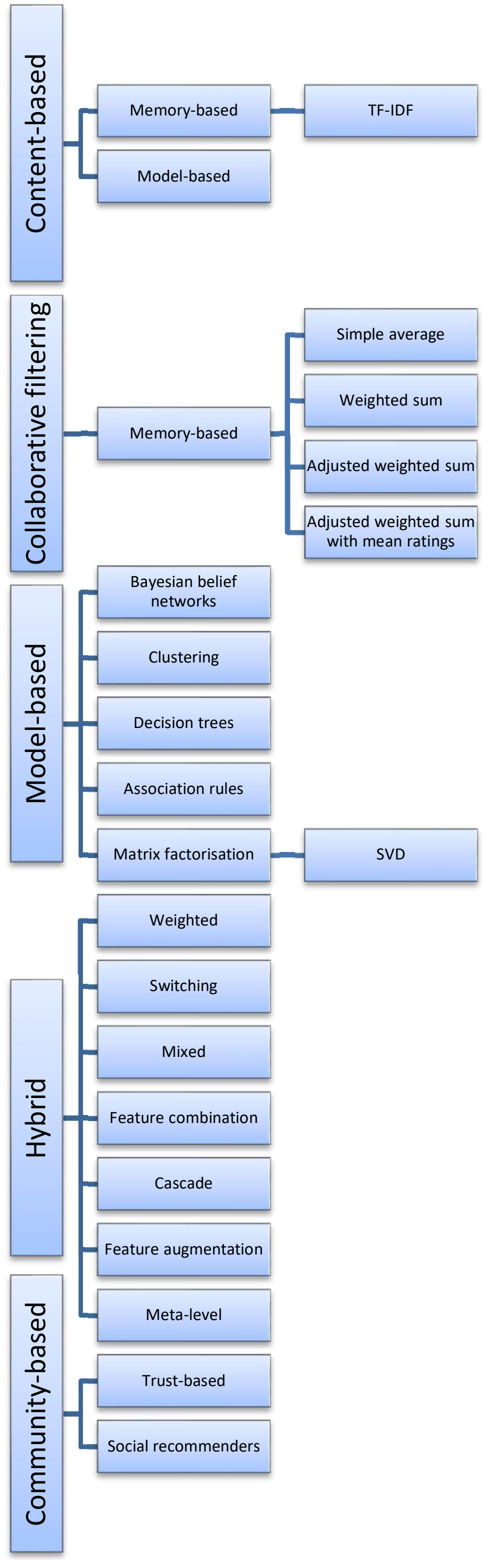


Figure 3-3 Recommendation approaches

Chapter 4

Trust-based Recommender Systems

A solution to overcome many of the limitations of traditional recommender systems, which will be analysed later in Section 5.1, is to enhance recommender systems with trust relationships which already exist in social networks and build the so-called trust-based or trust-aware recommender systems. While both collaborative and content-based filtering recommendation approaches are based on similarity measures, either between users or between items, respectively, trust-based systems use the weight of a trust relationship to measure the similarity between users. Trust relationships are weighted and can be expressed in various scales for stating trust or distrust or even intermediate degrees of trust. Liu *et al*. (2004) state that trust is positively related to whether an individual will purchase again, revisit a site, recommend the site to others and make positive comments about the site. Many of the major problems, which will be discussed later in Section 5.1, can be alleviated by incorporating trust in recommender systems (Victor, Cornelis and DeCock, 2011), although, in many trust-based approaches, the cold-start problem recurs, since a new user does not yet have connections. The enhancement of trust through existing connections in social networks also raises the problem that trust and similarity are not the same, and can be used complementary (Ziegler and Golbeck, 2007). Generally, research effort is needed in modelling the user for producing more accurate and qualitative recommendations, as will be analysed in Chapter 5.

4.1 Definition of Trust

Trust is vital for interpersonal relationships to be built in various fields, as it plays a key role in any interaction procedure in human societies. Thus, trust can be found as a notion in many disciplines, since it impacts social and network relationships, enables cooperation and can increase business revenue by reducing risk costs.

Trust as a field of research is extensively explored in sociology, social psychology, cognitive psychology, economics, political sciences, organisation science and computer science, among many others. In **social sciences**, trust is a factor that impacts human decisions and is explored for its structural components. In **economics,** trust enables people to do business with each other, which, in turn, affects the economy of a country (Harford, 2010) and is viewed as a calculative or institutional concept. Especially in **computer science,** trust is broadly used in IT security as identity verification and authentication for network access control or as a metric for the reliability of a source. The term 'web of trust' has its origins in authentication, too. Moreover, a source or an agent, in an electronic transaction, must be considered as trusted so as to be completed safely. Trust and reputation are also important in organisations for doing or extending business. In addition, employers' trustworthiness is a key issue for organisational trustworthiness too.

Wierzbicki (2010) makes a distinction between *human trust* and *computational trust*. Human trust is studied in psychology, economics, anthropology, sociology, etc., and refers to the mental state of humans. On the other hand, computational trust refers to trust that can be computed or represented, usually by modelling the human trust, so as to be used in trust management systems. The author defines trust management systems as decision support systems in which an agent considers a situation and makes a decision about the choice of an action *"by establishing trust or distrust in another agent on whose actions the decision maker's outcome depends."*

Although there are many different definitions of the trust concept, according to its scientific origin and the application domain, there has not been any agreed generic or cross-discipline definition. In fact, most definitions are oriented to the context and the research problem of any study and can be found as both a noun, for describing personality trait or social structure, and a verb, for describing belief or behavioural intention.

In social psychology, researchers view trust as a concept participating in situations of uncertainty while contributing to relationships. Deutsch (1958) gives one of the first definitions in social psychology and defines trust as a behaviour where *"an individual may be said to have trust in occurrence of an event if he expects its occurrence and his expectation leads to behaviour which he perceives to have greater negative motivational consequences if the expectation is not confirmed than positive motivational consequences if it is confirmed."*

One of the first probabilistic views of trust is that of Gambetta (1988), where trust is defined as *"a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action."* Similarly, Lehman and Sztompka (2001) provide another probabilistic view of trust defining it as *"a bet about the future contingent actions of others"*, while Mayer et *al.* (1995) focus only on trust as a decision and define trust as *"the willingness of a party to be vulnerable to the actions of another party based on the expectation that the other party will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party."* A sociological definition of trust describes how trust operates in society (Luhmann, 2000), where trust *"is a solution for specific problems of risk… and …an effective form of complexity reduction."*

A widely used definition in literature is that of Mui *et al*. (2002) defining trust as *"a subjective expectation an agent has about another's future behaviour based on the history of their encounters."* Another popular definition is that of McKnight and Chervany (2001), which refers to *"the willingness or intention of a person to depend on the other person generally and not in a specific situation, even though they were aware of potential problems in their relationship"* and identifies four attributes components for making a trust decision:

1. Benevolence: acting in one's interest aside from an egocentric profit motive
2. Integrity: fulfilling promises, being moral, telling the truth
3. Competence: ability to accomplish a specific task
4. Predictability: the degree to which the trustee's actions can be forecasted in a given situation

As another point of view, Jøsang and Lo Presti (2004) refer to trust as *the extent to which one is willing to depend on somebody in a given situation*, while Falcone and Castelfranchi (2001) give a socio-cognitive model of trust, aka a model based on the cognitive and social nature of trust.

Especially in computer science, Marsh (1994) introduced trust as a computational concept based on Deutsch's observations of how trust is used. He states that "*trust is a measurable level of risk, through which an agent X assesses the likelihood that another agent Y will successfully perform a particular action, both before X can monitor such action and in a context in which it affects its own actions*." Marsh also introduced distrust as the negative trust, which was later considered by many researchers (Lewicki, Mcallister and Bies, 1998; McKnight and Chervany, 2001; Yuan *et al.*, 2010; Victor, Cornelis and DeCock, 2011; O'Doherty, Jouili and Van Roy, 2012).

However, some researchers (Dokoohaki and Matskin, 2008) tried to converge some of the above definitions and give a more generic and context-neutral definition, such that *"trust is a complex issue, relating to fairness and straightforwardness, honesty and sincerity of a person or the service this person might offer."*

Finally, trust as a relationship is a need coming from the interdependence between humans. It is common to seek help for things we are not versed in and we rely on the experience of others to estimate whether we will move into an act. Of course, this involves an element of risk, which depends, among others, on the estimation of how reliable is the other person, aka the degree of trust. Thus, trust represents the willingness of the trustor to be vulnerable under conditions of risk and interdependence.

In the current study, we will focus on the computer science aspect of trust and, particularly, on how it is used and computed in intelligent systems where trust is referred to as a belief: *"the trustor believes that the trustee can be trusted for a specific goal in a specific context"* (Adali, 2013).

Hence, we define trust as a *"relationship between two agents namely the trustor and the trustee where the trustor trusts the trustee in a specific context."* For example, Alice trusts Bob in fixing her car. The role of context in a trust relationship is of major importance, e.g. Bob trusts John as a dentist, but does not trust him as a driver.

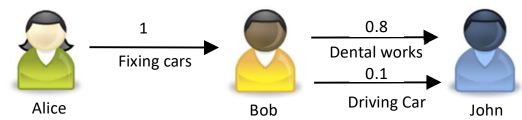


Figure 4-1 A typical trust network

However, there is an important issue about trust to which we also have to pay attention. Trust in a person is different to trust in a person's recommendation; that is, Bob may trust Alice as a cook, but may not trust her for recommending him a restaurant. In the first case, trust refers to the ability of an agent to perform an action adequately (trust in performance) and, in the second case, trust refers to the ability of the agent to recommend other agents to perform an action adequately (trust in recommendation). Although this is a very important distinction, for compatibility with the traditional way users participate in social networks, both of these ideas are represented as a single value, as is the common practice (Golbeck, 2005) in computer science.

Trust as a concept, is sometimes confused with confidence as these two terms have very close meanings in the English dictionary. However, in technical meaning, confidence is a factor impacting trust. Although Deutsch (1958) sees *"trust as confidence"*, Luhmann (2000) reports that trust presupposes an element of risk, whereas confidence does not. Moreover, Castelfranchi and Falcone (2010) point out that confidence is a broader view of trust, distinguishing these two terms as *"confidence in* Y *(a person) that will be able to do an action* a *appropriately"* and *"confidence that* g *(a specific goal) will be achieved"*. In the trust literature, the concepts of trust, trustworthiness and trust propensity are also distinguished. On the one hand, trustworthiness is the ability, benevolence and integrity of a trustee and, on the other hand, trust propensity is a dispositional willingness to rely on others, while trust is the intention to accept vulnerability to a trustee based on positive expectations of his or her actions (Colquitt, Scott and LePine, 2007).

Another concept to which trust is closely related is that of reputation. Mui (2002) defines reputation as *"the perception that an agent has of another's intentions and norms"*, while he supports Granovetter's (1985) point of view in which reputation is a 'social quantity' calculated based on actions by a given agent a_i and observations made by others in an 'embedded social network' in which a_i resides.

Recently, Wierzbicki (2010) gave an abstract definition for reputation as the *"information about the trustee that is available to the trustor and is derived from the history of the trustee's behaviour in some contexts",* which can be represented as a probabilistic model. Barber and Kim (2001) define reputation as *"the amount of trust an agent gives an information source based on previous interactions among them,"* When an information source delivers trustworthy information, and satisfies the needs of other agents, then these agents may increase the reputation of the information source. On the contrary, no satisfaction of the other agents may decrease the reputation of the information source. In the same way, a user's reputation clearly affects the amount of trust that others have towards them (Mui, 2002).

When trust as a notion is considered within security systems or Peer-to-Peer (P2P) networks, then it is closely related to reputation. Many websites are based on reputation, such as Ebay.com and Google, which rank the web pages as a search result, based on the PageRank (Brin and Page, 1998), which is a reputation-based algorithm. The concept of reputation is often used to measure the global trust of an agent within a network and will be analysed later in Section 4.4.2.

4.2 Computational properties of trust

Trust as a concept has properties which computational processes exploit to infer trust relationships. This section presents the functional properties of trust that computational trust models and which are based on for trust dissemination:

Asymmetry and subjectivity*.* Trust is typically asymmetric. Such that Alice trusts Bob to fix her car, but Bob doesn't trust Alice to fix his car, while Alice and Bob may fully trust each other for recommending a restaurant. Hence, trust, as a rule, is directed and asymmetric, although, sometimes, can be symmetric, as it may happen for two parties to trust each other. Then, the trust is mutual, although the degree of trust for each relationship is not necessarily the same. For example, Alice may trust 10/10 (fully) Bob for recommending her a restaurant, but Bob trusts 8/10 (not fully) Alice for recommending him a restaurant. Asymmetry occurs due to the fact that people may have different perceptions, beliefs or expectations. Two people may have different opinions about the same person or group of persons or topics. Consider the Brexit referendum where British people divided into two parties for remaining or not in the European Union. Thus, trust is also personal and subjective.

***Propagation**.* In real life, if we trust a friend, we also tend to trust the friend of our friend. For example, if Alice trusts Bob and Bob trusts Frank, then Alice can derive some conclusion about the degree of trust she can have about Frank, based on the degree of her trust for Bob and the degree of Bob's trust for Frank. So, in a social network, trust information can be propagated and create trust chains. By propagating trust on a social network, we can infer more trusted persons and, hence, improve the predictive performance of recommender systems by building a bigger trust network. Propagation will be discussed in detail in Section 4.3.2.

***No-Transitivity**.* Generally, trust is not transitive (Sherchan, Nepal and Paris, 2013). Suppose Alice trusts Bob and Bob trusts Frank, this does not necessarily imply that Alice will trust Frank. Although trust is propagative, as stated before, this does not imply that it is also transitive. In the literature, many times transitivity and propagation are confused, although the propagative property is really concerned and extensively researched as a computational property of trust.

***Composability**.* When there is not direct trust for an agent and trust information is propagated from more than one source, then there is a need to compose all the propagated trust information in one trust score. Let's say Alice does not know anything about Frank and she receives information from her friends, Bob and Jenny, about Frank's trustworthiness. Alice has to combine the suggestions from her friends to make a conclusion about his trustworthiness based on her own degree of trust for each of her friends. Of course, information from different sources can be contradictory. The mechanism exploiting the composability property is the trust aggregation.

Dynamicity. Trust is time dependent and can change due to various reasons. Positive or negative evidence or experience plays a key role, not only to the initial establishment of a trust relationship, but also to the gradual change of the already established relationship. For example, Alice trusts Bob 7/10 for recommending her a restaurant, but the four last times that Bob recommended her some restaurants eliminated any remaining doubts about him and now she trusts him 10/10 for recommending her a restaurant. On the other hand, although Bob trusted Frank 9/10 to fix his car, as he had a long-time experience of his work, after the last time, when Frank stated that he had fixed a problem with the engine, but the car broke down, Bob lost his confidence in Frank. Thus, his trust was dramatically decreased and now he does not trust Frank as much (just 3/10). In fact, trust degree may decrease or increase due to negative or positive evidence, respectively, and it is commonplace that it is easier to crash than to build a trust relationship.

Context dependency. Trust is context and time dependent (Adali, 2013). A person being trustworthy, i.e. as a restaurant recommender, may not be trustworthy enough as a movie recommender. Preferences also change over time. Suppose a user searching information and recommendations to buy a car. As soon as the purchase is completed, the user may not be interested again for information regarding cars for many years thereafter. Moreover, information is sometimes location dependent. Think of a person travelling across a country and, on the way, is looking for tourist information for every new place that he/she visits. Then, a recommendation from a native will have more weight than that of a stranger.

4.3 Trust inference

One of the main challenges in trust-based approaches is to expand the personal trust network of a user by inferring new trust relationships. Trust inference is the mechanism via which a trust relation can be established between two nodes not being yet connected. This mechanism is implemented through a trust inference algorithm or, in other words, a 'trust metric', recommending an unknown trust value from one user to another. The inference algorithm applies the computational properties of trust on the information provided by the existing trust network or on other information that can imply relations between users with no use of any trust network information. The implicit trust, then, can be calculated from similarity metrics, as in the typical recommender systems presented in Section 2.1, that use information from ratings or items. Several methods exist for producing recommendations based on implicit trust (Yuan *et al.*, 2010; Kim and Kim, 2012; Martín-Vicente, Gil-solla and Ramos-Cabrer, 2012; Htun and Tar, 2013; Guo *et al.*, 2014), but none of these take advantage of the computational properties of trust, such as propagation. Moreover, these methods are just variations of the typical recommendation methods, whilst it is demonstrated (Golbeck, 2005; Ray and Mahanti, 2010) that incorporating information from trust networks provides more accurate recommendations than that from typical recommender systems.

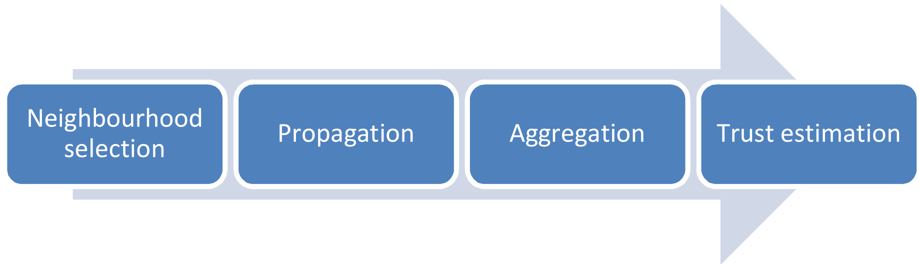


Figure 4-2 Process to infer trust

The process that trust metrics follow to infer a new trust relationship between two users is depicted in Figure 4-2. The first step is to form the user neighbourhood. When the system is based on explicit trust statements, the algorithm reads the trust network and selects the trustees of the user. These trustees constitute the user's neighbourhood. The step of propagation is optional, depending on the trust model, and its objective is to expand the existing trust network by dissemination of trust. This step is based on one of the main computational properties of trust presented in Section 4.2 and which will be analysed later in Section 4.3.3. The next step of a trust metric is to aggregate the trust information being collected from the user's neighbourhood. Usually, new trust relationship can be inferred through the different trust paths which have to be aggregated to produce a single outcome. The process of trust aggregation will be analysed in Section 4.3.4. Finally, the last step of a trust metric is to calculate the trust value of the inferred relationship. Trust weight can be estimated using various formulas, as will be presented in Section 4.2.

4.3.1 The small-world phenomenon

It was in 1967 when Stanley Milgram, a social psychologist at Harvard, introduced the 'Small-World problem;, which was formulated as: *"starting with any two people in the world, what is the probability that they will know each other?"* Moreover, taking into account that, although two persons may not know each other directly, they may share a mutual acquaintance, he went a step further and refined the above question as *"given any two people in the world, person X and person Z, how many intermediate acquaintance links are needed before X and Z are connected?"* With his unique experiment, Milgram showed that any two people in the US were separated by approximately six degrees and that is why the 'small-world phenomenon' is also known as 'six degrees of separation'. The scenario of his experiment involved source individuals in Nebraska who had to deliver a letter to a target individual in Massachusetts (a distance of more than 2000km). The source individuals were provided with some basic information about the target person and were informed that their task was to reach this person not directly, but through intermediates. They were asked to forward a packet (initially given to them by the researcher) to an acquaintance whom they knew on a first name basis and by which they believed it was more likely to reach the target person. The researcher created traceable chains to the target person through his experiment and finally proved that an average of five intermediates was needed for the source person to reach the target person, which makes a ‘six degrees of separation.

Many years later, an experiment was conducted (Dodds, Muhamad and Watts, 2003) to investigate the ‘small-world phenomenon’ in the Internet era. The experiment then was much easier to include persons across the world, since the task (similar to Milgram’s) was to reach the target (80 targets instead of the single one of Milgram’s experiment) through e-mail. The experiment involved about 60,000 participants and approximately 24,000 message chains. The extremely low chain completion rate (only 384 of 24,163 chains reached their targets) gave a misleading number of four for average path length (L=4.05); however, refining the results, they measured the median path length as $5 \leq L \leq 7$, confirming Milgram’s ’six degrees of separation’.

Research on the dynamics of small-world (Watts and Strogatz, 1998; Watts, 1999) indicate that social networks have the same structural properties of small-world networks: a) the characteristic path length, which is the shortest path between two nodes, and b) the clustering coefficient, which measures the cliquishness of a typical neighbourhood (the fraction of edges between neighbours of the node, that actually exist, over the maximal number of possible edges). Characteristic path length is a global property of a network, while clustering coefficient is a local property. In small-world networks, the characteristic path length is small, even in huge networks such as Facebook, while the clustering coefficient is high. In other words, in social networks, where users form groups and are highly connected, the shortest path between two nodes in the network is small.

Additionally to the above two characteristics, Kleinberg (2000) supported that, in a network, an entity can use local information to find short paths, even with limited knowledge of the entire network. Additionally, Kleinberg developed a decentralised algorithm capable of finding short paths and showed that there is a unique model of the ones he developed, in which the algorithm is effective. On the other hand, Watts and Strogatz (1998) demonstrated that small-world networks can be generated by randomly rewiring a small number of nodes in a regular network, like a lattice.

4.3.2 Trust prediction as a link prediction problem

Trust prediction is a link prediction problem which is a common research problem investigated by various researchers (Liben-Nowell and Kleinberg, 2003; Cui, Wang and Zhai, 2010; Easley and Kleinberg, 2010; Aiello *et al.*, 2012; Lou *et al.*, 2013; Symeonidis and Tiakas, 2014; Ciotti *et al.*, 2016) and refers to the problem of inferring missing links in an observed network. It can be found within various application areas, such as website link prediction (Zhu, Hong and Hughes, 2002), protein-protein interaction prediction (Airoldi *et al.*, 2006), detection of duplicate records in databases (Elmagarmid, Ipeirotis and Verykios, 2007), recommender systems (Ziegler and Lausen, 2004; Avesani and Massa, 2005; Golbeck, 2005; Kuter and Golbeck, 2007; Ma, King and Lyu, 2011; Htun and Tar, 2013) etc. The link prediction problem is usually modelled as a supervised classification problem (Aggarwal, 2011) and is generally relied on measures of similarity based on the available information -features- of the graph. These features can be structural, regarding the network topology, or contextual, regarding behaviours, activities and characteristics of the nodes. Moreover, Aiello *et al*. (2012) support that a social link prediction can rely only on the topical similarity between users.

Trust is a social relationship being established when risk between two ties is reduced. Since similarity and, therefore, homophily (a notion that will be discussed later in Section 6.1), reduces the risk of building associations between people and simultaneously increases the possibility to establish connection, it comprises the basic factor for building trust. Thus, trust prediction within trust networks is mainly based on measuring the similarity between users (Portes and Sensenbrenner, 1993; Banks and Carley, 1996), just as in link prediction for social networks. As will be analysed later in Section 4.2, most algorithms to infer trust are based on similarities of structural characteristics of the trust graph. Although trust prediction can be based also on contextual features, it is evidenced (Zolfaghar and Aghaie, 2012) that the performance of trust predicators based on structural features outperforms the algorithms based on contextual features. Moreover, Shang *et al.* (2009) claim that structural-based similarity produces better results than Pearson Correlation Coefficient when the dataset is sparse.

Several structural similarity, also called proximity, indices exist based on graph theory and Social Network Analysis, categorised in various ways by researchers, depending on the method they use, such as supervised vs. unsupervised methods (Tang *et al.*, 2013), or depending on the features, such as the categorisation of Lu and Zhou (2010) in local, global and quasi-local indices or that of Hasan and Zaki (2011) who distinguish between neighbourhood-based and path-based indices.

Below are given three of the most popular node similarity indices, known also as neighbourhood-based (Hasan and Zaki, 2011). All the three indices are based on the intuition that the more common neighbours between two nodes, the higher is the similarity between them. So, for two nodes x and y in the graph, let $\Gamma(x)$ and $\Gamma(y)$ denote the set of neighbours of each node, respectively, the three indices are described as follows.

Common neighbours

Two nodes x and y are more likely to have a link if they have many common neighbours.

$$sim^{CommonNeighbours}(x,y) = \left|\Gamma(x) \cap \Gamma(y)\right| \qquad \text{(Eq. 4.1)}$$

It is the simplest form of similarity between two nodes and is defined as the number of the common neighbours. Newman (2001) exploited this measure in the context of collaboration networks to verify that there is a correlation between the number of common neighbours of two users and the probability of collaborating in the future.

Jaccard coefficient

Jaccard coefficient is a very old similarity metric (Jaccard, 1901) defined as:

$$sim^{JaccardCoefficient}(x,y) = \frac{\left|\Gamma(x) \cap \Gamma(y)\right|}{\left|\Gamma(x) \cup \Gamma(y)\right|} \qquad \text{(Eq. 4.2)}$$

In fact, it is a normalised version of common neighbours and it measures the probability that a common neighbour of nodes x and y would be selected, of a randomly selected neighbour that either x or y has.

Adamic/Adar

The original Adamic/Adar index (Adamic and Adar, 2003) is a metric for computing the similarity between two sites. Liben-Nowell and Kleinberg (2003) customised this index to compute the similarity between two nodes as follows:

$$sim^{AdamicAdar}(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log|\Gamma(z)|} \qquad \text{(Eq. 4.3)}$$

This index assigns more weight to less-connected neighbours and penalises the high degree nodes.

Liben-Nowell and Kleinberg (2003) investigated several different similarity indices for link prediction in social networks and evaluated them for their performance on various real datasets. The results of this study indicate that simple indices like Adamic/Adar and common neighbours outperform more complex indices which consider longer paths.

4.3.3 Propagation

In Social Network Analysis, propagation phenomenon is used to examine how information, diseases or rumours and fads spread across a social network. In addition, propagation is one of the most common methods to infer trust relationships between users not yet connected in a social network. In real life, if we trust a friend, we also tend to trust the friend of our friend. For example, if Alice trusts Bob and Bob trusts Frank then Alice can derive some conclusion about the degree of trust she can have about Frank based on the degree of her trust for Bob and the degree of Bob's trust for Frank.

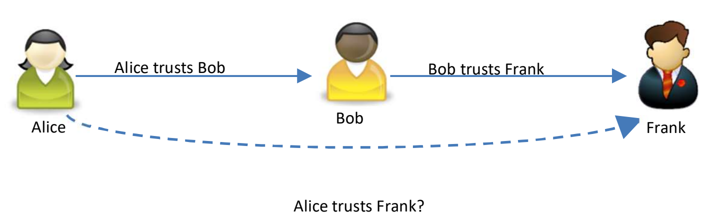


Figure 4-3 An example of direct propagation

However, some authors (Golbeck and Hendler, 2006; Sherchan, Nepal and Paris, 2013) state that trust is not perfectly transitive in the mathematical sense, such as if Alice highly trusts Bob and Bob highly trusts Frank, it is not necessarily always true that also Alice highly trusts Frank. In the literature, many times transitivity and propagation are confused; although transitivity may imply propagation, this does not mean that propagation implies transitivity. Moreover, Bhuiyan (2013) distinguishes between the ability to provide recommendations, *'referral trust',* and the ability to perform an action, *'functional trust'.*

Consequently, the propagative property is really concerned and extensively researched (Gray *et al.*, 2003; Guha *et al.*, 2004; Ziegler and Lausen, 2004; Quercia, Hailes and Capra, 2007; Kim and Song, 2011; Victor, Cornelis and DeCock, 2011; Chakraborty and Karform, 2012; Hamdi *et al.*, 2013) as a computational property of trust. In a social network, trust information can be propagated and create trust chains. By propagating trust on a social network, we can infer more trusted persons and, hence, improve the predictive performance of recommender systems by building a bigger trust network. An illustration of the direct propagation mechanism can be seen in Figure 4-3.

Guha *et al*. (2004) extensively investigated the trust and distrust propagation mechanism and proposed a trust metric combining a 'basis set' of atomic propagations. Representing with M a set of beliefs, in other words, the trust matrix, the four atomic propagations are operations on this matrix M.

- **Direct Propagation:** Trust is propagated along an edge. If A trusts B and B trusts X then A trusts also X. The direct propagation means as matrix operation that the new matrix contains all paths with the length 2 of the initial belief graph, i.e., M^2.
- **Co-Citation:** A user trusts those users who are trusted by users providing similar trust ratings as him/herself. If A and B trust C, and A trusts X then B will also trust X. The matrix operation is $M^T M$.
- **Transpose Trust:** if A trusts X then X will start to trust A to some extend. It is represented in the belief matrix as M^T.
- **Trust Coupling:** if A and X trust the same users then trusting A should imply trusting also X. The matrix operation is MM^T.

These four atomic propagations are weighted and combined to a single matrix $C_{M,\alpha}$ with the weights represented in a vector $\alpha = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$.

$$C_{M,\alpha} = \alpha_1 M + \alpha_2 M^T M + \alpha_3 M^T + \alpha_4 MM^T \quad \text{(Eq. 4.4)}$$

Table 4-1 Atomic propagations according to Guha *et al*. (2004)

Atomic Propagation	**Operator**	**Description**	**Figure**
Direct Propagation	M	A→B and B→x then A→x	Figure 4-4 (a)
Co-citation	$M^T M$	A→C and B→C and A→x then B→x	Figure 4-4 (b)
Transpose Trust	M^T	A→x then x→A	Figure 4-4 (c)
Trust Coupling	MM^T	A→B and x→B and D→A then D→x	Figure 4-4 (d)

In another motivating approach (Hang, Wang and Singh, 2009), trust is propagated based on three operators, namely aggregation, concatenation and selection, through a weighted directed graph while Heß (2007) propagates trust in a multi-layered architecture for a document recommendation system, from an author trust network to a document reference network. Moreover, the concept of *'trust decay'* for transitive propagation is applied in various algorithms (Gray *et al.*, 2003; Ziegler and Lausen, 2004; Golbeck, 2005; Chakraborty and Karform, 2012) where trust decays as the propagation level increases.

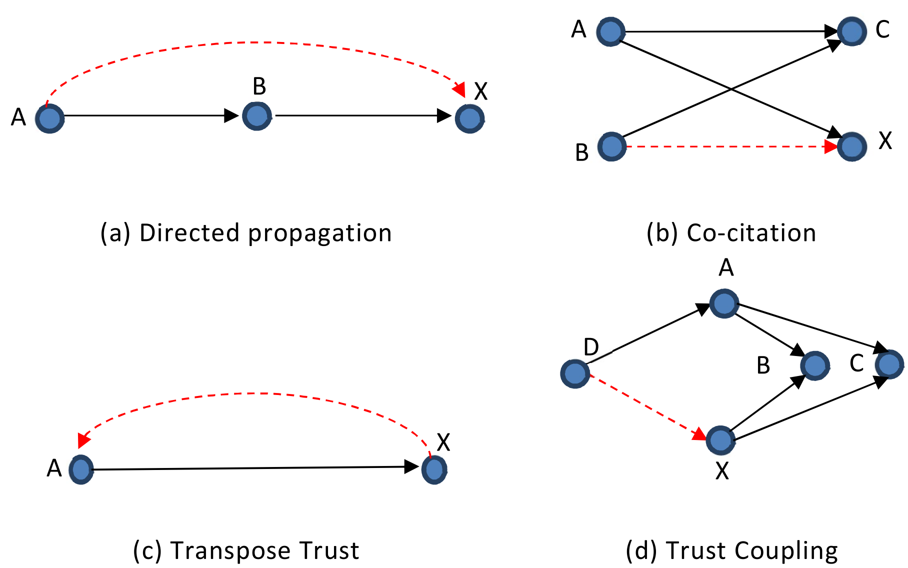


Figure 4-4 Atomic Propagation examples

This transitive propagation of trust is used to infer trust relationships in a trust network by inferring trust paths. Two typical approaches that use trust path inference are TidalTrust (Golbeck, 2005) and MoleTrust (Massa and Avesani, 2005), which will be discussed thoroughly in Sections 4.5.4 and 4.5.5, respectively. The aim of a trust path inference algorithm is to link two users not yet directly related by discovering a trust path that may pass through other intermediate linked users. The trust value for the inferred trust relationship is calculated by aggregating the trust values of the intermediate trust relationships, as described in the next section.

4.3.4 Aggregation

A trust inference algorithm, besides the propagation mechanism, must also include an aggregation strategy. It is very common in a large network to have multiple paths between two nodes and thus the inference of trust between these two nodes needs to aggregate the various propagation paths into one trust estimate. For example, suppose Alice connects to several trusted friends being connected to Frank and thus there are several paths through which Alice can be connected to Frank. Then all these several paths have to be combined and aggregated to only one trust estimation. In case there is contradictory information about a user, which is not the exception, the aggregation is much more difficult. Although the aggregation strategy is of the same significance with the propagation mechanism to infer trust, researchers have not given enough attention compared to that on propagation.

There are various mathematical aggregation operators like the arithmetic mean, the weighted mean, the median, the minimum, and others. It is also conceivable to combine trust from multiple paths with the 'maximum operator' where the trust estimate for the inferred relationship is the maximum value of all the available trust paths. Consider for example the trust network of Figure 4-5, where we want to estimate the trust score between users A and x. The three possible paths from A to X are A→B→C→X, A→D→E→F→X and A→D→G→H→F→X. To infer the trust relationship between the two users, the trust metric has to propagate trust scores through the three possible trust paths in the network, calculating a trust estimate for each path and then to aggregate all of them into one single value. Supposing that the trust metric calculates the propagated trust as the mean of trust values of the path and then using the 'maximum' as aggregation operator, the final trust estimation of user A to user x is:

$$max\left(\frac{1.0+0.6+0.9}{3},\frac{0.5+0.8+1.0+0.8}{4},\frac{0.5+0.3+0.2+0.4+0.5}{5}\right)=0.83$$

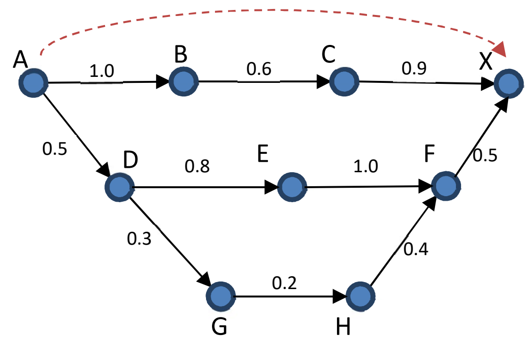


Figure 4-5 Aggregation example

However, an aggregation through paths with contradictory trust values could produce problematic results. In the trust network of Figure 4-5, for example, we can observe that A→B→C→X and A→D→G→H→F→X are two contradictory trust paths as the average trust value for the first path is 0.83 and for the second path is 0.38. In other words, A can highly trust X through the users B and C but also A cannot trust X through the users D, G, H and F. But since A does not trust enough D, may not also trust D's recommendation for other users and so the path with the low trust degrees can be omitted as it is not considered strong enough. In this way, Golbeck's (2005) TidalTrust selects the first path as the strongest one through which A can trust X.

One of the first methods for aggregating multiple opinions in social networks was that of Richardson *et al*. (2003) which calculates the belief of a social network user in a statement by finding the paths (leading to the statement) and incorporates a concatenation operator for calculating the belief of each path and an aggregation operator which combines the beliefs of all the paths. On the other hand, Golbeck's (2005) trust metric propagates through the shortest and strongest path which can be characterised as a selection rather than a combination (aggregation) of paths.

Recently, Victor *et al*. (2011) extensively studied the trust score aggregation problem and introduced some families of operators based on Ordered Weighted Averaging (OWA-based), and knowledge-enhanced operators (K-OWA, K-IOWA, the knowledge awarding averaging trust score operator KAAV and the knowledge-based trust score OWA). They moreover, proposed four trust score aggregation strategies:

- TMAX: maximum trust degree for the lowest possible knowledge level
- DMAX: maximum distrust degree,
- KAV: average of the most knowledgeable trust scores

- KMAX: maximum trust and distrust degree

Their experiments demonstrate that the families of the operators they proposed produce more accurate results in real-world large datasets which are typically noisy. Yet, research regarding the aggregation process in trust inference is still in its infancy (Victor, Cornelis and DeCock, 2011) needing to give greater attention to new aggregation operators combined with propagation but also to operators preserving privacy.

4.4 Trust metrics

While propagation and aggregation are the two key factors for building trust relationships, trust metrics define the way to measure a relationship in a trust network. The main feature of trust-based recommender systems is certainly the trust weight of the trust relationship between two users, otherwise known as local trust, or the total trustworthiness of a user within a network, also known as global trust. Trust weights are statements in trust networks where a user can explicitly state which user(s) they trust by rating them directly, or indirectly by rating his/her comments. However, trust relationships can be defined also implicitly, by calculating similarity measures between users, based on the explicit trust statements of the trust network or by calculating the similarity between users independently of any trust statement. Trust metrics are measures to infer trust relationships and build new trust networks or expand the existing ones; they utilise computational trust properties to compute and infer a new trust relationship.

O'Doherty, Jouili and Van Roy (2012) define *"a trust metric from user u to user v in a social network can be seen as the subjective probability that the trustor, u will have the same preferences and tastes as the trustee v."* In other words, a trust metric is a similarity metric between two users in a social network. As already mentioned, (Section 3.2) traditional recommender systems are based on the similarity between items, or/and users to make predictions. In Section 3.2.1, we saw various similarity metrics that measure the distance between two objects. In trust-based recommender systems, the trust weights between the users in the trust network represent the similarity between them. The trust weights can be computed, by the explicit trust statements given in the social network or can be inferred through trust metrics.

A lot of effort has been put into different disciplines for computing trust and several approaches have been proposed. A socio-cognitive model of trust has been built (Falcone, Pezzulo and Castelfranchi, 2003) by using Fuzzy Logic to compute the value of the trustfulness, starting from belief sources that refer to trust features. Hooijmaijers and Stumptner (2007) proposed an ontology integration tool using suggestions for dynamically changing trust of a document author, whereas Selvaraj and Anand (2012) used historical data in peer-to-peer networks and genetic algorithm. Golbeck *et al.* (2003) built a trust network in the Semantic Web by extending FOAF profiles to include trust relationships, while Appleseed (Ziegler and Lausen, 2004) is a local and group trust metric for ranking all the nodes in the network. These last two metrics will be discussed in detail later in Sections 4.5.4 and 4.5.3, respectively.

However, the first applications of trust and reputation metrics were used for security reasons, discovering trust paths through Public Key Infrastructure (PKI) (Maurer, 1996; Zhao and Smith, 2006). PKI is a tool for performing secure transactions by mapping a user with a public key. A 'chain of certificates' is constructed through certification paths, from the first certificate issuer to the target. The procedure is very similar to that of discovering trusted neighbours in a trust network. Another popular area in which trust metrics were initially used is peer-to-peer (P2P) networks for addressing the problem of data quality. In P2P networks, each peer connected in the network can share files with other peers. The problem is that every anonymous user can share any file of whatever quality with no limitations and no guarantee of the availability of the connection. For maintaining quality and security, several approaches (Aberer and Despotovic, 2001; Kamvar, Schlosser and Garcia-Molina, 2003; Kim and Song, 2011; Selvaraj and Anand, 2012) provide 'trustworthiness' for each peer in the network, depending on its reliability. The 'trustworthiness' is calculated through global trust metrics by measuring the reputation of the peer.

Generally, trust algorithms can be distinguished into two major categories, local or global, depending on the trust metric they use. Global trust, otherwise the 'reputation' of a user, is the universal trust value for that user within the network, while local trust defines trustworthiness and refers to the trust value that one user gives to another and it represents the user's subjective opinion of the other user. Generally, local trust metrics are preferred in opinion-based applications, while global trust metrics are preferred in systems where trust is not dependent on personal opinions, such as peer-to-peer networks and file sharing applications (Roy, Jouili and Skhiri, 2012). Moreover, local group trust metrics are suitable for computing neighbourhoods in decentralized systems (Ziegler, 2004) while Massa and Avesani (2005) showed that, compared to global metrics, the local metric produces more accurate results in case of controversial users also retaining good coverage.

4.4.1 Local trust metrics

Local trust metrics take into account personal and subjective opinions between users and compute different values of trust for each pair of users. It refers to the personalised trust relationship between two users and reflects the trustworthiness of a user A to the personalised and subjective view of a user B. Thus, trust calculation is personalised for each pair of users in the network and it does not take into account any shared opinions in the network.

Local trust metrics are proved (Massa and Avesani, 2005) to be very valuable in the case of controversial users. A controversial user is a user for whom the opinions of the other users are contradictory. Except for controversial users, local trust metrics provide better results in the case of grey-sheep users and users with unique tastes (for details refer to Section 3.4) as, in these cases, the global opinion about a user might also be controversial.

Local trust metrics exploit structural information of the trust graph and calculate the personalised trust value between the source and target user through a propagation mechanism. Consequently, trust can be inferred from local trust metrics only if there exists a connection path between the source and the target user and, therefore, it is unlikely to infer values for the whole network. Some well-known local trust metrics that will be analysed thoroughly next are TidalTrust (Golbeck, 2005), MoleTrust (Massa and Avesani, 2005) and Advogato (Levien, 2009).

4.4.2 Global trust metrics

Another concept with which trust is closely related is that of reputation. However, reputation is often confused with trust. The key distinction is that reputation of an agent is a factor affecting his trustworthiness. Trustworthiness is based on previous experience with the agent, while reputation is based on measures that this agent (node) has within a social network. Mui *et al.* (2002) state that reputation is the perception that an agent creates through past actions about its intentions and norms. As already mentioned, Wierzbicki (2010) defines reputation as the *"information about the trustee that is available to the trustor and is derived from the history of the trustee's behaviour in some contexts",* which can be represented as a probabilistic model. Moreover, Barber and Kim (2001) defined reputation as *"the amount of trust an agent gives an information source based on previous interactions among them."* When an information source delivers trustworthy information and satisfies the needs of other agents, then these agents may increase the reputation of the information source. On the contrary, no satisfaction of the other agents may decrease the reputation of the information source. Reputation is widely used in Peer-to-Peer (P2P) networks for preserving security within the systems.

Global trust, otherwise the 'reputation' of a user, is the universal trust value for that user within the network, while local trust refers to trust value that one user gives to another user and it represents the user's subjective opinion of the other. The main objective of global trust metrics is to rank all nodes in the network, rather than inferring new trust relationships, by computing and assigning one single trust value (reputation) for every node in the system. In other words, the global trust assigned to a user X is exactly the same trust value that a user A has for user X, as also the trust value that another user B has for user X. The simplest way would be to compute the overall user reputation by averaging the trust values received from every user. PageRank (Brin and Page, 1998) is one of the most popular global trust metrics for measuring the importance of a website. Similarly, Kamvar *et al.* (2003) proposed the EigenTrust algorithm which is a global trust metric for peer-to-peer networks. These two algorithms are very widely used in the literature as global metrics.

4.5 Major algorithms for trust inference

This section presents the algorithms of the most popular trust-based recommendation techniques. All these methods are based on the propagation property of trust to infer new trust relationships and are either global or local trust metrics.

4.5.1 Advogato

Advogato (Levien, 2009) is a group trust metric for discovering which users in an online community are trusted by other users and which are not. In other words, it calculates the global reputation of a node within the community by using a network flow model.

Specifically, it utilises a social graph representing community users as nodes and their relationships as directed edges. The relationship between two users in the community is a 4-level certificate assigned, from one user to another, as Observer, Apprentice, Journeyer, and Master. These certificates are calculated by the Advogato trust metric and determine the trust level of a user within a group of users.

The input for Advogato is the number n of members to trust, as well as the trust seed s, which is a subset of the entire set of users V. The output is a characteristic function that maps each member to a Boolean value, indicating trustworthiness

$$Trust_M : 2^V \times \mathbb{N}_0^+ \rightarrow (V \rightarrow \{trust, false\}) \qquad \text{(Eq. 4.5)}$$

The Advogato trust metric is computed in three conceptual steps.

1. Each node in the network is assigned a capacity based upon the shortest-path distance from the seed to the user x. The capacity of the seed is given by the input parameter n, whereas the capacity of each successive distance level is equal to the capacity of the previous level l divided by the average outdegree of trust edges $e \in E$ extending from l.
1. The graph is transformed into a graph with extra edges from each node to a special 'supersink' node.
2. A maximum network flow is computed for the transformed graph. Each node having flow across its corresponding edge to the supersink is accepted by the trust metric.

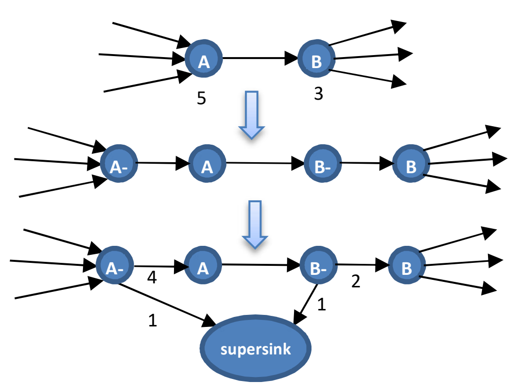


Figure 4-6 The three conceptual steps in Advogato trust graph conversion

Algorithm 1 Advogato trust graph conversion (Levien, 2009)

```
function transform (G = (A, E, C_A)) {
    E' =∅, A' =∅;
    for all x ∈ A do
        add node x⁺ to A';
        add node x⁻ to A';
        if C_A(x)   1 then
            add edge (x⁻, x⁺) to E';
            C_E' (x⁻, x⁺) =C_A(x)−1;
            for all (x, y) ∈ E do
                add edge (x⁺, x⁻) to E';
                C_E' (x⁺, x⁻) =∞;
            end do
            add edge (x⁻, supersink) to E';
            C_E' (x⁻, supersink) =1;
        end if
    end do
return G' = (A', E', C_E');
}
```

The algorithm shown below converts the existing graph into a new graph where capacities are assigned to edges. Each node x is split into two nodes, x^- and x^+. For a node x with capacity C_A, an edge is added from x^- to x^+ with capacity $C_A - 1$. For each edge from x^-, x^+ in the original graph, an infinite capacity edge from x^-, x^+ is added to the new graph. Finally, from each node x, a unit capacity edge is added from x^- to the supersink node.

Advogato trust metric is designed as 'attack resistant' by identifying the 'bad' nodes in the network excluding them from the network. Although the metric determines which users are trusted, the main drawback of the method is that it does not provide any ranking of trusted users so it is more applicable in P2P networks and not in a trust-based recommender system where the trust rank plays a key role.

4.5.2 EigenTrust

EigenTrust (Kamvar, Schlosser and Garcia-Molina, 2003) is another global trust metric designed for decreasing inauthentic files in peer-to-peer networks. In fact, it is a variation of the PageRank algorithm (Brin and Page, 1998) used by Google for ranking web pages.

Through the algorithm, a peer in the network is connected through a trust link with another peer based on its historical performance. The local trust values are aggregated and after normalisation they are propagated. All the local trust values in the network are represented in a matrix C and through a number of iterations, a new globally-accepted trust value t is computed for each peer. Specifically, this trust value is a measure of performance for the peer and thus it is fundamentally different from the trustworthiness of a user in a social trust-network which is mainly based on personal information and not on performance criteria.

Algorithm 2 Basic EigenTrust algorithm (Kamvar, Schlosser and Garcia-Molina, 2003)

$\vec{t^{o}} = \vec{p}$

repeat

$\qquad \vec{t}^{k+1} = C^T \vec{t}^{(k)};$

$\qquad \vec{t}^{k+1} = (1-a)\vec{t}^{(k)} + a\vec{p};$

$\qquad d = \left\| t^{k+1} - t^{(k)} \right\|;$

until $\delta < \varepsilon$;

4.5.3 Appleseed

Like Advogato, Appleseed (Ziegler and Lausen, 2004) is a group trust metric propagating energy from a node to other nodes based on the link weight. However, instead of using maximum network flow, the basic intuition of Appleseed is motivated by spreading activation theory found in neuroscience and cognitive psychology and it refers to the *"retrieval from long-term memory in which activation subdivides among paths emanating from an activated mental representation"* (Nolen-Hoeksema, Fredrickson and Loftus, 2009).

Similar to Advogato, the output is an assignment function of trust with domain V. However, Appleseed, given a network and a source, returns a ranking of all the nodes in the network (allows rankings of agents with respect to trust degree). In contrast to

Advogato, Appleseed uses transitive trust propagation. Source node s is activated through an injection of energy in^0. The energy is then fully propagated to other nodes along edges. The higher the strength of the edge (link weight) between two nodes, the higher the energy is propagated through these nodes. Yet, in order to avoid endless energy flow, a threshold T specifies the minimum energy for a node to receive for not becoming 'dry'. The algorithm also introduces the 'trust decay' by incorporating a spreading factor d.

Authors also discuss about distrust and backward propagation still one of the main advantages of their method is that nodes are accessed only when reached by energy flow and thus reduce the computational complexity.

Algorithm 3 Appleseed trust metric (Ziegler and Lausen, 2004)

```
function Trust_A  {(s ∈ V, in^0 ∈ ℝ_0^+, d ∈ [0,1], T_c ∈ ℝ^+)
    in_0(s)=in_0;
    trust_0(s)=0;
    i=0;
    V_0={s};
    repeat
        i = i+1;
        V_i = V_i-1;
        ∀ x∈ V_i-1: in_i(x)=0;
        for all x ∈ V_i-1 do
            trust_i(x) = trust_i-1(x)+(1-d) · in_i-1(d);
            for all (x,u) ∈ E do
                if u∉V_i then
                    V_i= V_i ∪ {u};
                    trust_i(u)= 0;
                    in_i(u)=0;
                    add edge(u,s);
                    W(u,s)=1;
                end if
                w =W(x, u) / ∑_(x,u')∈E W(x, u');
                in_i(u)=in_i(u)+d · in_i−1(x) · w;
            end do
        end do
        m = max_y∈Vi {trust_i (y)−trust_i−1(y)};
    until m ≤ T_c
    return trust: {(x, trust_i (x)) | x ∈ V_i};
}
```

4.5.4 TidalTrust

TidalTrust (Golbeck, 2005) is a modified breadth-first search algorithm infering trust through propagation and exploited in an experimental platform for recommending movies called FilmTrust (Golbeck, 2006b). The algorithm was also exploited in TrustMail (Golbeck and Hendler, 2004) which is not just a spam filter but a layer ranking email priority depending on the trustworthiness (trust value given manually by the recipient) of the sender. Golbeck *et al*. (2003) also introduce an ontology extending FOAF vocabulary for modelling trust relationship between users. FOAF stands for friend-of-a-friend, aiming to enrich with machine-readable content (through ontology), user profiles containing personal information, preferences, activities and relations to other people and objects.

Specifically, the TidalTrust algorithm uses the explicit trust values provided by the users of the network, for estimating unknown trust values between users, not yet being connected. Trust is inferred by the TidalTrust algorithm taking into account not only the shortest but also the strongest paths from the source user a to the sink user u. The unknown trust value between the source user a and the sink u is calculated as the **weighted average** of the trust scores attributed to u by the neighbours of a (users trusted by a) and is given by the following formula:

$$t_{a,u} = \frac{\sum_{u \in TN_a +} t_{a,b} t_{b,u}}{\sum_{u \in TN_a +} t_{a,b}} \qquad \text{(Eq. 4.6)}$$

The pseudocode presented in Algorithm 4 below, describes the steps followed in the algorithm. The source node begins a search for the sink. It polls each of its neighbours to obtain their rating of the sink. If the neighbour has a direct rating of the sink, that value is returned. If the neighbour does not have a direct rating for the sink, it queries all of its neighbours for their ratings, computes the weighted average, and returns the result. This process is repeated for each neighbour. Any available path at the minimum depth is recorded, and for each one of these shortest paths, is computed the aggregated trust values between the source's neighbours and the sink users weighted by the direct trust values between the source and its neighbours. Essentially, the nodes perform a breadth-first search from the source to the sink, and then inferred values are passed back to the source.

To limit the size of the search TidalTrust adopts the 'shortest path' practice since Golbeck (2006a) has demonstrated that it leads to a lower error. However, limiting the depth can lead to fewer nodes that can be reachable. The same study, confirms that using nodes with higher trust ratings ('strongest paths') leads to lower error. Strongest paths are incorporated into the algorithm by establishing a minimum trust threshold above or at which connections are taken into account. This threshold is not a fixed value but is defined as a variable, max, representing the largest trust value, that can be used as a minimum threshold, such that a path can be found from source to sink.

As a final task to predict the 'Recommended Rating' $p_{a,i}$ for an active user a and movie i the formula (Eq. 3.14) is modified to incorporate not only the explicit but also the inferred trust values of the users who have rated the film (the raters). More specifically, the **trust-based weighted mean** given by the formula (Eq. 4.7) is the baseline for including trust values in the process of computing the predicted rating. It is the weighted mean (Eq. 3.14) with trust weights for expressing the similarity weights i.e. $w_{a,u}=t_{a,u}$. Golbeck in two studies (Golbeck, 2005, 2006b) used this algorithm to compute the unknown rating $p_{a,i}$ for target item i and active user a

$$p_{a,i} = \frac{\sum_{u \in R^T} t_{a,u} r_{u,i}}{\sum_{u \in R^T} t_{a,u}} \qquad \text{(Eq. 4.7)}$$

where R^T represents the neighbours of user a (the set of users that rated i) and for whom the trust value $t_{a,u}$ is greater than or equal to a threshold $\theta > 0$.

To give an example of how the TidalTrust algorithm works, we can consider the trust network illustrated in Figure 4-7 where nodes represent users and arrows represent the

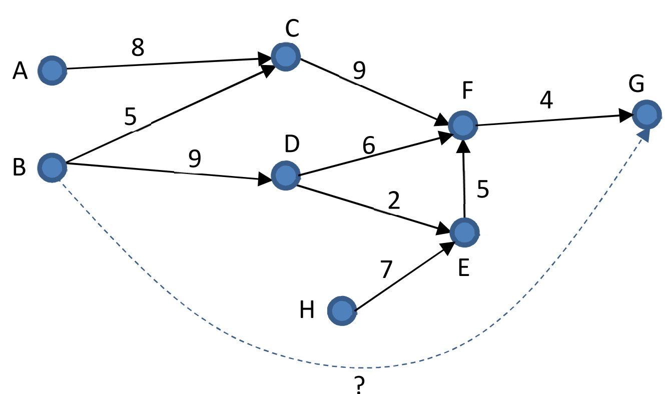


Figure 4-7 An example of direct trust network

trust relationships between users. Each trust relationship is directed and weighted with the values depicting the trust weight of the relationship on a scale between 0 for full distrust and 10 for full trust.

Given the above trust network (Figure 4-7) suppose we want to calculate, through the TidalTrust algorithm, the inferred trust value between the source user B and the sink G. Three possible paths are available from B to G: i) the B→C→F→G, ii) the B→D→F→G and the iii) B→D→E→F→G. As we can observe also, in the above network, users (nodes) A, B, H are in depth level 1, whereas C and D are in depth level 2 for users A and B. Notice that user E is in depth level 2 for user H. However, for users C and D, node H is in depth level 3.

The algorithm initially starts to find the neighbours of B for next level, thus depth level 2, These neighbours are C and D. Similarly, the neighbours for depth level 3 are F and E. Finally, the user G can be reached directly from user F or through one more level from user E. But the 'shortest path' factor indicates that the best way is to reach the user G directly from user F and not through the user E. Now the factor of 'strongest path' indicates that the path D→E will not be accessed as the minimum threshold for trust weight is set to 3 ($max=3$) and so the best path is D→F→G. Thus, the inferred trust value $t_{B,G}$ can be calculated applying the (Eq. 4.6) as:

$$t_{B,G} = \frac{\sum_{u \in TN_B+} t_{B,u} t_{u,G}}{\sum_{u \in TN_B+} t_{B,u}} = \frac{t_{B,C} t_{C,G} + t_{B,D} t_{D,G}}{t_{B,C} + t_{D,G}} = \frac{5 \cdot t_{C,G} + 9 \cdot t_{D,G}}{5+9}$$

$$t_{C,G} = \frac{t_{C,F} t_{F,G}}{t_{C,F}} = \frac{9 \cdot 4}{9} = 4 \ , \ t_{D,G} = \frac{t_{D,F} t_{F,G}}{t_{D,F}} = \frac{6 \cdot 4}{6} = 4$$

$$t_{B,G} = \frac{5 \cdot 4 + 9 \cdot 4}{5+9} = 4$$

So the inferred trust value is $t_{B,G}=4$. From the above example, we can see that the trust values $t_{C,G}$ and $t_{D,G}$ were also inferred between users C, G and D, G respectively. In the same way, more trust values can be inferred from the existing trust paths between other users and thus expand the trust network.

The next step for the TidalTrust recommender system is to predict 'recommended ratings'. So, considering the above trust network and the ratings Table 4-2, we would like to predict what rating would give user B for Movie2 applying the (Eq. 4.7)

Table 4-2 A sample user/item ratings table

User/item	Movie1	Movie2	Movie3
A	9	6	
B	5	?	9
C	7		2
D	5	8	7
E	8		
F			8
G		5	
H			6

The neighbourhood of user B consisting of the users which rated Movie2 and are also connected with user B, either directly through explicit trust ratings or implicitly through inferred trust values. In Table 4-2 we can see that users A, D, G have rated Movie2 while B is connected with just the two of them. Specifically, B is explicitly connected with D (Figure 4-7) and implicitly with G, as inferred from the previous steps of the algorithm. Consequently, the predicted rating of user V to Movie2 is calculated as follows:

$$p_{B,Movie2} = \frac{\sum_{u \in R^T} t_{B,u} r_{u,Movie2}}{\sum_{u \in R^T} t_{B,u}} = \frac{t_{B,D} r_{D,Movie2} + t_{B,G} r_{G,Movie2}}{t_{B,C} + t_{B,G}} = \frac{9 \cdot 8 + 4 \cdot 5}{9+4} = \frac{92}{13} \simeq 7$$

This way the system can predict values for recommending movies to other users. Although the specific table and trust network are a small sample, we can nevertheless notice that there is a user for which no recommendations can be produced. More precisely, user G does not trust any other user, and so, no predictions can be made applying the (Eq. 4.7). This problem reflects the real-world datasets which are usually sparse and of course, it depicts the cold-start problem for a user, not on ratings but on building trust network.

Golbeck (2006b) claims that the accuracy of the recommended ratings produced by TidalTrust outperforms the predictions produced by both simple average and common collaborative filtering algorithms. However, the most interesting part is that TidalTrust produces significantly more accurate recommendations for grey-sheep users, and more specifically, for users who have different opinion for a specific item compared to the average opinion.

Algorithm 4 TidalTrust (Golbeck, 2005; Victor, Cornelis and DeCock, 2011)

```
for each user u do
PathStrength[u] = −1, PathStrength[a]=1;
maxDepth =∞, depth=1, add a to queue
while queue not empty and depth ≤ maxDepth do
    x=queue.dequeue, push x on stack
    if x and u are not adjacent then
        for each user i adjacent to x do
            add i to next_level_queue if i is not yet visited
            if next_level_queue contains i then
                strength = min(PathStrength[x], t_{x,i})
                PathStrength[i] = max(PathStrength[i], strength)
            end if
        end for
    else
        maxDepth = depth, strength = PathStrength[x]
        PathStrength[u] = max(PathStrength[u], strength)
    end if
    if queue is empty then
        queue = next_level_queue, next_level_queue=new_queue, depth++
    end if
end while

for each user u do trust to sink[u] = −1
while maxDepth!=MAX and stack is not empty do
    v=stack.pop
    if maxDepth =∞ then t_{a,u} = 0
    if v is adjacent to u then
        trust to sink[v] = t_{v,u}
    else
        numerator=denominator=0
        for each user i adjacent to v do
            if t_{v,i} ≥ PathStrength[u] and trust to sink[i]! = −1 then
                numerator+ = t_{v,i} ∗ trust to sink[i], denumerator+ = t_{v,i}
            end if
        end for
        if denumerator > 0 then
            trust to sink[v] = numerator/denumerator
        end if
    end if
end while
ta,u = trust to sink[a]
if trust to sink[a] = −1 then t_{a,u} = 0
```

4.5.5 MoleTrust

Similar to TidalTrust, researchers (Avesani and Massa, 2005; Massa and Avesani, 2005) proposed the MoleTrust algorithm which also incorporates explicit trust statements to propagate and infer trust. However, MoleTrust, instead of applying the breadth-first search method of TidalTrust it uses depth-first search method to infer trust. Another difference between the two algorithms also is that while TidalTrust is based on the weighted mean formula, on the contrary, MoleTrust is based on the classic collaborative filtering as shown in (Eq. 4.8).

$$p_{a,i} = \overline{r_a} + \frac{\sum_{u \in R^T} t_{a,u}(r_{u,i} - \overline{r_u})}{\sum_{u \in R^T} t_{a,u}} \qquad \text{(Eq. 4.8)}$$

In fact, this technique replaces the PCC similarity, in the Resnick's formula (Eq. 3.15) of collaborative filtering, with the trust value between the two users. In that way. The MoleTrust algorithm incorporates the weighted average of (Eq. 4.6) trust metric in the recommendation process producing trust-based collaborative filtering recommendations. The formula for estimating the trust value between the two users is very similar to TidalTrust but MoleTrust uses the $horizon$ parameter as a threshold of the distance that trust can be propagated. The horizon parameter indicates the maximum-depth that trust will be propagated but here, the parameter is a fixed value, instead of the shortest and strongest path incorporated by TidalTrust. The $horizon$ parameter can lead to less satisfactory results, as low values may decrease the possibilities for a user to find neighbours, on the contrary, high values can increase the possibility of incorporating noisy or undesirable information from users that should not be influential such as users in long distance. However, in contrast to TidalTrust, MoleTrust performs additionally a backward exploration for finding trust paths.

Algorithm 5 presents the pseudocode of MoleTrust for target user a, horizon d and trust threshold α; The algorithm can be conceptually divided into phases. The first step is to remove cycles in the graph, by ordering users based on distance from source user and keeping only trust edges going from users at distance n to users at distance $n+1$. The result is a modified directed acyclic graph. Thus, the algorithm becomes more time-efficient as each user has to be visited no more than once to compute the trust value of the inferred relationship. The second step is a simple graph walk over the modified trust network, starting from source user a. Initially, the trust scores of users at distance (depth) 1 are computed, thus, the explicit trust scores of the neighbours of user a are obtained. Then the trust scores of users in depth 2 are calculated according to the weighted-average formula of (Eq. 4.6). The predicted trust score of a user is the average of all the accepted incoming trust edge values, weighted by the trust score of the user who has issued the trust statement. Not all the trust edges are accepted since for each distance, MoleTrust discards any edge to users with a trust score less than the trust threshold α since these users are considered as not trustworthy. Trust scores are predicted for users in each depth until the maximum depth d.

The experimental results show that the improvement of accuracy comparing with collaborative filtering algorithms remains similar, however, the coverage is improved by 20% when trust is propagated and moreover coverage is significantly improved especially for cold-start users. Massa and Avesani (2005) also proved that their approach outperforms global trust metrics in prediction accuracy for controversial users (users being trusted by one group and distrusted by another) while retaining a good coverage.

Algorithm 5 MoleTrust for target user a, horizon d and trust threshold α (Avesani and Massa, 2005; Victor, Cornelis and DeCock, 2011)

```
Step 1:
    dist=0
    users[dist] = a
    add a to modified_trust_network
    while dist < d do
        dist++
        users[dist]=users adjacent to users[dist−1] and not yet visited
        for each user b from users[dist] do
            add b to modified_trust_network
            add all edges from users[dist−1] to b to modified_trust_network
        end for
    end while
Step 2:
    dist = 1
    t_{a,a} = 1
    ∀u∈ users[1]: t_{a,u} = trust statement issued by a
    while dist < d do
        dist++
        for each user u in users[dist] do
            predecessors=users v for whom t_{v,u} ≥ α in modified_trust_network
            t_{a,u} = ∑_{v∈predecessors} t_{a,v} · t_{v,u} /∑_{v∈predecessors} t_{a,v}
        end for
    end while
```

4.6 Conclusion

This chapter provided an extensive presentation of the trust-based recommender systems, including definitions and the computational properties of trust, while special focus was given to trust metrics. Current research, focuses on the computer science aspect of trust, and particularly on how it is used and computed, based on the definition of trust as a *"relationship between two agents namely the trustor and the trustee where the trustor trusts the trustee in a specific context."*

Several studies (Golbeck, 2005; Massa and Avesani, 2005; Wan and Chen, 2011; O'Doherty, Jouili and Van Roy, 2012; Victor *et al.*, 2013; Ziegler and Golbeck, 2015) demonstrate that the use of trust can significantly improve both the coverage and the accuracy of recommendations, especially with sparse datasets, which is the usual case. Moreover, trust can significantly improve recommendations accuracy when item ratings are more extreme and show disagreement between users. Section 4.2 examined all the major methods for inferring trust. A summary table of all these methods can be found below (Table 4-3). The analysis of these methods to infer trust, revealed their limitations. More specifically, although the accuracy of the recommended ratings with TidalTrust and MoleTrust has been proved to outperform some baseline recommender system algorithms, it is, however, strongly affected by the density of the trust network. Especially for users with no trusted neighbours or even with not at least moderately trusted neighbours, it is impossible to find any trust path with the specific algorithms. The investigation of the trust inference mechanism and the survey of the existing trust metrics is the necessary preparatory step for getting all the in-depth knowledge to fulfil the aim of this study.

Table 4-3 Major algorithms for trust inference

Algorithm	Approach	Propagation	Trust metric
Advogato (Levien, 2009)	Maximum network flow	Shortest path and capacity	Group
EigenTrust (Kamvar, Schlosser and Garcia-Molina, 2003)	Similar to PageRank		Global
Appleseed (Ziegler and Lausen, 2004) i	Spreading activation	Energy propagation	Group
TidalTrust (Golbeck, 2005)	Breadth-first search	Shortest + strongest path	Local
MoleTrust (Massa and Avesani, 2007)	Depth-first search Collaborative+trust	Shortest with horizon+ strongest with threshold	Local

Chapter 5

The need for a new system

5.1 Limitations of current systems

Several recommendation approaches have been introduced in the commercial and research area. One of the constant pursuits of researchers is to find out how to recommend the 'best choice for the user'. The reason for the constant research effort in this area is that, except for the constant need for improving the performance of the system, there is a need also to overcome certain limitations that face the existing systems:

Data sparsity. This is a problem encountered in all the aforementioned approaches and refers to the user ratings matrix, which is typically sparse, as most users do not usually rate most items. This leads to the formation of small neighbourhoods and poor recommendations. Methods dealing with this problem are the Default Voting (Breese, Heckerman and Kadie, 1998), the Singular Value Decomposition (Sarwar *et al.*, 2000; Rennie and Srebro, 2005; Salakhutdinov and Mnih, 2007, 2008; Koren, Bell and Volinsky, 2009; Yu *et al.*, 2009) and other methods (Wang, Vries and Reinders, 2006). Moreover, as previously stated (Section 3.5.2), trust-based recommender systems suffer from insufficient number of trust statements, which leads to sparsity of the trust matrix.

Cold-start problem. This is related to data sparsity as it is encountered when a new user or a user with low activity does not provide enough knowledge about his/her preferences to the system. Similarly, a new item in collaborative filtering needs a kick start rating to be included in an algorithm. A possible solution could be to suggest the user to rate a minimum number of items so as to receive more accurate recommendations. But this means human effort and users are often not willing to spend time on a new system that has not yet offered them any service. However, techniques like hybrid and knowledge-based (Ricci *et al.*, 2011) recommender systems deal better with the cold-start problem.

Lack of transparency. Another basic drawback of recommender systems is the lack or absence of explanations. If there is not enough *explanation* of the way recommendations emerged, the user does not trust the source enough. Ray and Mahanti (2010) argue that the possibility of people accepting recommendations made by trusted friends is higher than that of accepting recommendations made by strangers. Hence, trust-based recommender systems provide a solution to this problem by providing recommendation from trusted neighbours, especially when the trust relationship is stated explicitly.

Grey-sheep. Most of the recommendation approaches use similarity measures between users to provide their recommendations; however, there exist users being not similar to others and their preferences are not consistently similar or dissimilar with any group of users. As a result, it is difficult to fit these users in any neighbourhood, group or cluster. Dealing with this user is a great challenge for researchers who have proposed a number of approaches (Cantador, Bellogín and Castells, 2008; Lucas *et al.*, 2013; Ghazanfar and Prügell-Bennett, 2014). Furthermore, from the statistical point of view, as the number of users increases in the system, the chance of finding other similar users, increases respectively and so recommendations can be more accurate in large-scale systems. Trust-based recommender systems, and especially those calculating trust, based on local trust metrics provide significantly more accurate recommendations for these users, as Golbeck (2006b) showed.

Synonymy and polysemy. Another challenge for recommender systems is when items, although similar, may have different names, that is, synonymy. Respectively, polysemy is the multiple meaning for one word, which means that items described by the same word does not imply that they are always similar. As a result, similar items could be ignored due to synonymy, while wrong items could be considered as relevant while they are not. Semantics are necessary to be implanted to deal with this problem. Two methods which can deal with synonymy and polysemy problems are the Latent

Semantic Indexing (Deerwester *et al.*, 1990) and semantic analysis (Ricci *et al.*, 2011) with lexicons and ontologies for interpreting the meaning of words.

Change of preferences. Most collaborative filtering approaches assume that user preferences are stable and consistent over time, although, in real life, preferences can change over time, even during the day. Consider a user being interested in buying a mobile phone, but simultaneously also searching to buy a present for a friend. Thus, user profile has to be constantly adapted so as for the system to maintain a satisfactory level of performance (Nanas, De Roeck and Vavalis, 2009).

Security and Privacy. One of the drawbacks of recommender systems is that they are prone to malicious attacks, fraud and shilling attacks. Especially, common collaborative-filtered recommender systems are prone to malicious attacks by copying, for example, a user's rating profile for gaining similarity. A typical example of this is fake profiles, and the so-called 'shilling attack' where a user can intentionally highly rate specific items and low rate other competitive products, for reasons such as 'fun or profit' (Lam and Riedl, 2004). These pseudo-ratings to particular products can significantly bias the recommendations and affect the trustworthiness of the system. Several studies deal with shilling attacks and propose various solutions (Lam and Riedl, 2004; O'Mahony and Hurley, 2004; Chirita, Nejdl and Zamfir, 2005; Mobasher *et al.*, 2005; Zhang, 2009). Security has to deal also with privacy and ownership of data. In order to build user profiles, a recommender system requires the users to register personal information and, sometimes, their preferences. Of course, most of the recommender systems maintain also purchase and search historical data. Moreover, social-based recommenders maintain information about users regarding their relationships, friendships and opinions about other users. All this sensitive information has to be kept secured and hidden to malicious attacks. In case of abuse by the provider or leak by a hacker, users will lose their confidence in the system and this means that they will leave or at least they will no longer provide their opinions or other information to the system. Users expect the systems to protect their privacy and prevent any leak of their personal information and habits. In the decision-making process, one of the critical components for considering and accepting a recommendation or not is the trustworthiness of the source. As it is crucial for a system to be trustworthy, privacy preservation gave rise to research(Canny, 2002; Polat and Du, 2003; Zhan *et al.*, 2010; Jeckmans *et al.*, 2013) to deal with this issue, but there remains yet a lot to be done in this specific area.

However, it is common in real life to seek advice for topics we are not expert in or have no experience of. Friends can be a valuable information source, but the Internet can also play an important role in information seeking. Nowadays, with the explosive growth of Web 2.0 technologies, the two sources, friends, and the Internet, are combined in services called Social Networks. Recommender systems taking advantage of these technologies are the so-called social recommender systems, exploiting not only opinions and ratings about items, but also the relationships between users. The emergence of Web 2.0 and the ability of the user to express himself/herself and be heard all over the world changed not only the way of communication, but also business and marketing. The expression of personal opinion is *"more likely to be believed by today's sceptical consumer than advertisements or professional input"* (Smith, Menon and Sivakumar, 2005). Social networks are characterised by relations between 'friends' or 'followers' with similar interests. As O'Connor (2008) points out, *"consumers are no longer dependent on website owners to publish the information they seek, as they can increasingly rely on unfiltered, dynamic and topical information provided by their own peers."* For example, in a usual shopping day, just as a friend's opinion counts more than the seller's, in the same way, word-of-mouth *"is perceived as being more vivid, easier to use and more trustworthy than marketer-provided information."*

Hence, research in recommender systems has turned to the exploitation of trust relationships to improve predictions, as discussed in the previous chapter. Golbeck (2006b) proposed Filmtrust, which is a movie recommender system based on FOAF vocabulary for creating a social network of trust with trust values given by the users. Massa and Avesani (2004) computed similarity between users using trust metrics based on trust networks. In another study (Bedi, Kaur and Marwaha, 2007), trust-based recommendations are enhanced with ontologies for creating annotated content for the Semantic Web. In a recent study (Mehta and Banati, 2012), shuffled frog leaping algorithm was applied for clustering the users' different social contexts. Generally, trust-based systems are proved (Ray and Mahanti, 2010) to overcome known limitations, such as frauds and attacks and lack of transparency, while they demonstrate better accuracy than traditional recommender systems in grey-sheep users and data sparsity by increasing coverage.

5.2 Classification of trust models

Various approaches exist in the literature for modelling trust, but, here, we concentrate on trust models for recommender systems and classify them based on the technique they use. Another classification of trust-based recommender system is that of Victor, Cornelis and DeCock (2011) who adopt a classification of probabilistic versus gradual approaches while examining which of the trust/distrust concepts are represented. In the former, trust is computed as the probability of trusting someone or not, while, in the latter, trust is gradually expressed, as in everyday life, using a scale to represent the degree of trust in another user. But this is a very general classification and does not examine in depth which technique it is based on.

Here trust models are classified into five major categories based on the techniques they use: (i) statistical technique, (ii) heuristics-based, (iii) graph-based, (iv) semantic-based and (v) fuzzy logic. In some of them, we distinguish their subcategories.

As shown in Table 5-1, statistical techniques are very popular, within which probabilistic techniques have the majority of variants. Probabilistic techniques are also very popular in predicting user ratings in traditional collaborative filtering recommender systems. However, graph-based trust models introduce a new philosophy and treat the concept of trust as a relationship between users within a social network. The technique is consistent with everyday social interactions and exploits the properties of the existing web-based social networks through Social Network Analysis, for discovering similarities between users.

5.3 Comparison of Graph-Based Recommendation Models

Essentially, a trust network is a social network which can be represented as a directed graph in which nodes are the users and edges are the trust relationships. When trust weights follow a gradual scale, then this graph is a labelled as a directed graph with the degree of trust as labels. Therefore, graph theory and propagation property of trust can be exploited to infer not only existing trust relationships, but also to cluster users and create groups of users with common preferences for improving the performance of recommender systems; however, trust models exploiting graph theory are in infancy. Hence, it is meaningful to study recommender systems based on graph theory and identify possible gaps and improvements. Consequently, these models were examined for the way that trust is computed and propagated, the network perspective of the trust metrics and also the trust establishment.

Table 5-1 Classification of trust models

<table>
<tr><th>Statistical techniques</th><th>Heuristics-based solutions</th></tr>
<tr><td rowspan="7">• Probabilistic techniques
○ Bayesian systems
(Mui, Mohtashemi and Halberstadt, 2002)
(Jøsang and Lo Presti, 2004)
○ Belief models
(Falcone, Pezzulo and Castelfranchi, 2003)
(Barber and Kim, 2001)
(Guha et al., 2004)
■ Dempster-Shafer theory
(Yu and Singh, 2002)
■ Subjective logic
(Jøsang, 2001)
(Jøsang, Hayward and Pope, 2006)
○ Markov Models
(Fouss et al., 2007)
(Dong and Frossard, 2012)
(ElSalamouny, Sassone and Nielsen, 2010)
(Song, Phoha and Xu, 2004)
• Machine learning
○ Artificial Neural Networks
(Bedi and Kaur, 2006)
○ Bayesian classifiers
(Hooijmaijers and Stumptner, 2007)
(Guanfeng et al., 2010)
(Patel et al., 2005)
(Guanfeng, Yan and Orgun, 2009)
○ Decision trees
(Zolfaghar and Aghaie, 2012)</td><td>• Genetic algorithms
(Selvaraj and Anand, 2012)
• Ant colony
(Bedi and Sharma, 2012)</td></tr>
<tr><td>Graph-based</td></tr>
<tr><td>(Golbeck, 2005)
(Golbeck, 2006b)
(Golbeck 2006)
(Zhang, Bai and Gao, 2009)
(Walter, Battiston and Schweitzer, 2008)
(Li et al., 2012)
(Yang, Steck and Liu, 2012)
(Victor, Cornelis and DeCock, 2011)
(Ziegler and Lausen, 2004)
(Golbeck and Hendler, 2006)</td></tr>
<tr><td>Semantic-based</td></tr>
<tr><td>(Golbeck, 2006a)
(Heath, Motta and Petre, 2007)
(Kim and Kwon, 2007)
(Oufaida and Nouali, 2009)
(Gao, 2010)
(Martín-Vicente, Gil-solla and Ramos-Cabrer, 2012)
(Golbeck, Parsia and Hendler, 2003)
(Bedi, Kaur and Marwaha, 2007)</td></tr>
<tr><td>Fuzzy logic</td></tr>
<tr><td>(Aberer et al., 2006)
(Bedi and Kaur, 2006)
(Capuruço and Capretz, 2012)
(Chen et al., 2005)
(Li and Kao, 2009)</td></tr>
</table>

5.3.1 Comparison criteria

Accordingly, the literature about graph-based recommender systems can be categorised based on three major categories regarding (i) propagation, (ii) network perspective and (iii) trust establishment.

Propagation

As mentioned before, propagation is a property of trust that benefits the process of predicting the trust score through known trust paths. Direct trust relations in a user's trust network build a path through which new indirect connections can be established with other users, not known. There are various strategies for computing trust propagation. A very common one is the random walk approach, which assigns a transition probability to each edge by walking from one node to another. It is a method used also in PageRank (Brin and Page, 1998) and EigenTrust (Kamvar, Schlosser and Garcia-Molina, 2003) for computing the global trust of a node. Appleseed (Ziegler and Lausen, 2004) is another method that propagates by spreading 'trust energy' based on the strength of the edge. A graph-based popular method is that of Golbeck's TidalTrust (Golbeck, 2005) wherein propagation is based not only on shortest path, but also on strongest path. An extension of TidalTrust is the MoleTrust (Massa and Avesani, 2007), which uses a fixed parameter, horizon, as maximum path length and a trust threshold for participating in the process.

Network perspective: Global versus Local trust

As already mentioned, trust can be inferred through global or local trust measures. Local trust is the subjective measure of a user for the trustworthiness of another user. In other words, it is the degree of a trust relationship between two users being stated explicitly. Global trust, on the other hand, is the average opinion of the whole community about the trustworthiness of a user. Namely, it is the reputation that a user has in the network. In trust- based recommender systems literature, local trust metric is generally preferred (Bedi and Kaur, 2006; Bedi, Kaur and Marwaha, 2007; Heß, 2007; Massa and Avesani, 2009; Zarghami *et al.*, 2009; Victor, Cornelis and DeCock, 2011; Charif, Anne and Azim, 2012; Mehta and Banati, 2012), although there are systems adopting both local and global trust (Andersen *et al.*, 2008; Hang and Singh, 2010; Yang, Steck and Liu, 2012).

Trust establishment

Trust establishment can be based on explicit or implicit trust networks. Explicit networks are built with explicit trust statements, whereas implicit are inferred from user behaviours. Implicit trust relationships can be computed through user similarity and other trust metrics.

Explicit and implicit trust can be either bivalent or expressed on a gradual scale. Several studies (Golbeck, 2005; Heß, 2007; Andersen *et al.*, 2008; Massa and Avesani, 2009; Victor, Cornelis and DeCock, 2011; Charif, Anne and Azim, 2012) use explicit trust; however, several other (Bedi and Kaur, 2006; Bedi, Kaur and Marwaha, 2007; Zarghami *et al.*, 2009; Bedi and Sharma, 2012; Mehta and Banati, 2012; Yang, Steck and Liu, 2012) infer trust relationships to build the implicit trust.

Table 5-2 Comparison of main literature in graph-based trust recommender systems.

	Algorithm approach	Propagation	Network perspective	Trust establishment
(Golbeck, 2005)	TidalTrust (breadth-first search)	Direct propagation Shortest path + strongest path	Local	explicit
(Massa and Avesani, 2007)	MoleTrust Collaborative+trust	Direct propagation Shortest path with horizon+ strongest path with threshold	Local	explicit
(Victor, Cornelis and DeCock, 2011)	EnsebleTrust trust as a weight PCC+trust	Direct propagation	Local	explicit
(Heß, 2007)	Multilayer network trust network+ document reference network	Propagation through layers	Local	explicit
(Hang and Singh, 2010)	Graph similarity	Graph propagation	Local + global	explicit

Table 5-2 illustrates briefly the main literature which uses graph models in their trust-based recommender systems based on the categorisation criteria defined previously in this section. It becomes immediately apparent that the focus of these models is in computing trust values through propagation by exploiting various techniques in the trust network based on graph theory. In all these models, trust networks are explicit, whereas, in the majority of these, trust is computed through local metrics. The reason for this is that, as already discussed (Section 4.4.1), local trust metrics provide more accurate recommendations (Massa and Avesani 2005) especially for cold-start users or when dealing with controversial items and users also retaining good coverage. Except for controversial users, local trust metrics provide better results in the case of grey-sheep users and users with unique tastes as, in these cases, the global opinion about a user might be also controversial. Moreover, they are more resistant against attacks due to relationships explicitly stated, although in global trust, fake user profiles can impact the reputation of a user.

Another issue observed in Table 5-2, is that, although propagation is extensively addressed in all the examined models, in all studies that trust is inferred through local metrics it is based on the direct propagation of the atomic propagations (Section 4.3.3). However, as will be analysed later (Section 6.1), the homophily phenomenon on which the propagative property of trust is based, indicates that not only *"the friend of my friend is also my friend"* but also that *"if two persons, not yet connected, have a common friend, then there is an increased possibility for these two persons to become friends,"* This kind of relation corresponds to the co-citation of the atomic propagations, which is not considered in any of the examined approaches. While Guha *et al.* (2004) proposed a trust metric combining a 'basis set' of atomic propagations, including co-citation, in fact, the way that co-citation is exploited does not address the above definition.

All the above indicate a gap in the literature, leading to the need of designing a new method of propagating trust, based on 'common friends' (homophily phenomenon) and incorporating it into a trust-based recommender system.

5.4 Conclusion

This chapter presented the limitations of the current systems, leading to the need for a new system. Summarising, the major limitations of the current approaches, 'lack of transparency', 'grey-sheep', 'synonymy and polysemy' and 'security and privacy' can be alleviated with trust-based approaches. However, 'data sparsity' is the major limitation characterising all the current systems, which affects not only the item-ratings matrix, but also the trust matrix of the trust-based systems. This leads also to the 'cold-start' problem' which is another typical limitation of the current systems. Furthermore, the examination and comparison of the main literature in graph-based models shows that current approaches propagate trust based purely on the direct propagation strategy of the atomic propagation. However, the homophily phenomenon on which propagation is based is not fully addressed by the direct propagation. In final consideration, it is apparent that there is a need for a new approach that fully exploits the propagative property of trust, as defined by the homophily phenomenon, considering not only 'the friend of my friend' to propagate trust but also the 'common friends' intuition and, furthermore, addressing the data sparsity problem in recommender systems. The next chapter introduces such a novel approach based on the homophily phenomenon by exploiting structural information of the trust graph to infer and calculate the trust value.

Chapter 6

A novel method to infer trust

One of the fundamental issues for a successful recommender system, as highlighted in Section 3.2.1, is the calculation of similarity between users, for a collaborative filtering system or between items, for a content-based system. In trust-based systems, the weight of a trust relationship is used for measuring the similarity between users. Additionally, trust metrics calculate the similarity between users while computing and inferring new trust relationships. Similarity, thus, constitutes the basis for providing recommendations and/or inferring trust. In collaborative filtering and trust-based recommender systems, the user-to-user similarity is usually computed based on the ratings matrix. Similarity measures compute the correlation between users, comparing the rating values of the common rated items and the score is used as an input to a recommendation algorithm or to a trust inference algorithm in order to produce implicit trust values for a trust network. Additionally, in social networks, similarity is a basic measure leading to link association between users, but also impacts information diffusion (De Choudhury *et al.*, 2010) due to the social diffusion that exists between connected members. The association of similar users in a social network is the effect of the *'homophily phenomenon',* which will be discussed in Section 6.1.

Similarity in social networks can be captured from information provided by the graph structure, such as the number of common neighbours or the number of short paths between users or other contextual information regarding the nodes. Graph similarity measures try to answer the question: *"given a node which other nodes are similar to this node?"* by manipulating any available information given in the network. The procedure of link prediction is mainly based on graph similarity measures, as analysed in Section 4.3.2.

6.1 The homophily phenomenon

Homophily is a concept observed and launched as a term in sociology (Homans, 1951; Lazarsfeld and Merton, 1954; Laumann, 1966) referring to the tendency of humans in a social environment to form more associations with similar parties than with dissimilar ones. *"Similarity breeds connection"* claim McPherson *et al.* (2001) and give attribution of the old proverb *"birds of a feather flock together",* as being the heart of the homophily concept, to Lazarsfeld and Merton (1954) who, in turn, attributed the phrase to Burton (1621). Similarly, Burton acknowledged the classic Western philosophers Aristoteles and Plato who argued that *"people love those who are like themselves"* and *"similarity begets friendship",* respectively.

The homophily phenomenon affects a range of social relationships, varying from friendships, marriages and partnerships to organisational memberships and information exchange. McPherson *et al*. (2001) also state that homophily implies that social distance translates to network distance, but also implies that social entities with high degrees tend to be localised in social space and obey certain fundamental dynamics following social forms. Lazarsfeld and Merton (1954) distinguished two types of homophily:

- **status homophily**, in which similarity is based on informal, formal or ascribed status, including the major sociodemographic dimensions (race, ethnicity, sex, age) and acquired characteristics (religion, education, occupation, behaviour patterns), and
- **value homophily**, which is based on values, attitudes and beliefs, including the wide variety of internal states presumed to shape our orientation towards future behaviour.

Community in a social network can occur by choice, due to similarities (status homophily) between ties or/and social influence (the tendency of people to follow the behaviour of their friends or their social group) since individuals may become more similar over time by influencing each other. Likewise, McPherson and Smith-Lovin (1987) identified 'choice homophily' and 'induced homophily' as the two mechanisms for the origin of homophily. In fact, homophily is a self-reinforcing phenomenon affecting not only the structure of the social network of a person, but also the information diffusion within the network through the social influence mechanism. Additionally, homophily is identified as a factor affecting information diffusion in social media (De Choudhury *et al.*, 2010), the diffusion of ideas, behaviours and new technologies (Anagnostopoulos, Kumar and Mahdian, 2008), the group composition in enterprises (Ruef *et al.*, 2003) and many other communication and social behaviours including even health-related issues (Christakis and Fowler, 2007; 2008; Centola, 2010; Rosenquist *et al.*, 2010).

6.2 Triadic closure

The mechanism of propagating trust is based on the homophily effect in social networks. As described earlier, homophily is the principle according to which people tend to associate with others, all having something in common (McPherson, Smith-Lovin and Cook, 2001) and, thus, forming homogeneous personal networks. Algorithms based on propagation to infer trust, as discussed thoroughly in Section 4.2, exploit the direct propagation from all the atomic propagations analysed by Guha *et al*. (2004). In fact, these studies infer trust based on the intuition that *"the friend of my friend is also my friend."* However, the homophily principle implies also that *"if two persons, not yet connected, have a common friend, then there is an increased possibility for these two persons to become friends."* In other words, this kind of relation corresponds to the co-citation of the atomic propagations described in Section 4.3.3. Additionally, propagation, as exploited in TidalTrust (Golbeck, 2005) and MoleTrust (Massa and Avesani, 2005), can be applied only in directed networks (Figure 6-1). This way, the concept of homophily is not fully exploited in these two algorithms, since two persons that might have a common friend will not be associated. The co-citation property of atomic propagation, as part of the homophily phenomenon, is also known as *'triadic closure*' in social networks.

Rapoport (1953) introduced the notion of triadic closure, purely based on the homophily phenomenon, stating that two strangers who possess a mutual friend will tend to become friends in the future. The term 'triadic closure' derives from the observation that, in a social network graph, this phenomenon can be perceived as triads tending to close up. Triadic closure is the special case of the cyclic closure, originating from the notion of transitivity (Rapoport, 1953), with cycles of length 3. The link formation process through triadic closure has the advantage of not depending on the features of the nodes involved in the process.

In applying the triadic closure effect in an undirected social network (Figure 6-1(a)), new relationships are formed, closing any triads in the social network. More specifically, we can see that links are formed between nodes sharing a common friend, i.e. A and B have a common friend D and, thus, they tend to be connected. In the same way, B and C tend to be connected as they share a common friend E. The final network formed after the triadic closure effect is illustrated in Figure 6-1(b).

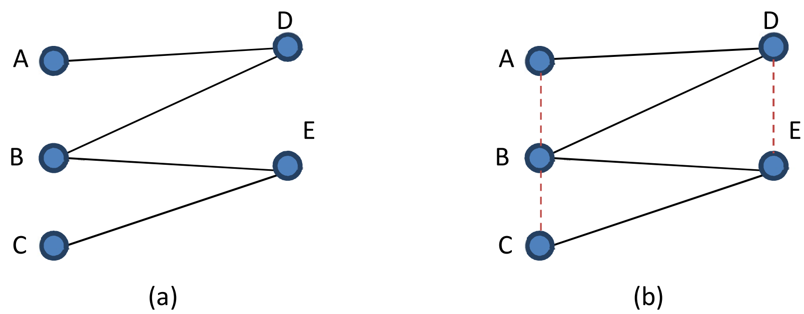


Figure 6-1 The triadic closure effect in an undirected network

In Social Network Analysis, triadic closure is observed as one of the most frequent processes of link formation. Thus, triadic closure mechanism has motivated the creation of metrics for measuring its prevalence. One of these metrics is the clustering coefficient (Watts and Strogatz, 1998; Newmann, 2003), defined as the probability of two randomly selected friends of a specific node being friends with each other. In other words, it is the fraction of pairs of friends connected to each other by edges in the social network graph or, alternatively, it is the number closed of triples in the social network graph. More formally, according to Newmann (2003), the clustering coefficient is defined as:

$$C = \frac{3 \times n_{\Delta}}{\mathrm{T}} \qquad \text{(Eq. 6.1)}$$

where n_{Δ} is the number of triangles in the graph and T is the number of connected triples of nodes. In effect, C measures the fraction of triples having their third edge filled in, to complete the triangle. As discussed in Section 4.3.1, the clustering coefficient is a local property of networks, while, in small-world networks, it is high, indicating that users form groups highly connected.

Kossinets and Watts (2009) observed that roughly 60% of all the new connections in a social network are formed via triadic closure. However, the establishment of new links in the social network is a rather dynamic phenomenon that can take place at any time with no restrictions. Consequently, exploiting the phenomenon on a static snapshot of the social network can contribute to the prediction of social links taking place in the future.

6.3 Inferring trust with triadic closure

Whilst triadic closure is a natural mechanism to make new connections in a social network, it is also a dynamic phenomenon, as mentioned above. In particular, Bianconi *et al.* (2014) found that communities emerge naturally via triadic closure, especially when the network is very sparse. Nevertheless, it is plausible to incorporate this mechanism into a trust-based recommender system, by predicting trust links on a static snapshot of the trust network. This intuition is at the basis of the method proposed and analysed in this section.

By exploiting the triadic closure property of social networks, a new method is proposed to infer new trust relationships in a trust network. Specifically, the proposed model relies on the intuition that *"if two people in a social network have a friend in common, then there is an increased possibility that they will become friends themselves too."*.

To illustrate this, suppose we have a directed graph representing a trust network of Figure 6-2(a) where A, B, C, D, E, F and G are the nodes, indicating the users, and arrows represent the trust relationships between users.

Applying the triadic closure principle on this trust network, we can infer new trust relationships. Therefore, since A and B trust D, then there is an increased possibility that A trusts B and B trusts A. In other words, A and B share a common trusted party and, thus, there is a tendency to trust each other. Similarly, since D, E and G trust F, then there is an increased possibility that D and E trust each other, D and G trust each other and E and G trust each other. In this case, we have three users sharing a common trusted party. It can be considered as three different cases of two users, sharing a common trustee. It can also be compared with a real life case of a meeting of four people where three of them are complete strangers, while they are all familiar with the fourth person. After the meeting, there is an increased possibility that these three persons will communicate with each other again.

The resulting network, after applying the triadic closure effect on the trust network of Figure 6-2(a), is illustrated in Figure 6-2(b). Notice that the new relationships represented by the red dashed lines are mutual, as indicated by the two-headed arrows.

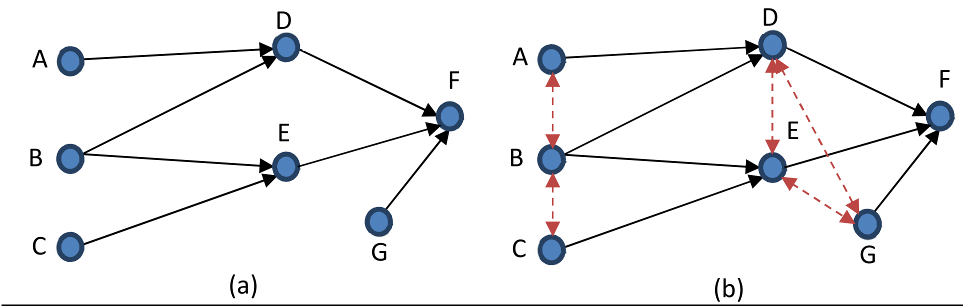


Figure 6-2 The triadic closure effect on a directed trust network

The algorithm to infer trust relationships is called TriadicClosure and is briefly outlined below.

Algorithm 6 TriadicClosure

Input: trust network
Output: expanded trust network
Begin
for each pair of users a, b in the trust network not adjacent
    Set Ta= T'a =trust network of user a
    Set Tb= trust network of user b
    if $|Ta \cap Tb|>0$
        T'a.Add(b)
        ADD b to T'b
    end if
end for
return the new trust network

The TriadicClosure algorithm takes as input the existing trust network and checks the neighbourhood (trust network) of a pair of users. If the two users are not already connected and the intersection of these two neighbourhoods is greater than zero (if these two users share at least one neighbour), then each one of these users is added to the trust network of the other user, viz. they are mutually trusted. This step is repeated until the whole trust network is examined. Finally, the algorithm outputs the expanded trust network consisting of the trust networks of all the users.

6.4 Calculating trust with JaccardCoefficient

In the previous section, we examined how the TriadicClosure algorithm infers new trust relationships in an existing trust network. However, these trust relationships, although directed, are not weighted. In real-world social and trust networks, relationships can be found as directed or undirected and the former can be found either as weighted or unweighted. In the TriadicClosure approach, as we saw in the example (Figure 6-2) of the previous section, the relationships of the trust network must be directed. The algorithm can infer trust relationships based on the direction of the relationships. In binary trust networks, trust weight is either 1, indicating full trust, or 0, indicating no trust. Additionally, there are trust networks with gradual trust weights of various scales, but not being the usual case.

Hence, after inferring the new trust network, the next step could be to give a weight to the inferred relationships. As mentioned, TidalTrust and MoleTrust use the weighted average (Eq. 4.6) to calculate the unknown trust value. Consequently, one possible solution would be to use the weighted mean for calculating the unknown trust value of the relationships inferred from the TriadicClosure algorithm. In the experimental study (Section 7.1), the TriadicClosure inference, combined with the weighted average for calculating trust, will be one of the compared methods.

However, the weighted mean algorithm is meaningless in binary trust networks where trust weights are not gradual, since the calculated trust value would be always equal to 1. Therefore, a new method is proposed to calculate the unknown trust value of an inferred trust relationship. The proposed method is called JaccardCoefficient and it is a modified version of the Jaccard coefficient similarity (Eq. 4.2). Specifically, the trust weight of an inferred trust relationship between an active user a and a target user u is defined as the ratio of the number of common neighbours of the two users over the number of the total neighbours of the two users and is given by the formula:

$$t_{a,u} = \frac{|T_a \cap T_u|}{|T_a \cup T_u|} \qquad \text{(Eq. 6.2)}$$

where T_a is the trust network (neighbourhood) of the active user a and T_u is the trust network of the target user u with $a \notin T_u$ and $u \notin T_a$, meaning that the two users are not yet connected within the initial trust network. The calculated trust weight can take values between 0 and 1. Thus, the specific method can be used to infer trust values for trust relationships on a continuous scale.

Algorithm 7 JaccardCoefficient trust weight calculation

Input: trust network
Output: updated trust network with new trust values
Begin
for each pair of users a, b in the trust network not adjacent
    if |Ta ∩ Tb|>0

$$t_{a,b} = \frac{|T_a \cap T_b|}{|T_a \cup T_b|}$$

    end if
end for
return the updated trust network

6.5 Incorporating *TriadicClosure* in the recommendation process

As already highlighted, typical recommender systems suffer from sparsity of the ratings matrix, but also suffer from the cold-start problem for new users or new items. Incorporating trust in recommender systems alleviates this problem by providing rating predictions based on the trust network of each user. Although these methods significantly improve prediction accuracy and ratings coverage (Avesani and Massa, 2005), let us consider that real-world trust networks follow the power law distribution, meaning that most users have a small neighbourhood and few users have a large trust network. The sparsity of the trust matrix affects the number of the recommendations that can be produced. Increasing the neighbourhood of users, in other words expanding their trust network, can increase the rating coverage and, thus, the number of the produced recommendations.

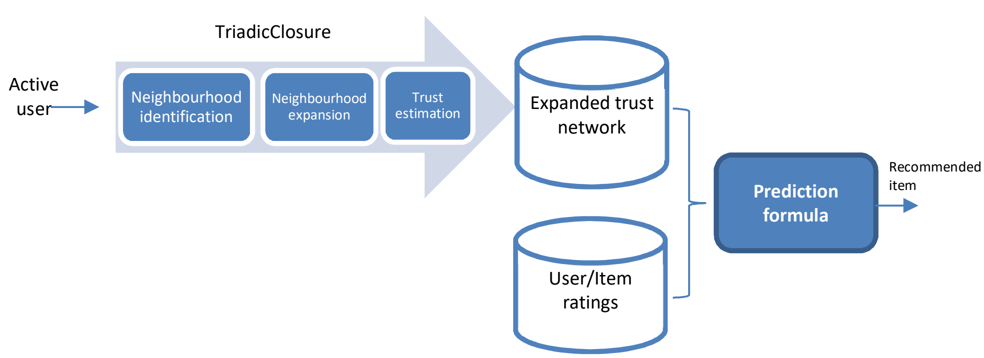


Figure 6-3 Incorporating TriadicClosure in the recommendation process

Prediction of triads tending to close in a trust network and, thus, expanding an existing trust network, can be exploited in a trust-based recommender. Incorporating the TriadicClosure algorithm in the recommendation process is the proposed method to increase not only user coverage, but also rating coverage in a recommender system. The whole recommendation process is illustrated in Figure 6-3. The first step, taking as input the existing trust network, is to infer the new relationships and extract the expanded trust network. In this step, the trust value for the predicted relationship can be calculated by various methods, such as the weighted average (Eq. 4.6) or the JaccardCoefficient (Eq. 6.2). The next step is similar to the methods presented in Section 4.2 where the trust metrics are incorporated into recommendation algorithms. Therefore, for predicting a recommended rating $p_{a,i}$ for an active user a and an item i, any of the two formulas (Eq. 3.14) and (Eq. 3.15) can be used. As stated, these two formulas were modified to be used in TidalTrust (Eq. 4.7) and MoleTrust (Eq. 4.8), respectively.

6.5.1 An example with synthetic data

The current section exemplifies the whole process described in the previous section, step-by-step. Assuming there is recommender system with eight items, denoted as i_j, where $j \in [1, 8]$ and seven users, denoted as u_k where $k \in [1, 7]$. Each user u_k can rate items i_j with integer valued ratings $r_{kj} \in [1, 5]$. The user-item ratings matrix is depicted in Table 6-1. Additionally, users can declare their trust to other users forming their own neighbourhood. The total trust network is depicted in Figure 6-4 with user-user (adjacency) matrix represented in Table 6-2. Note that the trust network in this example is binary, hence, value 1 indicates the directed trust.

Suppose now that we want to predict a rating value $p_{2,3}$ for item i_3 for an active user $u_{2,}$ as indicated with the question mark in Table 6-1. From the ratings matrix, we can observe that the specific item has been rated by users u_1 and u_7. However, these two users do not belong to the neighbourhood of user u_2 as they are not directly connected.

Following the process as depicted above in Figure 6-3, the first step of this would be to expand the existing trust network of the user by predicting new connections. Thus, as described previously, the user-user matrix is initially processed to produce the expanded trust network. Consequently, the system identifies that u_2 has two neighbours directly connected to the user, which forms the trust network Tu_k of the user, thus $Tu_2=\{u_4,\ u_5\}$. Choosing to infer trust relationship through propagation, we can observe (blue arrows in Figure 6-5(a) and blue-bold values in Table 6-3) that only three relationships can be inferred: $u_1 \rightarrow u_5$, $u_2 \rightarrow u_6$, $u_3 \rightarrow u_6$. Accordingly, we can note that, even with the propagation strategy, user u_2 would connect only with u_6 who has not also provided any rating for item i_3 and, thus, no prediction can be made for this item.

On the other hand, by exploiting the TriadicClosure algorithm, the trust network of user u_2 is expanded, as illustrated with red arrows in Figure 6-5(a) and thus $Tu_2=\{u_1,\ u_3,\ u_4,\ u_5,\ u_7\}$. The result of applying the TriadicClosure algorithm on all users of Table 6-2 is the trust matrix of Table 6-3 with the inferred relationships highlighted. Initially, the trust weight can be set to 1 following the binary mode of the original trust network. Nevertheless, if we want to estimate the trust value of the new trust relationships, we can use any similarity or trust metric. By applying the JaccardCoefficient metric (Eq. 6.2) proposed in Section 6.4, we can calculate the trust value of the inferred relationships, resulting in Table 6-4. Note that this metric could also be applied on the relationships inferred through propagation.

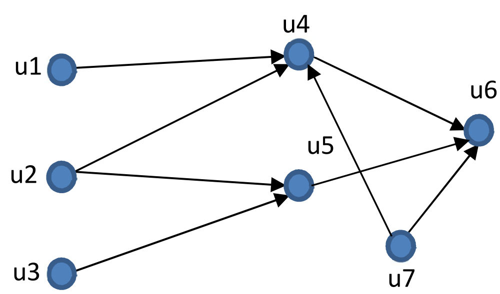


Figure 6-4 Example trust network

Table 6-1 User-item ratings matrix

	i1	i2	i3	i4	i5	i6	i7	i8
u1	3		4	5			4	
u2			?					
u3		3			5			
u4	3	2		5	5			1
u5		4		5			4	
u6		2						
u7		3	3	5		3	5	2

Table 6-2 User-user (adjacency) matrix of the trust network of Figure 6-4

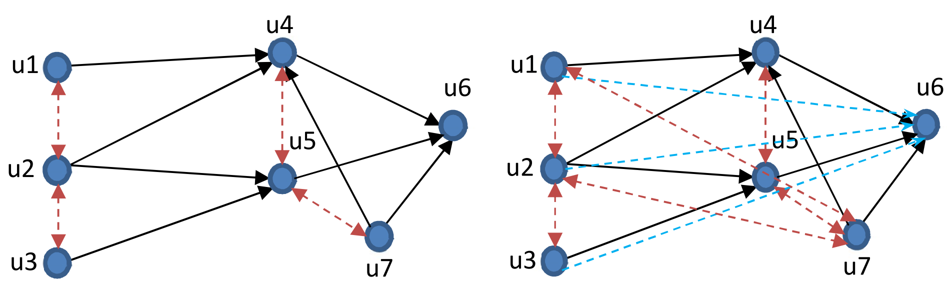
	u1	u2	u3	u4	u5	u6	u7
u1				1			
u2				1	1		
u3					1		
u4						1	
u5						1	
u6							
u7						1	

Table 6-3 Updated user-user trust matrix with inferred trust relationships

	u1	u2	u3	u4	u5	u6	u7
u1		1		1		1	1
u2	1		1	1	1	1	1
u3		1			1	1	
u4					1	1	
u5				1		1	1
u6							
u7				1	1	1	

Table 6-4 Trust matrix with JaccardCoefficient values of the inferred relationships

	u1	u2	u3	u4	u5	u6	u7
u1		0.5		1			1
u2	0.5		0.5	1	1		0.5
u3		0.5			1		
u4					1	1	
u5				1		1	1
u6							
u7				1	1	1	



(a) (b)

Figure 6-5 Updated trust network of Figure 6-4 after TriadicClosure and propagation

After the inference of the new relationships, the trust matrix of the expanded network (Table 6-4) and the user-item ratings matrix, as depicted in Table 6-1, are the inputs of the algorithm to predict a rating value for item i_3 for the active user u_2. In this example, we can use the trust-based weighted mean (Eq. 4.7) to calculate the unknown rating $p_{2,3}$:

$$p_{2,3} = \frac{\sum_{u \in R^T} t_{2,u} r_{u,3}}{\sum_{u \in R^T} t_{2,u}}$$

where R^T represents the trusted users by u_2 that have also rated i_3 thus, $R^T = \{u_1,\ u_7\}$. Consequently, $p_{2,3}$ is calculated as:

$$p_{2,3} = \frac{\sum_{u \in R^T} t_{2,u} r_{u,3}}{\sum_{u \in R^T} t_{2,u}} = \frac{t_{2,1} r_{1,3} + t_{2,7} r_{7,3}}{t_{2,1} + t_{2,7}} = \frac{0.5 * 4 + 1 * 3}{0.5 + 1} = 3.33$$

The procedure continues until all the items rated by trusted users, either directly or through triadic closure, are covered. Note that, with the propagation method, only three new relationships are inferred, with common trustee the user u_6. We can observe also that u_6 has provided rating for only one item, i_2. This signifies that, in the particular case of our example, predictions from any inferred relationships through propagation could be made only for i_2. Comparing the inference through the TriadicClosure method with the propagation we can observe that the TriadicClosure could increase not only user coverage, but also rating coverage. In the next chapter, a series of experiments with real-world data verifies this observation and, moreover, proves that the TriadicClosure also increases accuracy.

6.6 Conclusion

Homophily states that people tend to associate with those having something in common and is a phenomenon affecting social relationships at various levels. Based on this phenomenon, triadic closure is a fundamental mechanism of link formation in social networks which can be perceived as triads tending to close up. About 60% of all the new connections in a social network are formed via triadic closure (Kossinets and Watts, 2009).

Based on this mechanism, a novel method is proposed to infer trust relationships in trust networks. The proposed method is called TriadicClosure and is based on the intuition that *"if two people in a social network have a friend in common, then there is an increased possibility that they will become friends themselves too."* In addition, a novel method to calculate the trust weight of a trust relationship is proposed based on the Jaccard Coefficient similarity metric exploited on the trust network. Both methods and their algorithms are thoroughly described along with the way to incorporate the proposed methods in the recommendation process. Finally, both methods were evaluated with synthetic data to prove their validity. As a next step, there is a need to further evaluate the two methods with a real-world dataset and compare their performance with other state-of-the-art methods.

Chapter 7

Experimental evaluation

This chapter presents a series of experiments conducted for testing and evaluating the effectiveness of the proposed methods discussed in the previous chapter. The empirical study was conducted with two different datasets (Filmtrust and Epinions), aiming to compare the performance of the proposed methods with different state-of-the-art trust-based methods. Recall that the experimental evaluation intends to address the specific questions set in Section 2.5.3.

7.1 Experimental design

The experiments are divided into four conceptual stages as described in Section 2.5.3.

- **Stage 1** **TriadicClosure basic evaluation**, in which the TriadicClosure algorithm is compared with basic trust-based approaches.
- **Stage 2** **TriadicClosure total performance**, in which the TriadicClosure algorithm is incorporated into the state-of-the-art trust-based approaches.
- **Stage 4** **JaccardCoefficient performance**, in which the JaccardCoefficient method is compared against all the above methods.
- **Stage 5** **Performance of TriadicClosure and JaccardCoefficient for different views**, where all the above methods are compared for different views (as defined in Section 2.5.3) of users.

The state-of-the-art trust-based approaches with which TriadicClosure and JaccardCoefficient will be compared with, are the TidalTrust and MoleTrust as described earlier in Sections 4.5.4 and 4.5.5 respectively. Specifically, the methods used in the experiment are defined as:

- **TT** is the TidalTrust algorithm (Eq. 4.7)
  - **TTx** is the TidalTrust implemented for propagation length x, where x={1,2,3,4}
- **MT** is the MoleTrust algorithm (Eq. 4.8)
  - **MTx** is the MoleTrust implemented for propagation length x, where x={1,2,3,4}
- **TC** is the TriadicClosure method as described in algorithm 6
  - **TC CF** is the TriadicClosure implemented for collaborative filtering (CF) as in (Eq. 4.8)
  - **TC WM** is the TriadicClosure implemented for weighted mean (WM) as in (Eq. 4.7)
- **MTx+TC** is the combined TriadicClosure with MoleTrust for propagation length x, where x={1,2,3,4}
- **TTx+TC** is the combined TriadicClosure with TidalTrust for propagation length x, where x={1,2,3,4}
- **JC** is the JaccardCoefficient method as described in algorithm 7.
- **METHODx+JC** is the incorporation of JaccardCoefficient into one of the stand-alone METHODs (MT or TT) or the combined METHODs with the TriadicClosure (MT+TC or TT+TC) for propagation length x.

7.2 Evaluation results and analysis

The current section presents the evaluation results of all the experiments conducted as described in Section 7.1 which are divided into four stages:

<u>Stage 1</u> TriadicClosure basic evaluation

This is the initial stage, in which the experiments intend to answer the questions:

(Q1) How does the TriadicClosure algorithm perform on accuracy compared with different state-of-the-art trust-based methods?

(Q2) What is the impact of the TriadicClosure on coverage?

(Q3) How does the TriadicClosure algorithm perform on large datasets?

Initially, the TriadicClosure was implemented in two versions: i) TriadicClosure with weighted mean (TC-WM) and ii) TriadicClosure with collaborative filtering (TC-CF). The results of the experiments on the Filmtrust dataset (Table 7-1) showed that in terms of accuracy, the TC-CF presents the best performance of all the other compared methods. Recall that lower values of RMSE and MAE indicate better accuracy. Notice, that TC-WM compared with TidalTrust (TT) performs better in terms of accuracy but this is not the case when compared with MoleTrust (MT). The reason is that incorporating collaborative filtering in trust-based systems, may increase the accuracy, as one can notice from the comparison of accuracy performance also between TT and MT (recall that MT incorporates CF). The results of the experiments on the Epinions dataset (Table 7-2) on the other hand show that not only the TC-CF but also the TC-WM increases the accuracy compared with the TT and MT algorithms. This means that on large datasets, the positive impact of the TriadicClosure on accuracy is greater than the impact of incorporating the CF algorithm.

In terms of coverage, we can observe that on Filmtrust the ratings coverage (RC) and the user coverage (UC) are improved with TriadicClosure at about 34.50%. Moreover, on Epinions, the improvement of coverage is remarkable with an increase of 578.52% for RC and 107.93% for UC. Finally, the percentage improvement of the overall performance (FMeasure) with the implementation of TriadicClosure is about 18% for the first propagation length for both datasets.

Table 7-1 First-step evaluation results on Filmtrust dataset

Algorithm	**RMSE**	**MAE**	**RC**	**UC**	**FMeasure**
TT	1.133	0.852	21.20	32.23	1.387
MT	1.004	0.758	21.15	31.03	1.447
TC-WM	1.048	0.800	28.53	33.69	1.439
TC-CF	0.943	0.716	28.47	32.23	1.489

Table 7-2 First-step evaluation results on Epinions dataset

Algorithm	**RMSE**	**MAE**	**RC**	**UC**	**FMeasure**
TT	1.370	0.969	4.47	30.26	1.146
MT	1.306	0.949	4.47	30.26	1.171
TC-WM	1.214	0.891	30.33	62.92	1.362
TC-CF	1.171	0.869	30.32	62.91	1.382

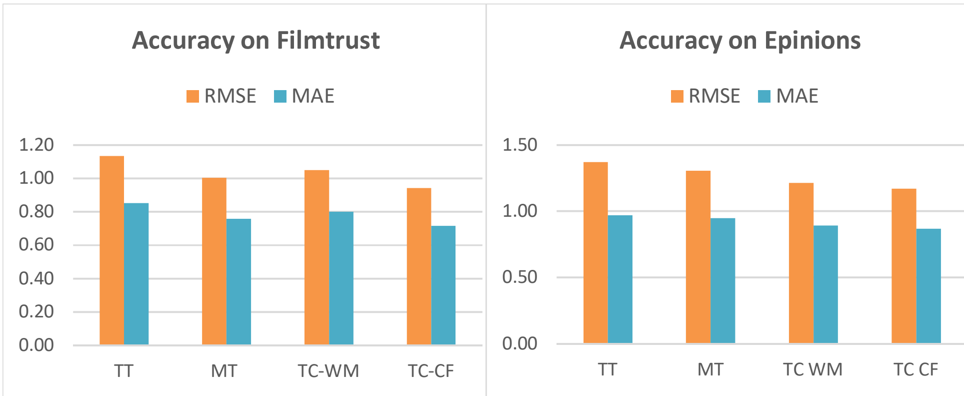


Figure 7-1 Accuracy performance

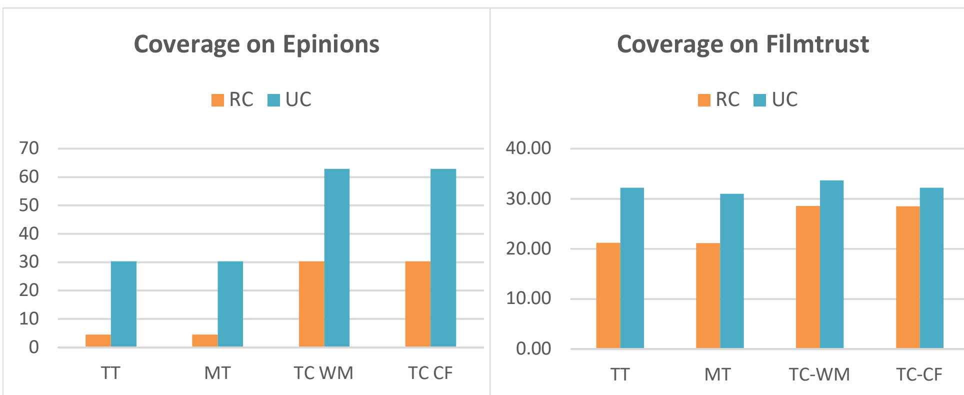


Figure 7-2 Coverage performance

<u>Stage 2</u> TriadicClosure total performance

The experiments of this stage, intend to answer the questions:

(Q4) How does the TriadicClosure algorithm perform on accuracy and coverage, when integrated within other state-of-the-art trust-based methods?

(Q5) What is the impact of propagation on the TriadicClosure?

In this stage, TT and MT are implemented for different propagation lengths and specifically for lengths 1, 2, 3, 4. Additionally, TC is integrated with TT and MT for the same propagation lengths. The results of the experiments on the Filmtrust dataset (Table 7-3) show that the best performance is achieved when TriadicClosure is combined with MoleTrust for propagation length 4 (MT4+TC). Similar to the observations of the experimental results of stage 1, we can notice that the integration of TriadicClosure to TidalTrust outperforms the original version of TidalTrust but does not outperform the original version of MoleTrust. However, when the dataset is large and sparse, like the Epinions, the results (Table 7-4) indicate that TriadicClosure has a positive impact on the performance improvement.

Regarding coverage (RC and UC), the results on both the datasets showed that TriadicClosure is steadily improving coverage as the propagation length increases, nevertheless the percentage improvement compared to the original methods is minimal when propagation length is greater than 3 (Table 7-5).

The evaluation results of this stage of experiments show that the integration of TriadicClosure with other state-of-the-art trust-based methods can improve their performance, yet, the percentage improvement tends to zero when propagation length is greater than 3.

Table 7-3 Evaluation results for different propagation lengths on Filmtrust dataset

Algorithm	RMSE	MAE	RC	UC	FMeasure
TT1	1.133	0.852	21.2	32.23	1.387
TT2	1.046	0.795	27.96	33.82	1.439
TT3	1.013	0.771	30.37	33.95	1.458
TT4	0.999	0.762	31.32	33.95	1.465
MT1	1.004	0.758	21.15	31.03	1.447
MT2	0.940	0.713	27.89	32.23	1.489
MT3	0.912	0.692	30.31	32.29	1.506
MT4	0.904	0.686	31.36	32.29	1.511
TT1+TC	1.048	0.800	28.53	33.69	1.439
TT2+TC	1.036	0.790	29.17	34.02	1.445
TT3+TC	1.009	0.769	30.95	34.15	1.460
TT4+TC	0.998	0.762	31.82	34.15	1.467
MT1 +TC	0.943	0.716	28.47	32.23	1.489
MT2 +TC	0.933	0.708	29.11	32.43	1.494
MT3 +TC	0.910	0.691	30.89	32.49	1.507
MT4 +TC	**0.903**	**0.686**	**31.86**	**32.49**	**1.512**

Table 7-4 Evaluation results for different propagation lengths on Epinions dataset

Algorithm	RMSE	MAE	RC	UC	FMeasure
TT1	1.370	0.969	4.47	30.26	1.146
TT2	1.249	0.914	25.83	61.58	1.340
TT3	1.156	0.861	44.23	67.2	1.400
TT4	1.121	0.841	50.87	67.79	1.419
MT1	1.306	0.949	4.47	30.26	1.171
MT2	1.202	0.890	25.84	61.63	1.362
MT3	1.120	0.836	44.27	67.24	1.417
MT4	1.089	0.816	50.93	67.84	1.435
TT1+TC	1.214	0.891	30.33	62.92	1.362
TT2+TC	1.209	0.889	32.35	64.37	1.366
TT3+TC	1.154	0.860	44.83	67.84	1.401
TT4+TC	1.121	0.841	51.07	68.37	1.419
MT1 +TC	1.171	0.869	30.32	62.91	1.382
MT2 +TC	1.166	0.866	32.35	64.37	1.387
MT3 +TC	1.118	0.835	44.85	67.83	1.418
MT4 +TC	**1.089**	**0.816**	**51.12**	**68.38**	**1.435**

Table 7-5 Impact of TriadicClosure on TidalTrust and MoleTrust

Filmtrust		**% improvement**			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	7.47	3.75	34.58	4.53
	2	0.94	0.43	4.33	0.59
	3	0.36	0.17	1.91	0.59
	4	0.15	0.09	1.60	0.59
TC on MT	1	6.13	2.91	34.61	3.87
	2	0.79	0.35	4.37	0.62
	3	0.18	0.10	1.91	0.62
	4	0.10	0.07	1.59	0.62
Epinions					
TC on TT	1	11.41	18.80	578.52	107.93
	2	3.23	1.97	25.24	4.53
	3	0.20	0.10	1.36	0.95
	4	-0.02	0.00	0.39	0.86
TC on MT	1	10.34	18.08	578.30	107.90
	2	2.99	1.80	25.19	4.45
	3	0.19	0.09	1.31	0.88
	4	-0.01	0.00	0.37	0.80

<u>Stage 3</u>   <u>J</u>accardCoefficient performance

In this stage the experiments intend to answer the question:

(Q6) How does the JaccardCoefficient algorithm perform on accuracy and coverage, when implemented within TriadicClosure algorithm and other state-of-the-art trust-based methods?

Evaluation results from the previous stage showed that when propagation length is greater than 3 there is no significant improvement of the performance for none of the methods examined while the cost of calculation is substantially large. Hence, it would be pointless to compare any improvement on performance for propagation length greater than 3. Thus, the experiments from now on (for stage 3 and stage 4) will examine the performance of the algorithms for propagation lengths 1, 2 and 3.

In this stage, the JaccardCoefficient algorithm is incorporated in all the examined methods and specifically the TidalTrust (TT) the MoleTrust (MT) and the TriadicClosure (TC) combined with the other two methods (TT+TC and MT+TC) for different propagation lengths.

Comparing the performance of the JaccardCoefficient on Filmtrust dataset (Table 7-6) incorporated in the examined methods, with the performance of the original methods (Table 7-3), we can observe that the proposed method does not improve the performance of any method except in the case of propagation length 1 for the original methods of TidalTrust and MoleTrust (TT1 and MT1). Particularly when JaccardCoefficient is combined with TriadicClosure the performance is much worse than that of the original method. The same observations arise from the experiments on the Epinions dataset (Table 7-7). This can be explained due to the fact that in both datasets the trust statements are binary, however, the inferred weight of a relationship with JaccardCoefficient is gradual with a value between 0 and 1. Hence, the predicted trust weight (a value between 0 and 1) is always compared to a larger value of trust weight which is always 1. Moreover, in a large dataset such as the Epinions, the predicted trust weight takes really small values since the denominator ($T_a \cup T_u$) of the formula (Eq. 6.2) can take large values. Finally, the proposed method of JaccardCoefficient for calculating the weight of the inferred trust relationships seems to worsen the performance compared to the weighted average (Eq. 4.6) which in trust networks with binary trust weights produces also binary results, in other words, 1s for the inferred relationships.

Table 7-6 Performance of JaccardCoefficient combined with other methods on Filmtrust dataset

Algorithm	RMSE	MAE	RC	UC	FMeasure
			Filmtrust		
TT1+JC	1.088	0.814	9.23	11.07	1.349
TT2+JC	1.008	0.762	12.15	11.41	1.409
TT3+JC	1.000	0.757	12.27	11.41	1.413
TT1+TC+JC	1.074	0.816	28.53	33.69	1.427
TT2+TC+JC	1.063	0.806	29.17	34.02	1.432
TT3+TC+JC	1.026	0.780	30.95	34.15	1.452
MT1+JC	1.002	0.750	9.23	11.07	1.386
MT2+JC	0.940	0.707	12.11	11.41	1.439
MT3+JC	0.937	0.706	12.22	11.41	1.441
MT1+TC+JC	0.960	0.728	28.47	32.23	1.481
MT2+TC+JC	0.950	0.720	29.11	32.43	1.486
MT3+TC+JC	0.924	0.701	30.88	32.49	1.501

Table 7-7 Performance of JaccardCoefficient combined with other methods on Epinions dataset

Algorithm	RMSE	MAE	RC	UC	FMeasure
			Epinions		
TT1+JC	1.365	0.967	3.72	23.10	1.120
TT2+JC	1.267	0.923	18.68	44.05	1.318
TT3+JC	1.178	0.873	43.37	66.17	1.388
TT1+TC+JC	1.243	0.907	29.20	62.38	1.347
TT2+TC+JC	1.279	0.928	31.42	64.06	1.331
TT3+TC+JC	1.179	0.873	44.54	67.80	1.389
MT1+JC	1.300	0.945	3.69	22.78	1.142
MT2+JC	1.215	0.897	17.26	41.45	1.338
MT3+JC	1.185	0.878	21.99	43.58	1.364
MT1+TC+JC	1.195	0.884	29.20	62.38	1.369
MT2+TC+JC	1.226	0.904	31.42	64.05	1.357
MT3+TC+JC	1.139	0.848	44.54	67.79	1.408

Table 7-8 Impact of JaccardCoefficient on all algorithms

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
JC on TT	1	3.93	-2.69	-56.47	-65.64
	2	3.62	-2.07	-56.55	-66.27
	3	1.25	-3.03	-59.60	-66.41
JC on TT+TC	1	-2.42	-0.84	0.00	0.00
	2	-2.62	-0.89	0.00	0.00
	3	-1.64	-0.54	0.00	0.00
JC on MT	1	0.26	-4.15	-56.37	-64.32
	2	0.01	-3.36	-56.59	-64.61
	3	-2.77	-4.29	-59.70	-64.68
JC on MT+TC	1	-1.79	-0.54	0.00	0.00
	2	-1.90	-0.56	0.00	0.00
	3	-1.54	-0.44	-0.03	0.00
Epinions					
JC on TT	1	0.427	-2.334	-16.775	-23.665
	2	-1.463	-1.617	-27.671	-28.471
	3	-1.945	-0.809	-1.958	-1.524
JC on TT+TC	1	-2.370	-1.095	-3.727	-0.853
	2	-5.846	-2.539	-2.852	-0.484
	3	-2.191	-0.884	-0.659	-0.053
JC on MT	1	0.468	-2.496	-17.395	-24.724
	2	-1.099	-1.742	-33.222	-32.753
	3	-5.852	-3.764	-50.320	-35.185
JC on MT+TC	1	-2.092	-0.932	-3.690	-0.853
	2	-5.130	-2.127	-2.821	-0.477
	3	-1.904	-0.737	-0.659	-0.053

Stage 4 Performance of TriadicClosure and JaccardCoefficient for different views

The final stage intends to answer the last question:

(Q7) What is the performance comparison of the two proposed methods on different views of users?

In this stage, the two proposed methods are examined for their performance on different views. Each dataset is split according to Massa and Avesani (2007) to views in order to examine the performance of the proposed algorithms for cold-start (who rated less than 5 items), heavy raters (who rated more than 10 items), grey-sheep users (who rated more than 4 items, and the average difference between their average rating and the mean rating of each item is greater than 1), controversial items (which received ratings with standard deviation greater than 1.5) and niche items (which received less than 5 ratings). The statistics of these views for each dataset are shown in Table 2-2 and Table 2-3. The results of the experiments for each view showed that for:

Cold-start users Table 7-9 and Table 7-10

> The impact of TC on TT and MT for cold-start users is positive for both datasets. However, we can observe that for Epinions when propagation level is 2 although the coverage is improved, the RMSE is increased causing the FMeasure to decrease. This may be caused by the fact that the Epinions dataset of the experiments contain only 22 cold-start users since it is a dataset crawled from Tang *et al*. (2012) not to include too many cold-start users.
>
> Examining the behavior of JC, we can observe that the impact is mainly negative to all algorithms. However, there are some cases such as when JC is integrated into the combined MT+TC and TT+TC methods, which seem to improve the performance when propagation length is 1. The main problem is detected on the large decrease in the ratings and user coverage, when JC is exploited, causing the FMeasure also to decrease although the RMSE shows improvement. Generally, the behavior of JC for cold-start users does not produce stable results while it can be mostly observed that it does not improve the overall performance of any of the algorithms.

Heavy raters Table 7-11 and Table 7-12

> The impact of TC on TT and MT for heavy raters is positive for both datasets. Especially for Epinions, the overall performance is increased with the ratings and user coverage to show dramatic improvement. However, we can observe that for Filmtrust, when propagation level is 2, although the RMSE is improved, the coverage is decreased causing also the FMeasure to decrease.
>
> Regarding JC we can observe that again the impact is mainly negative to almost all algorithms except when JC is integrated with MT+TC where all the performance measures for all propagation lengths is increased but only for

Filmtrust. Again, the behavior of JC for heavy raters does not produce stable results but we can mostly observe that it does not improve the overall performance of the majority of algorithms.

Grey-sheep users Table 7-13 and Table 7-14

The results for grey-sheep users are very similar to that of heavy raters for both proposed algorithms (TC and JC) and both datasets. In other words, the impact of TC is positive on TT and MT for both datasets and all the propagation lengths.

Regarding JC, we can observe again that the impact is mostly negative to all algorithms.

Controversial items Table 7-15 and Table 7-16

For controversial items, the results show that TC has a positive impact only on TT and MT for propagation length 1 and only on Epinions. In all the other cases (algorithms and propagation lengths) does not show any positive impact.

Similar to grey-sheep users and heavy raters, the impact of JC is mainly negative to all algorithms.

Niche items Table 7-17 and Table 7-18

For niche items, we can observe that TC shows a positive impact on TT and MT for both datasets and all propagation lengths. Especially on Epinions, the algorithm demonstrates extremely good results with coverage increased over 1000% and FMeasure increased over 100% for the first propagation length.

Again, similar to the other views, the impact of JC is generally negative to all algorithms

Concluding, the TriadicClosure algorithm shows improvement on the performance of all views, compared to the original methods of TidalTrust and MoleTrust. On the other hand, JaccardCoefficient does not give stable results, with the majority of them being negative for the performance of all the algorithms.

Table 7-9 Performance of all methods for **cold-start users** on the two datasets

Algorithm	RMSE	MAE	RC	UC	FMeasure	RMSE	MAE	RC	UC	FMeasure
	Cold-start users									
	Filmtrust					**Epinions**				
TT1	1.199	0.853	17.11	20.28	1.345	0.922	0.700	7.81	18.18	1.401
TT2	1.201	0.880	23.19	24.91	1.359	0.897	0.734	31.25	45.45	1.514
TT3	1.146	0.819	23.85	25.62	1.386	1.127	0.764	56.25	63.64	1.419
TT1+TC	1.171	0.861	23.52	24.56	1.373	0.914	0.624	42.19	50.00	1.515
TT2+TC	1.155	0.854	24.51	25.62	1.383	0.981	0.701	43.75	50.00	1.484
TT3+TC	1.126	0.813	25.00	26.33	1.397	1.149	0.800	56.25	63.64	1.408
TT1+JC	1.088	0.814	9.23	11.07	1.349	0.936	0.646	6.25	13.64	1.365
TT2+JC	1.008	0.762	12.15	11.41	1.409	0.742	0.591	23.44	31.82	1.574
TT3+JC	1.155	0.876	5.76	4.98	1.266	1.156	0.771	56.25	63.64	1.404
TT1+TC+JC	1.074	0.816	28.53	33.69	1.427	0.915	0.609	42.19	50.00	1.515
TT2+TC+JC	1.063	0.806	29.17	34.02	1.432	1.048	0.740	43.75	50.00	1.452
TT3+TC+JC	1.127	0.816	25.00	26.33	1.396	1.156	0.771	56.25	63.64	1.404
MT1	1.210	0.889	14.14	13.88	1.329	1.037	0.722	7.81	18.18	1.353
MT2	1.255	0.923	19.24	16.37	1.325	1.046	0.693	29.69	40.91	1.441
MT3	1.246	0.900	19.74	16.73	1.331	1.268	0.815	51.56	50.00	1.348
MT1+TC	1.195	0.862	19.90	16.73	1.355	1.196	0.766	40.63	45.45	1.378
MT2+TC	1.220	0.893	20.56	17.08	1.344	1.194	0.804	42.19	45.45	1.380
MT3+TC	1.223	0.896	20.89	17.44	1.344	1.284	0.832	51.56	50.00	1.341
MT1+JC	1.002	0.750	9.23	11.07	1.386	1.045	0.652	6.25	13.64	1.321
MT2+JC	0.940	0.707	12.11	11.41	1.439	0.729	0.419	20.31	22.73	1.572
MT3+JC	1.074	0.781	5.76	4.98	1.298	0.750	0.487	25.00	27.27	1.574
MT1+TC+JC	0.960	0.728	28.47	32.23	1.481	1.211	0.748	40.63	45.45	1.371
MT2+TC+JC	0.950	0.720	29.11	32.43	1.486	1.227	0.830	42.19	45.45	1.364
MT3+TC+JC	1.226	0.901	20.89	17.44	1.342	1.272	0.815	51.56	50.00	1.346

Table 7-10 Impact of TC and JC on all methods for **cold-start users**

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	2.38	2.08	37.50	21.05
	2	3.84	1.76	5.67	2.86
	3	1.74	0.81	4.83	2.78
TC on MT	1	1.32	1.94	40.70	20.51
	2	2.77	1.45	6.84	4.35
	3	1.81	0.98	5.83	4.26
JC on TT	1	4.33	-10.20	-76.92	-82.46
	2	3.67	-6.86	-75.18	-80.00
	3	-0.80	-8.63	-75.86	-80.56
JC on TT+TC	1	1.84	0.74	0.00	0.00
	2	0.86	0.34	0.00	0.00
	3	-0.10	-0.04	0.00	0.00
JC on MT	1	24.07	-3.03	-72.09	-74.36
	2	15.24	-1.74	-70.09	-69.57
	3	13.76	-2.46	-70.83	-70.21
JC on MT+TC	1	0.03	0.01	0.00	0.00
	2	-1.31	-0.56	0.00	0.00
	3	-0.25	-0.11	0.00	0.00
Epinions					
TC on TT	1	0.88	8.16	440.00	175.00
	2	-9.40	-1.99	40.00	10.00
	3	-1.96	-0.76	0.00	0.00
TC on MT	1	-15.34	1.85	420.00	150.00
	2	-14.14	-4.23	42.11	11.11
	3	-1.24	-0.57	0.00	0.00
JC on TT	1	-1.56	-2.60	-20.00	-25.00
	2	17.23	3.97	-25.00	-30.00
	3	-2.62	-1.01	0.00	0.00
JC on TT+TC	1	-0.12	-0.04	0.00	0.00
	2	-6.83	-2.18	0.00	0.00
	3	-0.64	-0.26	0.00	0.00
JC on MT	1	-0.86	-2.38	-20.00	-25.00
	2	30.27	9.08	-31.58	-44.44
	3	40.82	16.73	-51.52	-45.45
JC on MT+TC	1	-1.28	-0.54	0.00	0.00
	2	-2.80	-1.17	0.00	0.00
	3	0.91	0.43	0.00	0.00

Table 7-11 Performance of all methods for **heavy raters** on the two datasets

Algorithm	RMSE	MAE	RC	UC	FMeasure	RMSE	MAE	RC	UC	FMeasure
	Heavy raters									
	Filmtrust					**Epinions**				
TT1	1.135	0.854	21.53	36.24	1.386	1.368	0.968	4.47	30.73	1.147
TT2	1.048	0.797	28.25	37.18	1.438	1.248	0.913	25.77	61.84	1.340
TT3	1.014	0.772	30.75	37.18	1.457	1.156	0.861	44.13	67.25	1.399
TT1+TC	1.048	0.800	28.85	36.97	1.439	1.214	0.891	30.25	63.14	1.362
TT2+TC	1.037	0.790	29.42	37.18	1.445	1.208	0.889	32.27	64.59	1.366
TT3+TC	1.011	0.770	31.27	37.18	1.460	1.154	0.860	44.73	67.90	1.401
TT1+JC	1.091	0.816	9.55	13.91	1.352	1.362	0.964	3.72	23.49	1.121
TT2+JC	1.014	0.765	12.57	14.02	1.409	1.267	0.922	18.65	44.27	1.318
TT3+JC	1.005	0.760	12.69	14.02	1.414	1.179	0.873	44.43	67.87	1.388
TT1+TC+JC	1.075	0.818	28.85	36.97	1.426	1.243	0.907	29.12	62.59	1.347
TT2+TC+JC	1.066	0.809	29.42	37.18	1.431	1.279	0.928	31.34	64.27	1.332
TT3+TC+JC	1.028	0.781	31.27	37.18	1.451	1.179	0.873	44.43	67.87	1.388
MT1	1.000	0.756	21.53	36.24	1.450	1.303	0.948	4.47	30.73	1.172
MT2	0.935	0.711	28.25	37.18	1.492	1.201	0.889	25.79	61.88	1.363
MT3	0.907	0.689	30.76	37.18	1.509	1.119	0.836	44.16	67.31	1.418
MT1+TC	0.938	0.713	28.85	36.97	1.492	1.170	0.869	30.25	63.13	1.383
MT2+TC	0.927	0.704	29.42	37.18	1.497	1.165	0.866	32.27	64.59	1.387
MT3+TC	0.905	0.688	31.28	37.18	1.510	1.117	0.834	44.73	67.90	1.419
MT1+JC	1.003	0.750	9.55	13.91	1.390	1.296	0.943	3.70	23.16	1.143
MT2+JC	0.943	0.710	12.52	14.02	1.441	1.214	0.896	17.23	41.69	1.339
MT3+JC	0.939	0.708	12.64	14.02	1.443	1.184	0.878	21.95	43.77	1.364
MT1+TC+JC	0.955	0.726	28.85	36.97	1.483	1.194	0.884	29.13	62.59	1.370
MT2+TC+JC	0.945	0.717	29.42	37.18	1.489	1.225	0.903	31.35	64.27	1.357
MT3+TC+JC	0.920	0.699	31.27	37.18	1.503	1.138	0.848	44.43	67.87	1.408

Table 7-12 Impact of TC and JC on all methods for **heavy raters**

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	7.61	3.79	33.99	2.01
	2	1.03	0.46	4.14	0.00
	3	0.35	0.16	1.70	0.00
TC on MT	1	3.85	-2.52	-55.63	-61.60
	2	3.29	-2.01	-55.51	-62.29
	3	0.89	-2.98	-58.72	-62.29
JC on TT	1	-2.55	-0.88	0.00	0.00
	2	-2.78	-0.95	0.00	0.00
	3	-1.73	-0.57	0.00	0.00
JC on TT+TC	1	-0.32	-4.15	-55.63	-61.60
	2	-0.80	-3.43	-55.67	-62.29
	3	-3.57	-4.35	-58.92	-62.29
JC on MT	1	-1.89	-0.56	0.00	0.00
	2	-2.01	-0.59	0.00	0.00
	3	-1.66	-0.48	-0.03	0.00
JC on MT+TC	1	6.21	2.89	33.99	2.01
	2	0.89	0.37	4.14	0.00
	3	0.18	0.09	1.70	0.00
Epinions					
TC on TT	1	11.28	18.72	577.05	105.47
	2	3.18	1.94	25.21	4.45
	3	0.19	0.10	1.36	0.96
TC on MT	1	10.25	18.03	576.96	105.46
	2	2.96	1.78	25.14	4.37
	3	0.18	0.09	1.29	0.88
JC on TT	1	0.49	-2.29	-16.66	-23.56
	2	-1.48	-1.62	-27.64	-28.41
	3	-2.00	-0.79	0.69	0.91
JC on TT+TC	1	-2.37	-1.09	-3.75	-0.86
	2	-5.85	-2.54	-2.87	-0.49
	3	-2.20	-0.89	-0.66	-0.05
JC on MT	1	0.55	-2.45	-17.28	-24.62
	2	-1.10	-1.74	-33.17	-32.63
	3	-5.85	-3.76	-50.28	-34.96
JC on MT+TC	1	-2.09	-0.93	-3.71	-0.86
	2	-5.14	-2.13	-2.84	-0.49
	3	-1.91	-0.74	-0.66	-0.05

Table 7-13 Performance of all methods for **grey-sheep users** on the two datasets

Algorithm	RMSE	MAE	RC	UC	FMeasure	RMSE	MAE	RC	UC	FMeasure
	Grey-sheep users									
	Filmtrust					**Epinions**				
TT1	1.515	1.228	19.94	34.41	1.205	1.852	1.420	6.03	33.42	0.986
TT2	1.497	1.243	24.82	34.41	1.221	1.683	1.326	37.18	62.93	1.141
TT3	1.515	1.269	26.13	34.41	1.214	1.613	1.296	57.29	67.17	1.181
TT1+TC	1.453	1.195	25.96	34.41	1.243	1.652	1.306	42.11	64.43	1.158
TT2+TC	1.470	1.224	26.58	34.41	1.236	1.649	1.306	44.72	65.67	1.160
TT3+TC	1.502	1.262	27.31	34.41	1.221	1.612	1.294	57.89	68.14	1.182
TT1+JC	1.517	1.247	2.80	6.45	1.016	1.864	1.439	4.66	25.34	0.958
TT2+JC	1.463	1.228	4.36	6.45	1.107	1.700	1.332	26.56	45.99	1.126
TT3+JC	1.460	1.226	4.42	6.45	1.111	1.629	1.303	57.73	68.08	1.173
TT1+TC+JC	1.476	1.212	25.96	34.41	1.232	1.675	1.315	41.07	63.84	1.146
TT2+TC+JC	1.509	1.257	26.58	34.41	1.217	1.714	1.335	43.93	65.34	1.128
TT3+TC+JC	1.521	1.278	27.31	34.41	1.212	1.629	1.303	57.73	68.08	1.173
MT1	1.200	0.956	19.94	34.41	1.353	1.755	1.404	6.03	33.42	1.027
MT2	1.193	0.970	24.82	34.41	1.365	1.607	1.309	37.28	63.06	1.178
MT3	1.188	0.977	26.13	34.41	1.369	1.545	1.282	57.45	67.36	1.214
MT1+TC	1.179	0.953	25.96	34.41	1.373	1.577	1.290	42.11	64.43	1.194
MT2+TC	1.175	0.956	26.58	34.41	1.376	1.576	1.291	44.72	65.67	1.196
MT3+TC	1.182	0.968	27.31	34.41	1.374	1.544	1.280	57.88	68.14	1.215
MT1+JC	1.251	1.010	2.80	6.45	1.104	1.770	1.425	4.61	25.08	0.995
MT2+JC	1.247	1.036	4.08	6.45	1.178	1.611	1.305	24.24	43.58	1.166
MT3+JC	1.246	1.044	4.15	6.45	1.181	1.591	1.296	30.52	45.15	1.181
MT1+TC+JC	1.185	0.958	25.96	34.41	1.370	1.597	1.297	41.07	63.84	1.184
MT2+TC+JC	1.190	0.967	26.58	34.41	1.369	1.633	1.321	43.95	65.34	1.168
MT3+TC+JC	1.193	0.977	27.31	34.41	1.368	1.558	1.287	57.73	68.08	1.208

Table 7-14 Impact of TC and JC on all methods for **grey-sheep users**

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	4.05	3.13	30.16	0.00
	2	1.84	1.24	7.10	0.00
	3	0.87	0.62	4.50	0.00
TC on MT	1	1.70	1.51	30.16	0.00
	2	1.48	0.80	7.10	0.00
	3	0.50	0.32	4.50	0.00
JC on TT	1	-0.16	-15.68	-85.96	-81.25
	2	2.29	-9.28	-82.45	-81.25
	3	3.60	-8.51	-83.07	-81.25
JC on TT+TC	1	-1.52	-0.85	0.00	0.00
	2	-2.67	-1.51	0.00	0.00
	3	-1.29	-0.76	0.00	0.00
JC on MT	1	-4.29	-18.41	-85.96	-81.25
	2	-4.48	-13.70	-83.57	-81.25
	3	-4.88	-13.74	-84.13	-81.25
JC on MT+TC	1	-0.45	-0.18	0.00	0.00
	2	-1.25	-0.51	0.00	0.00
	3	-0.92	-0.38	0.00	0.00
Epinions					
TC on TT	1	10.80	17.40	598.12	92.79
	2	2.01	1.70	20.26	4.35
	3	0.09	0.07	1.04	1.45
TC on MT	1	10.16	16.31	598.12	92.79
	2	1.89	1.52	19.93	4.13
	3	0.08	0.06	0.76	1.16
JC on TT	1	-0.67	-2.86	-22.74	-24.17
	2	-1.04	-1.35	-28.57	-26.92
	3	-0.96	-0.64	0.78	1.36
JC on TT+TC	1	-1.44	-1.03	-2.47	-0.91
	2	-3.98	-2.78	-1.76	-0.50
	3	-1.06	-0.71	-0.26	-0.10
JC on MT	1	-0.84	-3.13	-23.52	-24.95
	2	-0.28	-1.03	-34.99	-30.89
	3	-2.94	-2.72	-46.87	-32.98
JC on MT+TC	1	-1.28	-0.86	-2.47	-0.91
	2	-3.59	-2.33	-1.71	-0.50
	3	-0.88	-0.55	-0.26	-0.10

Table 7-15 Performance of all methods for **controversial items** on the two datasets

Algorithm	RMSE	MAE	RC	UC	FMeasure	RMSE	MAE	RC	UC	FMeasure
	Controversial items									
	Filmtrust					**Epinions**				
TT1	2.725	2.350	13.70	15.00	0.623	2.084	1.667	6.25	10.01	0.890
TT2	2.816	2.429	19.18	20.00	0.583	2.016	1.655	31.92	39.84	0.977
TT3	2.607	2.176	23.29	23.33	0.686	2.068	1.737	55.16	59.52	0.958
TT1+TC	2.922	2.615	17.81	20.00	0.531	2.036	1.685	37.35	44.16	0.969
TT2+TC	2.816	2.429	19.18	20.00	0.583	2.045	1.697	39.92	46.56	0.966
TT3+TC	2.607	2.176	23.29	23.33	0.686	2.072	1.742	56.01	60.25	0.956
TT1+JC	3.246	3.214	9.59	10.00	0.370	2.059	1.652	5.06	7.97	0.886
TT2+JC	3.304	3.278	12.33	13.33	0.343	2.033	1.657	22.77	28.54	0.963
TT3+JC	3.304	3.278	12.33	13.33	0.343	2.106	1.754	55.62	60.05	0.939
TT1+TC+JC	2.922	2.615	17.81	20.00	0.531	2.064	1.694	35.92	43.01	0.955
TT2+TC+JC	2.816	2.429	19.18	20.00	0.583	2.128	1.738	38.75	45.68	0.925
TT3+TC+JC	2.567	2.103	23.29	23.33	0.705	2.106	1.754	55.62	60.05	0.939
MT1	2.238	1.938	13.70	15.00	0.854	2.006	1.619	6.25	10.01	0.923
MT2	2.458	2.169	19.18	20.00	0.756	1.931	1.593	31.93	39.87	1.018
MT3	2.306	1.980	23.29	23.33	0.832	1.953	1.638	55.20	59.58	1.014
MT1+TC	2.536	2.261	17.81	20.00	0.717	1.935	1.603	37.35	44.16	1.018
MT2+TC	2.458	2.169	19.18	20.00	0.756	1.943	1.615	39.92	46.56	1.015
MT3+TC	2.306	1.980	23.29	23.33	0.832	1.956	1.641	56.01	60.25	1.013
MT1+JC	2.667	2.637	9.59	10.00	0.644	1.986	1.608	4.98	7.83	0.915
MT2+JC	2.816	2.779	12.33	13.33	0.578	1.936	1.584	20.98	26.22	1.007
MT3+JC	2.816	2.779	12.33	13.33	0.578	1.959	1.614	26.85	31.53	1.001
MT1+TC+JC	2.536	2.261	17.81	20.00	0.717	1.963	1.614	35.94	43.01	1.004
MT2+TC+JC	2.458	2.169	19.18	20.00	0.756	2.020	1.657	38.77	45.69	0.977
MT3+TC+JC	2.260	1.920	23.29	23.33	0.854	1.986	1.653	55.62	60.05	0.998

Table 7-16 Impact of TC and JC on all methods for **controversial items**

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	-7.24	-14.79	30.00	33.33
	2	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00
TC on MT	1	-13.32	-15.98	30.00	33.33
	2	0.00	0.00	0.00	0.00
	3	0.00	0.00	0.00	0.00
JC on TT	1	-19.12	-40.65	-30.00	-33.33
	2	-17.34	-41.15	-35.71	-33.33
	3	-26.76	-50.01	-47.06	-42.86
JC on TT+TC	1	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00
	3	1.50	2.76	0.00	0.00
JC on MT	1	-19.18	-24.56	-30.00	-33.33
	2	-14.57	-23.51	-35.71	-33.33
	3	-22.14	-30.52	-47.06	-42.86
JC on MT+TC	1	0.00	0.00	0.00	0.00
	2	0.00	0.00	0.00	0.00
	3	1.97	2.63	0.00	0.00
Epinions					
TC on TT	1	2.28	8.91	497.98	341.32
	2	-1.46	-1.16	25.07	16.87
	3	-0.19	-0.19	1.54	1.22
TC on MT	1	3.53	10.30	497.98	341.32
	2	-0.63	-0.26	25.01	16.79
	3	-0.14	-0.12	1.48	1.12
JC on TT	1	1.19	-0.48	-18.95	-20.38
	2	-0.86	-1.46	-28.65	-28.38
	3	-1.85	-1.95	0.82	0.89
JC on TT+TC	1	-1.37	-1.45	-3.83	-2.61
	2	-4.04	-4.21	-2.93	-1.91
	3	-1.65	-1.77	-0.71	-0.33
JC on MT	1	1.01	-0.94	-20.24	-21.79
	2	-0.26	-1.06	-34.29	-34.23
	3	-0.31	-1.24	-51.35	-47.08
JC on MT+TC	1	-1.43	-1.38	-3.76	-2.59
	2	-3.97	-3.74	-2.87	-1.89
	3	-1.54	-1.46	-0.71	-0.33

Table 7-17 Performance of all methods for **niche items** on the two datasets

Algorithm	RMSE	MAE	RC	UC	FMeasure	RMSE	MAE	RC	UC	FMeasure
	Niche items									
	Filmtrust					**Epinions**				
TT1	1.366	1.031	14.04	19.63	1.258	1.108	0.722	0.53	2.30	0.612
TT2	1.342	1.011	19.35	26.44	1.285	1.250	0.850	4.17	19.92	1.181
TT3	1.278	0.962	25.36	31.94	1.325	1.263	0.870	14.93	47.37	1.308
TT1+TC	1.331	1.002	20.11	27.49	1.292	1.256	0.859	6.46	26.41	1.240
TT2+TC	1.323	0.994	20.52	28.53	1.296	1.259	0.861	7.21	28.75	1.252
TT3+TC	1.270	0.954	25.96	32.72	1.330	1.263	0.870	15.64	48.25	1.311
TT1+JC	1.357	1.040	10.44	12.57	1.243	1.086	0.706	0.46	1.94	0.562
TT2+JC	1.374	1.048	13.76	16.49	1.253	1.231	0.833	2.89	13.67	1.117
TT3+JC	1.364	1.040	14.14	17.02	1.259	1.280	0.878	15.14	47.81	1.302
TT1+TC+JC	1.339	1.006	20.11	27.49	1.288	1.260	0.860	5.83	24.90	1.226
TT2+TC+JC	1.331	0.998	20.52	28.53	1.293	1.266	0.863	6.60	27.40	1.239
TT3+TC+JC	1.274	0.954	25.96	32.72	1.328	1.280	0.878	15.14	47.81	1.302
MT1	1.189	0.908	14.01	19.37	1.338	1.033	0.728	0.53	2.30	0.618
MT2	1.214	0.920	19.23	25.39	1.344	1.182	0.853	4.17	19.94	1.206
MT3	1.174	0.895	25.21	30.63	1.374	1.187	0.857	14.94	47.43	1.343
MT1+TC	1.206	0.913	20.05	26.96	1.350	1.182	0.854	6.45	26.41	1.270
MT2+TC	1.198	0.907	20.40	27.49	1.354	1.185	0.856	7.21	28.75	1.283
MT3+TC	1.172	0.891	25.81	31.41	1.376	1.187	0.857	15.64	48.25	1.346
MT1+JC	1.122	0.870	10.44	12.57	1.346	1.016	0.720	0.46	1.93	0.567
MT2+JC	1.191	0.917	13.63	15.97	1.336	1.164	0.839	2.72	12.58	1.125
MT3+JC	1.185	0.912	14.04	16.75	1.340	1.173	0.847	4.46	18.26	1.220
MT1+TC+JC	1.214	0.919	20.05	26.96	1.346	1.186	0.856	5.84	24.90	1.256
MT2+TC+JC	1.207	0.912	20.40	27.49	1.350	1.190	0.859	6.62	27.40	1.270
MT3+TC+JC	1.176	0.894	25.81	31.41	1.375	1.201	0.866	15.14	47.81	1.337

Table 7-18 Impact of TC and JC on all methods for **niche items**

Filmtrust		% improvement			
Algorithm combination	**Propagation**	**RMSE**	**FMeasure**	**RC**	**UC**
TC on TT	1	2.55	2.67	43.24	40.00
	2	1.41	0.88	6.05	7.92
	3	0.63	0.35	2.37	2.46
TC on MT	1	-1.36	0.89	43.12	39.19
	2	1.25	0.73	6.09	8.25
	3	0.19	0.14	2.38	2.56
JC on TT	1	0.65	-1.21	-25.68	-36.00
	2	-2.37	-2.46	-28.92	-37.62
	3	-6.73	-4.99	-44.26	-46.72
JC on TT+TC	1	-0.57	-0.27	0.00	0.00
	2	-0.60	-0.29	0.00	0.00
	3	-0.28	-0.13	0.00	0.00
JC on MT	1	5.65	0.59	-25.51	-35.14
	2	1.86	-0.66	-29.11	-37.11
	3	-0.90	-2.47	-44.29	-45.30
JC on MT+TC	1	-0.66	-0.27	0.00	0.00
	2	-0.71	-0.29	0.00	0.00
	3	-0.31	-0.12	0.00	0.00
Epinions					
TC on TT	1	-13.40	102.73	1117.89	1047.39
	2	-0.72	6.02	72.84	44.37
	3	0.02	0.21	4.71	1.87
TC on MT	1	-14.44	105.42	1116.27	1047.39
	2	-0.24	6.38	72.79	44.19
	3	0.00	0.20	4.66	1.73
JC on TT	1	1.97	-8.08	-13.63	-15.65
	2	1.50	-5.41	-30.83	-31.36
	3	-1.30	-0.51	1.37	0.94
JC on TT+TC	1	-0.33	-1.16	-9.69	-5.74
	2	-0.53	-1.01	-8.46	-4.72
	3	-1.32	-0.72	-3.19	-0.91
JC on MT	1	1.68	-8.36	-13.84	-16.09
	2	1.49	-6.67	-34.72	-36.94
	3	1.23	-9.16	-70.17	-61.50
JC on MT+TC	1	-0.30	-1.12	-9.39	-5.74
	2	-0.49	-0.98	-8.23	-4.72
	3	-1.18	-0.62	-3.19	-0.91

7.3 Summary and conclusion

This chapter presented and analysed the results of the experimental evaluation conducted for the proposed methods (TriadicClosure and JaccardCoefficient). Initially, the TriadicClosure algorithm was compared with basic trust-based approaches and then it was incorporated into these approaches to measure their performance. Then, the JaccardCoefficient method was compared against all the above methods and, finally, the performance of both the proposed methods was evaluated for different views of datasets.

The results of all these experiments indicate that, in large and sparse datasets, the positive impact of TriadicClosure in coverage is impressive, reaching an improvement of 578% in RC, while the positive impact of the TriadicClosure on accuracy is greater than the impact of incorporating the Collaborative Filtering algorithm. Generally, the TriadicClosure algorithm outperforms the other methods when propagation level is 1, while, combined with the other methods (MoleTrust and TidalTrust), for all propagation lengths it produces better results compared with the results of each original method. Moreover, the results show that the integration of TriadicClosure with other state-of-the-art trust-based methods can improve their performance, while the percentage improvement tends to zero when propagation length is greater than 3.

Concluding, the TriadicClosure algorithm shows improvement on the performance of all views, compared to the original methods of TidalTrust and MoleTrust. However, the percentage improvement of the performance of the combined methods over the performance of the original methods is decreasing as the propagation level increases. Moreover, as the propagation level increases, it is computationally more expensive, especially for large trust networks. This means that the TriadicClosure as stand-alone or combined with the MoleTrust with propagation level 2 are the methods giving the best performances with the lower computational expense.

Regarding the JaccardCoefficient method for calculating the weight of the inferred trust relationships, the results, although not stable, in the majority showed that its impact on the performance is negative. A possible explanation for these results is that the weights of the trust networks of the datasets used in the experiments are binary.

Chapter 8

Discussion, future work and conclusions

8.1 Discussion

To deal with the information overload problem, recommender systems adopt various techniques to filter information and produce suggestions for users. However, current approaches face limitations such as 'lack of transparency', 'grey-sheep', 'synonymy and polysemy' and 'security and privacy'. With the advent of social networks, a new approach has been born, the so-called trust-based recommender systems, which exploit the relationships between users built in the social networks to produce more trustworthy recommendations than these from unknown users. Research on trust-based recommender systems, as discussed in Chapter 4, has demonstrated (Victor, Cornelis and DeCock, 2011) that the use of trust can significantly improve both the coverage and the accuracy of recommendations, especially with sparse datasets. Moreover, trust can significantly improve recommendations accuracy when item ratings are more extreme and show disagreement between users. However, the analysis of the major methods to infer trust in Chapter 5 revealed their limitations. More specifically, although the accuracy of the recommended ratings with trust-based approaches, such as TidalTrust (Golbeck, 2005) and MoleTrust (Massa and Avesani, 2005), has been proved to outperform some baseline recommender system algorithms, it is, however, strongly affected by the density of the trust network. Especially for users with no trusted neighbours or even with not at least moderately trusted neighbours, it is impossible to find any trust path with the specific algorithms.

Consequently, data sparsity is the major limitation characterising all the current systems, which affects not only the item-ratings matrix, but also the trust-ratings matrix of the trust-based systems. Furthermore, the examination and comparison of the main literature in graph-based models showed that current approaches propagate trust, based purely on the direct propagation strategy of the atomic propagation. However, the homophily phenomenon, on which propagation is based, is not fully addressed by the direct propagation. This revealed the need to fully exploit the propagative property of trust, as defined by the homophily phenomenon, by considering not only 'the friend of my friend' to propagate trust, but also the 'common friends' intuition. Thus, a novel method was proposed to infer trust, filling this gap.

More specifically, homophily is a phenomenon affecting social relationships at various levels, stating that people tend to associate with those having something in common. Based on this phenomenon, triadic closure is a fundamental mechanism of link formation in social networks, reaching about 60% of all the new connections formed in a social network (Kossinets and Watts, 2009). Additionally, Bianconi *et al*. (2014) found that communities emerge naturally via triadic closure, especially when the network is very sparse.

This mechanism is the basis for the novel method, called triadic closure, proposed in this study, to infer trust relationships in trust networks. In addition, a novel method to calculate the trust weight of a trust relationship is proposed based on the Jaccard Coefficient similarity metric exploited in the trust network. Both methods and their algorithms were thoroughly described in Chapter 6 along with the way to incorporate them in the recommendation process. Initially, both methods were evaluated with synthetic data to prove their validity and, then, a thorough experimental evaluation was conducted with real-world datasets in order to compare their performance with other state-of-the-art methods.

The results of the experiments showed that triadic closure outperforms other state-of-the-art trust-based methods with a substantial improvement on coverage. Incorporated in other methods, the triadic closure algorithm showed improvement on the performance, compared to the original methods. However, the percentage improvement was decreasing as the propagation level was increasing while the computational expense was increasing, especially for large trust networks. Thus, it is concluded that triadic closure, either as stand-alone or combined with the MoleTrust with propagation level 2, gives the best performances with lower computational expense. The experiments for different views of the real-world datasets indicated that triadic closure also produced particularly good results regarding accuracy for heavy-raters and grey-sheep users, but the best improvement in performance was achieved for cold-start users and items.

From the results of the experimental evaluation, it is apparent that the triadic closure method addresses various limitations of typical recommendation methods. More specifically the triadic closure method outperformed other well-known methods in regards to the cold-start problem, either for users or for items. Moreover, in grey-sheep users, the algorithm performed well, since it improved the accuracy and coverage compared to the other methods. But, most of all, the method contributed to the alleviation of data sparsity by substantially improving ratings and user coverage since the method expands the existing trust network, filling the sparse trust matrix with values.

The fact that the best percentage improvement for the triadic closure algorithm combined with direct propagation is achieved when the propagation level is 1 and 2 can be explained as follows: triadic closure is applied on explicit trust statements and, thus, triads close only for existing trust links and not for inferred. In other words, the triadic closure algorithm complements the inference of direct propagation and cannot be applied on the inferred relations of level 2. After the initial inference of the trust links with the triadic closure, no more links can be inferred by the algorithm. Then, the percentage improvement for different levels depends totally on the combined method. Since it is becoming too time-consuming and, thus, computationally too expensive, to calculate the inferred relations for propagation more than 2, while there is no significant improvement in the performance, it is concluded that the triadic closure algorithm combined with a direct propagation of level 2 can provide the best performance in terms of accuracy and coverage.

The triadic closure method actually is a novel trust metric to infer trust relationships based on a fundamental mechanism of link formation within social networks. Recently, Gao *et al.* (2016) introduced a novel trust metric based on collective semiring methods to address the sparsity of trust connections. However, the method was not experimentally compared with other state-of-the-art trust-based methods.

Several other trust-based approaches exist in the recent literature that model trust for recommendations, such as that of Mei *et al.* (2017) who built a trustee-influence based trust model wherein a trustee's activeness or trustworthiness is used to determine trust relationships. Moreover, Liu, Cao and Yu (2016) supported that users are influenced by public opinion and modelled the conformity phenomenon in online rating sites using three influence factors in order to understand the user behaviours and to improve the rating prediction accuracy. This method could prove valuable incorporated in the triadic closure method for combining local and global trust.

Recently, Rafailidis and Crestani, (2017) introduced a weighting strategy to balance the influences of 'friends and foes' selections by building two intermediate trust/distrust-preference user latent spaces to capture the correlations of users' preferences with friends' trust and foes' distrust degrees, accordingly. Previously, Rafailidis, (2016) modelled the trust and distrust relationships into signed graphs, and then generated clusters to incorporate them into a matrix factorisation framework. The method produced lower prediction errors for different levels of sparsity compared with other matrix factorisation methods. However, it did not deal with the sparsity of the trust-rating matrix.

Azadjalal, Moradi and Abdollahpouri, (2014) used the Pareto dominance concept to identify trust value between trusted users and active user. However, their experimental results showed that coverage was not increased compared with the other methods. Similarly, Ma, King and Lyu (2011) focused mainly on accuracy of recommendations and did not take into account coverage, either for ratings or for users.

Consider also that machine learning models, although popular within the research community, lack interpretability and do not provide enough explanations for their recommendations. However, explanations have been shown to improve the transparency of a recommender system by justifying recommendations, and this, in turn, can enhance the user's trust in the recommendations (Abdollahi, 2017). However, trust-based approaches can provide transparency, especially when the inferred relationships are resulting from the explicit personal network and, thus, the triadic closure method can contribute to explanations.

As a trust metric, triadic closure can also be combined with other methods to improve recommendation accuracy. Combination with content-based approaches could provide satisfactory results or combined with co-occurrence algorithms could give a possible solution to the context-dependency of trust.

Besides, it is very easy to adopt and incorporate the triadic closure algorithm into existing trust-based systems. The algorithm is based just on existing trust networks and can run offline in order not to affect the time calculation of the recommendation. So, this could improve the performance of a trust-based recommender system and, finally, the effectiveness of the system.

Finally, the triadic closure method not only outperforms the classic trust metrics, but also, compared to other recent methods, provides proof that it can contribute to the sparsity problem of user-item and user-user ratings. Considering also that the improvement of accuracy in recommendations is due to the prediction of user connections that would connect in the majority anyway, based on the triadic closure mechanism, the method can be proved as durable, as the links are formed based on this mechanism.

Regarding the Jaccard Coefficient, the results of the experiments were not stable and mainly showed that its impact on the performance was negative. A possible explanation for these results is that the weights of the trust networks of the datasets used in the experiments are binary. Thus, the comparison of the gradual weights produced by the Jaccard Coefficient method, which are values lower than 1, were always compared to the values of the datasets, which are always 1s. In other words, the produced values were always lower than the compared original values, which may lead to misleading results regarding accuracy. Similarly to the current study, Guo, Zhang and Thalmann (2014) proposed an approach to solve the cold-start and sparsity problems. Specifically, ratings of a user’s trusted neighbours were merged to complement and represent the preferences of the user and to find other users with similar preferences. Furthermore, a confidence-aware similarity measure between users was introduced, which included a parameter very similar to the Jaccard Coefficient method. Thus, combining this confidence-aware similarity measure with the triadic closure method may alleviate the limitation of the Jaccard Coefficient and produce more stable results.

Considering the experimental evaluation of Jaccard Coefficient that didn't manage to prove any improvement on accuracy due to the lack of gradual trust weights in the datasets, it revealed a strong limitation of current evaluation methodologies for the trust-based recommender systems. The majority of research studies dealing with trust metrics for recommender systems evaluate their approaches based on two datasets (Filmtrust and Epinions), since these two datasets contain both user-item and user-user ratings. Although, initially, the Filmtrust dataset was released with gradual trust weights, currently, this dataset is not available, but only with bivalent trust values. Since this limitation affects the evaluation of methods, like the Jaccard Coefficient, it is apparent that there is a need for producing real-world datasets with gradual ratings for both user-item and user-user matrices. Moreover, the Filmtrust dataset was crawled from an experimental platform for the needs of a project (Golbeck, 2006b) and, for this reason, the cold-start users and items are minimal and, additionally, the ratings may include a bias due to real-world social connection of the users participated in the project.

Another serious limitation that current trust-based methods face is that of privacy preservation. Several studies exist for preserving privacy, based on ratings perturbation, also called randomisation (Polat and Du, 2003; Polatidis *et al.*, 2017), anonymisation methods (Casino *et al.*, 2015; Zigomitros, Papageorgiou and Patsakis, 2016) and cryptographic methods (Backes *et al.*, 2010). The balance between transparency, security and privacy preservation when expanding trust networks to increase accuracy still remains a challenge.

Returning to the evaluation method used in this study, one alternative would be to use live user experiments for validation instead of offline. However, as already stated, the offline method is a less expensive and not time-consuming with reproducible results and the chance to compare with the same data and same conditions other methods. Moreover, with offline experiments, the algorithms could be tested for their performance on large datasets. Although live-experiments could evaluate user performance, satisfaction, participation and the usability of the interface, this approach has a number of limitations, such as small portion of users, not necessarily representative sample, it is expensive and may produce misleading results from biased actions.

Finally, the aim of this study, as initially set in Section 1.2, was achieved while all the research questions set in Section 1.5 have been addressed:

RQ1. How can the accuracy of recommender systems be improved? Is it possible to utilise trust data to improve accuracy?

The literature review of Chapter 3 revealed that the ultimate goal of almost every new approach in the research area of recommender systems is to improve the accuracy of recommendations. Moreover, the literature review of Chapter 4 proved that the utilisation of trust statements expressed between users within a social network can significantly improve the accuracy of recommendations. Additionally, the experiments of Section 7.2 proved that the triadic closure method, which utilises existing trust statements between users, can significantly improve recommendation accuracy.

RQ2. How can the sparsity in the item and trust ratings matrices be dealt with ?

One of the limitations that face typical recommender systems, as discussed in Section 5.1, is the sparsity of the item-ratings matrix. This can be alleviated with the adoption of trust statements between users, as described in Section 4.4. However, trust-based recommender systems also suffer from sparsity of the trust-ratings matrix. An approach to deal with this problem is to expand a trust network by inferring new trust relationships. Approaches that deal with this problem are the trust metrics, which are thoroughly described in Sections 4.4 and 4.2. Moreover, the triadic closure method deals with the sparsity of both matrices, since the experiments of Section 7.2 showed that the method can substantially improve the coverage for both item and trust ratings.

RQ3. How can we expand an existing trust network?

Based on the computational properties of trust described in Section 4.2 and the small-world phenomenon in Section 4.3.1, trust can be inferred in order to expand an existing trust network. Various methods exist that expand existing trust networks, as described in Sections 4.4 and 4.2. A novel method also is proposed, called the triadic closure, in Section 6.3 based on the homophily phenomenon and, specifically, on the triadic closure mechanism governing the social networks.

RQ4. Is there any new way of utilising existing trust data to expand the trust network? Can we predict the new connections from knowledge of the existing trust network?

Since trust between two users can be seen as a link between them in a social network, trust prediction can be seen as a link-prediction problem, as described in Section 4.3.2. Additionally, propagation is a popular method to infer trust from existing trust data. As described in Section 4.3.3, taking into account existing trust statements, new relationships can be inferred through the propagating property of trust. However, the homophily phenomenon on which propagation is based is not fully exploited by the existing state-of-the-art methods. For this reason, a novel method, the triadic closure, was proposed in Chapter 6 that utilises the knowledge of the existing trust data to expand the trust network based on the triadic closure mechanism for links. The novel method exploits the triadic closure mechanism that governs the link formation in a social network on the existing trust relationships in order to predict new trust relationships.

RQ5. Can we predict topical similarity from the trust network?

Taking advantage of the trust statements in an existing trust network, a novel method was proposed in Section 6.4, the Jaccard Coefficient, to predict the weight of trust relationships. Based on the Jaccard Coefficient similarity measure, the method predicts topical similarity, taking into account structural information of the trust network. The experimental evaluation of the method in Section 7.2 with real-world datasets didn't produce stable results, while, in the majority, the impact on the performance in terms of accuracy and coverage was negative. Although further investigation is needed with real-world datasets with non-binary trust statements, the answer for this question that emerged from the experiments is that in fact we cannot predict topical similarity from the structure of a trust network.

RQ6. What is the impact of expanding a trust network in the accuracy of recommender systems?

In Section 4.2, various state-of-the-art methods were presented that expand the trust network through propagation, in order to improve the accuracy of recommender systems. Moreover, the experimental evaluation (Section 7.2) of the triadic closure and the other state-of-the-art methods proved that not only the accuracy of recommender systems, but also the ratings coverage, can substantially be improved by inferring new trust relationships in an existing trust network.

8.2 Future work

The current study can be proved valuable to researchers and practitioners since it provides the procedures to be applicable in the field of bibliographic recommendations. Recently, Ciotti *et al.* (2016) found that homophily governs the creation of links in citation networks. Citation networks represent the relationships between researchers or papers constructed by the citations of academic papers. Citation networks have been widely used to study the evolution of science and investigate the dynamics regarding knowledge flows and sharing between papers and authors. Applying the triadic closure method, proposed in this study, can contribute to the scientific community for recommending academic papers to researchers closely related to their interests. Except the old papers missed by the researcher, the system also can recommend new papers currently published, which otherwise could not be found easily, since are not yet cited and do not have an impact factor on which current systems are based.

Moreover, scientific communities can also be constructed though triadic closure based on the co-citation and co-authorship data and, in conjunction with content analysis of the publications, can extract scientific trends and serendipity in recommendations. Thus, a possible extension of the proposed method would be to combine it with content-based algorithms. Furthermore, the combination with an algorithm finding co-occurrences would follow the recent trend implemented by Google, which identifies the similarity between two pages not only by the links, but also by frequency of occurrence and close proximity of similar keywords existing in the two pages. Co-occurrence algorithms also can be used to deal with the context-dependency property of trust. Context is a concept usually not considered in trust-based recommender systems, either for simplicity reasons (Golbeck, 2005) or because the system is context specific. However, as discussed in Section 4.2, trust is context-dependent and should be considered in the recommendation process. Consequently, a future direction would be to consider context within the triadic closure algorithm so as to produce context-oriented trust links.

Distrust is another concept that should also be considered and examined and which is usually examined separately (Lewicki, Mcallister and Bies, 1998; Guha *et al.*, 2004; Ziegler and Lausen, 2005; Victor, Cornelis and DeCock, 2011; Fang, Guo and Zhang, 2015; Tang, 2015) since its properties differ considerably from the properties of trust. Accordingly, as a first step, there is a need to extensively research the prediction of distrust links in order to examine the impact of the triadic closure in distrust prediction.

Additionally, as already mentioned, research regarding the aggregation process in trust inference is still in its infancy (Victor, Cornelis and DeCock, 2011) needing to give greater attention to new aggregation operators combined with propagation, but also to operators preserving privacy.

Finally, future work should also be done regarding the Jaccard Coefficient method. More specifically, experiments on datasets with gradual trust weights should be conducted in order to further examine the behaviour of the algorithm on recommender systems with gradual trust weights.

8.3 Conclusions

Generally, recommender systems as services can be found from a simple search in a search engine to a product search in an e-commerce platform or music platform, etc., in order to deal with the information overload problem. The initial idea for this study was originated from the need, as an Internet user, to find and be recommended more relevant items, such as products, research papers, bibliography, news and other general information. Thus, accuracy in recommendations is important to researchers, but also to Internet users as customers, for reducing the time of searching and gaining more relevant recommendations. Moreover, improving recommendation accuracy is valuable also to e-commerce service providers for increasing their revenue.

The aim of this study was to improve the accuracy of trust-based recommender systems through the inference of new connections between users, based on existing relationships within a trust network. The main idea was to utilise the available information given by the trust network in order to expand it, by inferring new trust connections and, finally, improve recommendation accuracy.

Propagation is one of the main methods for inferring new trust relationships, since one of the main challenges in trust-based approaches is to expand the personal trust network of a user by inferring new trust relationships. Current approaches propagate trust based on the intuition that "*the friend of my friend is also my friend*". However, the homophily phenomenon defines that similarity breeds connection. This means that persons having a common friend tend also to become friends. In social networks, this mechanism, also known as 'triadic closure', is a fundamental mechanism of link formation governing the majority of new connections formed. However, this mechanism was not considered to any of the existing methods, leading to the inspiration that, if triadic closure was considered to infer trust connections, this could lead to improved accuracy of recommendations.

Thus, based on the homophily phenomenon of social networks, and specifically on the triadic closure mechanism of link formation in social networks, a novel method was proposed to overcome the sparsity problem of the trust networks. Also, another method for calculating the trust value of an inferred trust relationship is proposed based on the Jaccard Coefficient similarity measure. Both the methods proposed exploit structural information of the trust graph to infer new trust relationships and calculate their trust value. Finally, through these two methods, trust-based recommendations were produced, which were compared to other existing state-of the-art methods. The results of the evaluation showed that the main method for inferring trust relationships surpasses the other methods in terms of accuracy and coverage.

All the objectives of this study were met, since it initially performed a review on the literature of existing methods, and techniques on recommender systems reviewed, along with a review, identification and evaluation of the state-of-the-art trust-based approaches. A new method, the triadic closure, was developed for expanding existing trust networks along with a new method, the Jaccard Coefficient, to calculate trust value of the inferred trust relationships. Then, the influence on the accuracy of recommendations was investigated for both the proposed methods as stand-alone methods, but also integrated into other state-of-the-art methods. Finally, both methods were evaluated and compared with existing state-of-the-art approaches and the results were presented and analysed discussing the learning outcome suggesting future work.

The main contribution of the current study is the use of a novel trust inference technique that increases the range of the existing neighbours of a user and improves recommendation accuracy, since it performs better than existing standard trust inference techniques. In fact, it is a novel trust metric. The novelty of the proposed approach is the way that relationships are handled, inspired by real-life scenarios and, especially, the triadic closure mechanism of social networks. Additionally, this study provided evidence that the proposed model substantially improves prediction accuracy and coverage with respect to previous methods.

The proposed method can be applied in many areas, from e-commerce platforms to bibliographic recommenders, but also into search engines. The triadic closure method can inspire researchers for extending it or finding new methods incorporating this one. In practice, the proposed method can improve bibliographic recommendations and also infer connections in collaboration networks. Consider also that Google currently uses co-occurrences in its algorithm to find the similarity between two pages by calculating the frequency of occurrence, and close proximity of similar keywords existing in the pages. Combining the triadic closure and co-occurrence algorithms can deal with the context-dependency property of trust.

Concluding, as the online social networks continue to exponentially grow, the exploitation of the properties and structure characteristics of social networks can inspire novel methods to deal with the information overload problem and produce recommendations closer to the user's needs.

Appendix A

Java code

```
public class TriadicClosure
{

   /**
    * TriadicClosure algorithm to infer trust
    *
    *
    * @param userTrusteesMap
    *
    * @param source
    *            source user
    * @param maxDepth
    *            maximum length of the searching path
    * @return
    */


   public static <U> Map<U, Double> runAlgorithm(Map<U, Set<U>>
userTrusteesMap,
        String source, int maxDepth, boolean jcFlag)
   {
     Map<U, Double> trustScores = new HashMap<>();

      /**
       * Triadic Closure: adds the connection between two nodes with common
neighbors
       */

         for (U user : userTrusteesMap.keySet()){
           if ( ! user.equals(source) ){

              // get the map of source
              Set<U> sourceMap = userTrusteesMap.get(source) != null ?
                     userTrusteesMap.get(source) : null;

              //get the map of users
              Set<U> userMap = userTrusteesMap.get(user) != null ?
                     userTrusteesMap.get(user) : null;

              if (userMap == null || sourceMap == null ) continue;

              Set<U> intersectionMap = findIntersection(userMap, sourceMap);
              Set<U> unionMap = findUnion(userMap, sourceMap);

              //calculate the Jaccard Coefficient
              if ( jcFlag ) {
                 Double jc  = getJaccardCoefficient(intersectionMap,
unionMap);

              if (intersectionMap.size() > 0){
                 trustScores.put (user, jc);
              }
              }
              else {
                 if (intersectionMap.size() > 0){
                     trustScores.put (user, 1d);
                 }
```

```
            }
          }
        }
       return trustScores;
   }


   /**
    * @param userTrusteesMap the input map
    * @param trustScores the output map
    * @param source
    * @return the triadic closure
    */
   public static <U> Map<U, Double> findTriadicClosure(Map<U, Map<U,Double>>
   userTrusteesMap, Map<U, Double> trustScores, U source, boolean jcFlag)
   {

      /**
       * Triadic Closure: adds the connection between two nodes with common
neighbors
       */

         for (U user : userTrusteesMap.keySet()){
           if ( ! user.equals(source) ){

              // get the map of source
              Set<U> sourceMap = userTrusteesMap.get(source) != null ?
                     userTrusteesMap.get(source).keySet() : null;

              //get the map of users
              Set<U> userMap = userTrusteesMap.get(user) != null ?
                     userTrusteesMap.get(user).keySet() : null;

              if (userMap == null || sourceMap == null ) continue;

              Set<U> intersectionMap = findIntersection(userMap, sourceMap);

              //calculate the Jaccard Coefficient
              if ( jcFlag ) {
                 Double jc  = getJaccardCoefficient(sourceMap,  userMap);


              if (intersectionMap.size() > 0){
                 trustScores.put (user, jc);
              }
              }
              else {
                 if (intersectionMap.size() > 0){
                 trustScores.put (user, 1d);
              }
              }
           }
         }
       return trustScores;
   }

   /**
    * Calculate the Jaccard Coefficient (jc) as the intersection of two given
sets/the union of these sets
    * The sets are built here from the given map: there is one set built for
source and another for user
    * @param userTrusteesMap
    * @param source
    * @param user
    * @return the Jaccard Coefficient or 1 in case source map or user map is
empty or in case their intersection is empty.
    */
   public static <U> Double getJaccardCoefficient( Map<U, Map<U,Double>>
```

```
   userTrusteesMap, U source, U user) {
       // get the map of source
       Set<U> sourceMap = userTrusteesMap.get(source) != null ?
            userTrusteesMap.get(source).keySet() : null;

       //get the map of users
       Set<U> userMap = userTrusteesMap.get(user) != null ?
            userTrusteesMap.get(user).keySet() : null;

       if (userMap != null && sourceMap != null ) {
          Set<U> intersectionMap = findIntersection(userMap, sourceMap);

          if (intersectionMap.size() == 0){
             return 0d;
          }

          Set<U> unionMap = findUnion(userMap, sourceMap);
          Double jc =  unionMap.size() > 0 ?
            (double)intersectionMap.size() /
(double)unionMap.size():(double)0;
          jc = round(jc);

          return jc;
       }
       else {
          return 0d;
       }
   }

   /**
    * Calculate the Jaccard Coefficient (jc) as the intersection of two given
sets/the union of these sets
    * @param intersectionMap the intersection of two sets
    * @param unionMap the union of two sets
    * @return the Jaccard Coefficient
    */
   private static <U> Double getJaccardCoefficient( Set<U> sourceMap, Set<U>
userMap) {
      Set<U> intersectionMap = findIntersection(userMap, sourceMap);
      Set<U> unionMap = findUnion(userMap, sourceMap);
      Double jc =  unionMap.size() > 0 ?
           (double)intersectionMap.size() / (double)unionMap.size() :
(double)0;
      jc = round(jc);

      return jc;
   }

   /**
    * Rounds a double value to 2 decimal places
    * @param value a double value
    * @return rounded double to 2 decimal places
    */
   private static double round(double value) {
       BigDecimal bd = new BigDecimal(value);
       bd = bd.setScale(2, RoundingMode.HALF_UP);
       return bd.doubleValue();
   }

   /**
    * Finds the common elements between two given sets
    * @param set1
    * @param set2
    * @return the intersection of 2 given sets
    */
   private static <U> Set<U> findIntersection(Set<U> set1, Set<U> set2){
      Set<U> commonSet = new HashSet<U>();
      for (U element1 : set1){
```

```
        for (U element2 : set2 ) {
          if (element1.equals(element2)){
            commonSet.add(element1);
          }
        }
      }
      return commonSet;
    }

    /**
     * Finds the union of elements between two given sets
     * @param set1
     * @param set2
     * @return the union of 2 given sets
     */
    private static <U> Set<U> findUnion(Set<U> set1, Set<U> set2){
      Set<U> commonSet = new HashSet<U>();

      commonSet.addAll(set1);
      commonSet.addAll(set2);

      return commonSet;
    }
}
```

References

Abdollahi, B. (2017) ‘Using Explainability for Constrained Matrix Factorization’, pp. 79–83. doi: 10.1145/3109859.3109913.

Aberer, K. *et al.* (2006) ‘The Complex Facets of Reputation and Trust’, *Computational Intelligence, Theory and Applications*. Edited by B. Reusch. Springer Berlin Heidelberg, 38, pp. 281–294. doi: 10.1007/3-540-34783-6_29.

Aberer, K. and Despotovic, Z. (2001) ‘Managing trust in a peer-2-peer information system’, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pp. 310–317. doi: http://doi.acm.org/10.1145/502585.502638.

Adali, S. (2013) *Modeling Trust Context in Networks*. Springer Publishing Company, Incorporated. Available at: http://dl.acm.org/citation.cfm?id=2490542 (Accessed: 25 September 2014).

Adamic, L. A. and Adar, E. (2003) ‘Friends and neighbors on the Web’, *Social Networks*, 25(3), pp. 211–230. doi: 10.1016/S0378-8733(03)00009-1.

Adomavicius, G. and Tuzhilin, A. (2005) ‘Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions’, *IEEE Transactions on Knowledge and Data Engineering*. Piscataway, NJ, USA: IEEE Educational Activities Department, 17(6), pp. 734–749. doi: http://dx.doi.org/10.1109/TKDE.2005.99.

Aggarwal, C. (2011) *Social Network Data Analytics*. Edited by C. C. Aggarwal. Boston, MA: Springer US. doi: 10.1007/978-1-4419-8462-3.

Aiello, L. M. *et al.* (2012) ‘Friendship prediction and homophily in social media’, *ACM Transactions on the Web*, 6(2), pp. 1–33. doi: 10.1145/2180861.2180866.

Airoldi, E. M. *et al.* (2006) ‘Mixed membership stochastic block models for relational data with application to protein-protein interactions’, *Proceedings of the international biometrics society annual meeting*, pp. 1–34. Available at: http://www-2.cs.cmu.edu/~epxing/papers/ENAR06.pdf.

Anagnostopoulos, A., Kumar, R. and Mahdian, M. (2008) ‘Influence and correlation in social networks’, *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining KDD 08*, 10(1), p. 7. doi: 10.1145/1401890.1401897.

Andersen, R. *et al.* (2008) ‘Trust-based recommendation systems: an axiomatic approach’, in *Proceeding of the 17th international conference on World Wide Web*. ACM, pp. 199–208.

Avesani, P. and Massa, P. (2005) ‘Moleskiing. it: a trust-aware recommender system for ski mountaineering’, *International Journal for Infonomics*, pp. 1–19. doi: 10.1145/1067036.

Azadjalal, M. M., Moradi, P. and Abdollahpouri, A. (2014) ‘Application of game theory techniques for improving trust based recommender systems in social networks’, *Proceedings of the 4th International Conference on Computer and Knowledge*

Engineering, ICCKE 2014, pp. 261–266. doi: 10.1109/ICCKE.2014.6993436.

Backes, M. *et al.* (2010) 'Anonymous webs of trust', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 130–148. doi: 10.1007/978-3-642-14527-8_8.

Balabanović, M. and Shoham, Y. (1997) 'Fab: content-based, collaborative recommendation', *Communications of the ACM*, 40(3), pp. 66–72. doi: 10.1145/245108.245124.

Banks, D. L. and Carley, K. M. (1996) 'Models for network evolution', *Journal of Mathematical Sociology*, 1995(March), pp. 1–36. doi: 10.1016/j.ehb.2012.05.003.

Barber, K. S. and Kim, J. (2001) 'Belief Revision Process Based on Trust: Agents Evaluating Reputation of Information Sources', in Falcone, R., Singh, M., and Tan, Y.-H. (eds) *Trust in Cyber-societies*. Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 73–82. doi: 10.1007/3-540-45547-7_5.

Basu, C., Hirsh, H. and Cohen, W. (1998) 'Recommendation as Classification : Using Social and Content-Based Information in Recommendation', *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 714–720. doi: 10.1.1.36.4620.

Bedi, P. and Kaur, H. (2006) 'Trust based Personalized Recommender System', *INFOCOM Journal of Computer Science*, 5(1), pp. 19–26.

Bedi, P., Kaur, H. and Marwaha, S. (2007) 'Trust based recommender system for the semantic web', in *Proceedings of the 20th international joint conference on Artifical intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc (IJCAI'07), pp. 2677–2682. Available at: http://dl.acm.org/citation.cfm?id=1625275.1625706.

Bedi, P. and Sharma, R. (2012) 'Trust based recommender system using ant colony for trust computation', *Expert Systems with Applications*. Elsevier Ltd, 39(1), pp. 1183–1190. doi: 10.1016/j.eswa.2011.07.124.

Bennett, J. and Lanning, S. (2007) 'The Netflix Prizes'. Available at: www.netflixprize.com.

Bhuiyan, T. (2013) *Trust for Intelligent Recommendation*. doi: 10.1007/978-1-4614-6895-0.

Bianconi, G. *et al.* (2014) 'Triadic closure as a basic generating mechanism of communities in complex networks', *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 90(4). doi: 10.1103/PhysRevE.90.042806.

Bing Wu, Luo Qi and Xiong Feng (2007) 'Personalized Recommendation Algorithm based on SVM', in *2007 International Conference on Communications, Circuits and Systems*, pp. 951–953. doi: 10.1109/ICCCAS.2007.4348205.

Bobadilla, J. *et al.* (2013) 'Recommender systems survey', *Knowledge-Based Systems*. Elsevier B.V., 46, pp. 109–132. doi: 10.1016/j.knosys.2013.03.012.

Bowman, D. a., Gabbard, J. L. and Hix, D. (2002) 'A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods', *Presence: Teleoperators and Virtual Environments*, 11(4), pp. 404–424. doi: 10.1162/105474602760204309.

Breese, J. S., Heckerman, D. and Kadie, C. (1998) 'Empirical analysis of predictive algorithms for collaborative filtering', in *UAI'98 Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 43–52. Available at: http://delivery.acm.org/10.1145/2080000/2074100/p43-breese.pdf?ip=195.251.66.108&id=2074100&acc=ACTIVE SERVICE&key=5641A0C343C36AC1.EDF3AD695DCBC58A.4D4702B0C3E38B35.4D4702B0C3E38B35&CFID=770399962&CFTOKEN=27232126&__acm__=1460628721_15ac710a2e6dde6.

Brin, S. and Page, L. (1998) 'The Anatomy of a Large-scale Hypertextual Web Search Engine', in *Proceedings of the Seventh International Conference on World Wide Web 7*. Amsterdam, The Netherlands: Elsevier Science Publishers B. V (WWW7), pp. 107–117. Available at: http://dl.acm.org/citation.cfm?id=297805.297827.

Burke, R. (2000) 'Knowledge-based recommender systems', *Encyclopedia of library and information systems*, 69(Supplement 32), pp. 175–186. doi: 10.2991/iske.2007.110.

Burke, R. (2002) 'Hybrid Recommender Systems: Survey and experiments', *User Modeling and UserAdapted Interaction*, 12(4), pp. 331–370. doi: 10.1023/A:1021240730564].

Burton, R. (1621) *The Anatomy of Melanholy*.

Canny, J. (2002) 'Collaborative Filtering with Privacy via Factor Analysis', *Proceeding SIGIR '02 Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, (i), pp. 238–245. doi: 10.1145/564376.564419.

Cantador, I., Bellogín, A. and Castells, P. (2008) 'A multilayer ontology-based hybrid recommendation model', *AI Communications*, 21(2–3), pp. 203–210. doi: 10.3233/AIC-2008-0437.

Capuruço, R. A. C. and Capretz, L. F. (2012) 'A fuzzy-based inference mechanism of trust for improved social recommenders.', in Herder, E. et al. (eds) *UMAP Workshops*. CEUR-WS.org (CEUR Workshop Proceedings). Available at: http://dblp.uni-trier.de/db/conf/um/umap2012w.html#CapurucoC12.

Casino, F. *et al.* (2015) 'A k-anonymous approach to privacy preserving collaborative filtering', *Journal of Computer and System Sciences*, 81(6), pp. 1000–1011. doi: 10.1016/j.jcss.2014.12.013.

Castelfranchi, C. and Falcone, R. (2010) *Trust Theory: A socio-cognitive and computational model*, *Zhurnal Eksperimental'noi i Teoreticheskoi Fiziki*. Wiley.

Centola, D. (2010) 'The spread of behavior in an online social network experiment.', *Science (New York, N.Y.)*, 329(5996), pp. 1194–7. doi: 10.1126/science.1185231.

Chakraborty, P. S. and Karform, S. (2012) 'Designing Trust Propagation Algorithms based on Simple Multiplicative Strategy for Social Networks', *Procedia Technology*, 6, pp. 534–539. doi: 10.1016/j.protcy.2012.10.064.

Charif, H., Anne, B. and Azim, R. (2012) 'Hybridising collaborative filtering and trust-aware recommender systems', in *WEBIST 2012 - Proceedings of the 8th International Conference*. New York, NY, USA: SciTePress, pp. 695–700. Available at: http://hal.inria.fr/hal-00679233 (Accessed: 25 September 2014).

Chee, S. H. S., Han, J. and Wang, K. (2001) 'RecTree: An efficient collaborative filtering method', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2114, pp. 141–151. doi: 10.1007/3-540-44801-2_15.

Chen, G. *et al.* (2005) 'A Fuzzy Trust Model for Multi-agent System', *Advances in Natural Computation*. Edited by L. Wang, K. Chen, and Y. Ong. Springer Berlin Heidelberg (Lecture Notes in Computer Science), 3612, pp. 444–448. doi: 10.1007/11539902_53.

Chirita, P.-A., Nejdl, W. and Zamfir, C. (2005) 'Preventing shilling attacks in online recommender systems', *Proceedings of the seventh ACM international workshop on Web information and data management WIDM 05*, 55(2), p. 67. doi: 10.1145/1097047.1097061.

De Choudhury, M. *et al.* (2010) '"Birds of a Feather": Does User Homophily Impact Information Diffusion in Social Media?', *arXiv:1006.1702 [physics]*, pp. 1–31. Available at: http://arxiv.org/abs/1006.1702%5Cnhttp://www.arxiv.org/pdf/1006.1702.pdf.

Christakis, N. A. and Fowler, J. H. (2008) 'The Collective Dynamics of Smoking in a Large Social Network', *New England Journal of Medicine*, 358(21), pp. 2249–2258. doi: 10.1056/NEJMsa0706154.

Christakis, N. a and Fowler, J. H. (2007) 'The spread of obesity in a large social network over 32 years.', *The New England journal of medicine*, 357(4), pp. 370–9. doi: 10.1056/NEJMsa066082.

Ciotti, V. *et al.* (2016) 'Homophily and missing links in citation networks', *EPJ Data Science*. Ciotti et al., 5(1). doi: 10.1140/epjds/s13688-016-0068-2.

Colquitt, J. A., Scott, B. A. and LePine, J. A. (2007) 'Trust, trustworthiness, and trust propensity: a meta-analytic test of their unique relationships with risk taking and job performance.', *The Journal of applied psychology*, 92(4), pp. 909–927. doi: 10.1037/0021-9010.92.4.909.

Cui, J., Wang, F. and Zhai, J. (2010) 'Citation Networks as a Multi-layer Graph: Link Prediction and Importance Ranking', *snap.stanford.edu*. Available at: http://snap.stanford.edu/class/cs224w-2010/proj2010/05_ProjectReport.pdf (Accessed: 25 September 2014).

Deerwester, S. *et al.* (1990) 'Indexing by Latent Semantic Analysis', *Journal of the American Society for Information Science*, 41(6), pp. 391–407. doi: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

Deutsch, M. (1958) 'Trust and suspicion', *Journal of Conflict Resolution*, 2(4), pp. 265–279. doi: 10.1177/002200275800200401.

Dodds, P. S., Muhamad, R. and Watts, D. J. (2003) 'An Experimental Study of Search in Global Social Networks', 301(August).

Dokoohaki, N. and Matskin, M. (2008) 'Effective Design of Trust Ontologies for Improvement in the Structure of Socio-Semantic Trust Networks', *International Journal On Advances in Intelligent Systems*, 1(1942–2679), pp. 23–42. Available at: http://www.iariajournals.org/intelligent_systems/intsys_v1_n1_2008_paged.pdf.

Dong, X. and Frossard, P. (2012) 'Clustering with multi-layer graphs: A spectral perspective', *IEEE Transactions on Signal Processing*, 60(11), pp. 5820–5831. doi: 10.1109/TSP.2012.2212886.

Easley, D. and Kleinberg, J. (2010) 'Networks , Crowds , and Markets : Reasoning about a Highly Connected World', *Science*, 81, p. 744. doi: 10.1017/CBO9780511761942.

Ekstrand, M. D., Riedl, J. T. and Konstan, J. A. (2007) 'Collaborative Filtering Recommender Systems', *Foundations and Trends in Human-Computer Interaction*, 4321(1), pp. 291–324. doi: 10.1504/IJEB.2004.004560.

Elmagarmid, A. K., Ipeirotis, P. G. and Verykios, V. S. (2007) 'Duplicate record detection: A survey', *IEEE Transactions on Knowledge and Data Engineering*, 19(1), pp. 1–16. doi: 10.1109/TKDE.2007.250581.

ElSalamouny, E., Sassone, V. and Nielsen, M. (2010) 'HMM-Based Trust Model', *Formal Aspects in Security and Trust*. Edited by P. Degano and J. Guttman. Springer Berlin Heidelberg (Lecture Notes in Computer Science), 5983, pp. 21–35. doi: 10.1007/978-3-642-12459-4_3.

Falcone, R. and Castelfranchi, C. (2001) *Trust and Deception in Virtual Societies*. Edited by C. Castelfranchi and Y.-H. Tan. Norwell, MA, USA: Kluwer Academic Publishers. doi: 10.1007/978-94-017-3614-5.

Falcone, R., Pezzulo, G. and Castelfranchi, C. (2003) 'A Fuzzy Approach to a Belief-based Trust Computation', in *Proceedings of the 2002 International Conference on Trust, Reputation, and Security: Theories and Practice*. Berlin, Heidelberg: Springer-Verlag (AAMAS'02), pp. 73–86. Available at: http://dl.acm.org/citation.cfm?id=1762128.1762135.

Fang, H., Guo, G. and Zhang, J. (2015) 'Multi-faceted trust and distrust prediction for recommender systems', *Decision Support Systems*, 71, pp. 37–47. doi: 10.1016/j.dss.2015.01.005.

Felden, C. and Linden, M. (2007) 'Ontology-Based User Profiling', in Abramowicz, W. (ed.) *Business Information Systems*. TU Bergakademie Freiberg, Fakultät für Wirtschaftswissenschaft, Professur ABWL, Informationswirtschaft/Wirtschaftsinformatik, Lessingstraße 45, 09599 Freiberg: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 314–327. Available at: http://dx.doi.org/10.1007/978-3-540-72035-5_24.

Fleder, D. M. and Hosanagar, K. (2007) 'Recommender systems and their impact on sales diversity', *Proceedings of the 8th ACM conference on Electronic commerce EC 07*, 55, p. 192. doi: 10.1145/1250910.1250939.

Fouss, F. *et al.* (2007) 'Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation', *IEEE Trans.on Knowl.and Data Eng.* Piscataway, NJ, USA: IEEE Educational Activities Department, 19(3), pp. 355–369. doi: 10.1109/TKDE.2007.46.

Gambetta, D. (1988) 'Can we trust trust?', in *Trust: Making and Breaking Cooperative Relations.*, pp. 213–237. doi: 10.2307/2076328.

Gao, M., Liu, K. and Wu, Z. (2010) 'Personalisation in web computing and informatics: Theories, techniques, applications, and future research', *Information Systems Frontiers*. Hingham, MA, USA: Kluwer Academic Publishers, 12(5), pp. 607–629. Available at:

http://dx.doi.org/10.1007/s10796-009-9199-3.

Gao, P. *et al.* (2016) 'STAR: Semiring Trust Inference for Trust-Aware Social Recommenders', *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, (ACM), pp. 301–308. doi: 10.1145/2959100.2959148.

Gao, Q. (2010) 'Towards Trust in Web Content Using Semantic Web', in Aroyo, L. et al. (eds) *The Semantic Web: Research and Applications: 7th Extended Semantic Web Conference, ESWC 2010*. Heraklion, Crete, Greece: Springer Berlin Heidelberg, pp. 457–461. doi: 10.1007/978-3-642-13489-0_43.

Ghazanfar, M. A. and Prügell-Bennett, A. (2014) 'Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems', *Expert Systems with Applications*. Elsevier Ltd, 41(7), pp. 3261–3272. doi: 10.1016/j.eswa.2013.11.010.

Golbeck, J. A. (2005) *Computing and applying trust in web-based social networks*.

Golbeck, J. A. (2006a) 'Combining provenance with trust in social networks for semantic web content filtering', in *Provenance and Annotation of Data*. Available at: http://link.springer.com/chapter/10.1007/11890850_12 (Accessed: 25 September 2014).

Golbeck, J. A. (2006b) 'Generating Predictive Movie Recommendations from Trust in Social Networks', in Stølen, K. et al. (eds) *Trust Management*. University of Maryland, College Park, 8400 Baltimore Avenue, College Park, Maryland 20740: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 93–104. Available at: http://dx.doi.org/10.1007/11755593_8.

Golbeck, J. A. and Hendler, J. (2006) 'Inferring binary trust relationships in Web-based social networks', *ACM Transactions on Internet Technology*, 6(4), pp. 497–529. doi: 10.1145/1183463.1183470.

Golbeck, J. A., Parsia, B. and Hendler, J. (2003) 'Trust Networks on the Semantic Web', in Klusch, M. et al. (eds) *Cooperative Information Agents VII*. Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 238–249. doi: 10.1007/978-3-540-45217-1_18.

Golbeck, J. and Hendler, J. (2004) 'Reputation Network Analysis for Email Filtering', *Proceedings of the 1st Conference on Email and Anti-Spam*, 44, pp. 54–58. doi: 10.1.1.59.8119.

Goldberg, D. *et al.* (1992) 'Using collaborative filtering to weave an information tapestry', *Commun.ACM*. New York, NY, USA: ACM, 35(12), pp. 61–70. doi: http://doi.acm.org/10.1145/138859.138867.

Goldberg, K. *et al.* (2001) 'Eigentaste: A Constant Time Collaborative Filtering Algorithm', *Information Retrieval*, 4(2), pp. 133–151. doi: 10.1023/A:1011419012209.

Granovetter, M. (1985) 'Economic-action and social-structure - the problem of embeddedness', *American Journal of Sociology*, pp. 481–510. doi: Doi 10.1086/228311.

Gray, E. *et al.* (2003) *Trust Propagation in Small Worlds*, *Lecture Notes in Computer Science*. doi: 10.1007/3-540-44875-6_17.

Gretzel, U. and Yoo, K. H. (2008) 'Information and Communication Technologies in Tourism 2008; Use and Impact of Online Travel Reviews', pp. 35–46. Available at:

http://dx.doi.org/10.1007/978-3-211-77280-5_4.

Guanfeng, L. *et al.* (2010) 'A Heuristic Algorithm for Trust-Oriented Service Provider Selection in Complex Social Networks', in *Services Computing (SCC), 2010 IEEE International Conference on*, pp. 130–137. doi: 10.1109/SCC.2010.47.

Guanfeng, L., Yan, W. and Orgun, M. A. (2009) 'Trust Inference in Complex Trust-Oriented Social Networks', in *Computational Science and Engineering, 2009. CSE '09. International Conference on*, pp. 996–1001. doi: 10.1109/CSE.2009.248.

Guha, R. *et al.* (2004) 'Propagation of trust and distrust', in *Proceedings of the 13th international conference on World Wide Web*. New York, NY, USA: ACM (WWW '04), pp. 403–412. doi: 10.1145/988672.988727.

Guibing, G. *et al.* (2015) 'LibRec: A Java Library for Recommender Systems', in *Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP)*. Dublin, Ireland.

Guo, G. *et al.* (2014) 'From Ratings to Trust: An Empirical Study of Implicit Trust in Recommender Systems', in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 248–253. doi: 10.1145/2554850.2554878.

Guo, G., Zhang, J. and Thalmann, D. (2014) 'Merging trust in collaborative filtering to alleviate data sparsity and cold start', *Knowledge-Based Systems*. Elsevier B.V., 57, pp. 57–68. doi: 10.1016/j.knosys.2013.12.007.

Guo, G., Zhang, J. and Yorke-Smith, N. (2013) 'A Novel Bayesian Similarity Measure for Recommender Systems', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*. Beijing, China: AAAI Press, pp. 2619–2625.

Hamdi, S. *et al.* (2013) 'Trust inference computation for online social networks', *Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013*, pp. 210–217. doi: 10.1109/TrustCom.2013.240.

Hang, C. and Singh, M. P. (2010) 'Trust-based recommendation based on graph similarity', in *in AAMAS Workshop on Trust in Agent Societies (Trust*, pp. 1–11.

Hang, C. W., Wang, Y. and Singh, M. P. (2009) 'Operators for propagating trust and their evaluation in social networks', in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 1025–1032. Available at: http://portal.acm.org/citation.cfm?id=1558155.

Harford, T. (2010) 'The Economics Of Trust', *Forbes*, 21 July. Available at: http://www.forbes.com/2006/09/22/trust-economy-markets-tech_cx_th_06trust_0925harford.html (Accessed: 25 September 2014).

Hasan, M. Al and Zaki, M. J. (2011) 'A Survey of Link Prediction in Social Networks', in Aggarwal, C. (ed.) *Social Network Data Analytics*. Springer US, pp. 243–275. doi: 10.1007/978-1-4419-8462-3.

Heath, T., Motta, E. and Petre, M. (2007) 'Computing word-of-mouth trust relationships in social networks from semantic web and web 2.0 data sources', in *4th European Semantic Web Conference (ESWC 2007)*. Innsbruck, Austria, pp. 44–56. Available at: http://oro.open.ac.uk/23610/ (Accessed: 25 September 2014).

Heckmann, D., Schwartz, T., Brandherm, B. and Kr, A. (2005) ‘Decentralized User Modeling with UserML and GUMO’, in *Workshop on Decentralized, Agent Based and Social Approaches to User Modelling (DASUM), 9th Intl Conference on User Modeling*, pp. 61–64.

Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., *et al.* (2005) ‘Gumo – The General User Model Ontology’, in Ardissono, L., Brna, P., and Mitrovic, A. (eds) *User Modeling 2005*. Saarland University, Saarbrücken, Germany: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), p. 149. Available at: http://dx.doi.org/10.1007/11527886_58.

Heckmann, D. and Krueger, A. (2003) ‘A User Modeling Markup Language (UserML) for Ubiquitous Computing’, in Brusilovsky, P., Corbett, A., and de Rosis, F. (eds) *User Modeling 2003*. European Post-Graduate College ‘Cognitive Systems and Speech Technology’ Germany: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), p. 148. Available at: http://dx.doi.org/10.1007/3-540-44963-9_55.

Herlocker, J. L. *et al.* (2004) ‘Evaluating collaborative filtering recommender systems’, *ACM Transactions on Information Systems*, 22(1), pp. 5–53. Available at: http://portal.acm.org/citation.cfm?doid=963770.963772.

Heß, C. (2007) *Trust-based recommendations in multi-layer networks*. University of Bamberg.

Homans, G. C. (1951) *The Human Group*, *International Library of Sociology and Social Reconstruction*. doi: 10.2307/2088295.

Hooijmaijers, D. and Stumptner, M. (2007) ‘Trust Calculation’, *Intelligent Information Processing III*. Edited by Z. Shi, K. Shimohara, and D. Feng. Springer US (IFIP International Federation for Information Processing), 228, pp. 111–121. doi: 10.1007/978-0-387-44641-7_12.

Htun, Z. and Tar, P. P. (2013) ‘A Trust-aware Recommender System Based on Implicit Trust Extraction’, *International Journal of Innovations in Engineering and Technology (IJIET)Technology (IJIET)*, 2(1), pp. 271–276. Available at: http://scholar.google.com/scholar?hl=en%5C&btnG=Search%5C&q=intitle:A+Trust-aware+Recommender+System+Based+on+Implicit+Trust+Extraction%5C#0 (Accessed: 25 September 2014).

Ivory, M. Y. and Hearst, M. A. (2001) ‘The state of the art in automating usability evaluation of user interfaces’, *ACM Computing Surveys*, 33(4), pp. 470–516. doi: 10.1145/503112.503114.

Jaccard, P. (1901) ‘Étude comparative de la distribution florale dans une portion des Alpes et des Jura’, *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37(JANUARY 1901), pp. 547–579. doi: http://dx.doi.org/10.5169/seals-266450.

Jamali, M. (2010) ‘A Distributed Method for Trust-Aware Recommendation in Social Networks’. Available at: http://arxiv.org/abs/1011.2245 (Accessed: 15 September 2014).

Jamali, M. and Ester, M. (2009) ‘TrustWalker: a random walk model for combining trust-based and item-based recommendation’, in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. Paris, France: ACM (KDD ’09), pp. 397–406. doi: citeulike-article-id:5151320.

Jannach, D. *et al.* (2010) *Recommender Systems: An Introduction*. 1st edn. New York, NY, USA: Cambridge University Press.

Jeckmans, A. J. P. *et al.* (2013) 'Privacy in Recommender Systems', in *Social Media Retrieval*, pp. 263–281. doi: 10.1007/978-1-4471-4555-4_12.

Jøsang, A. (2001) 'A Logic for Uncertain Probabilities', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*. River Edge, NJ, USA: World Scientific Publishing Co., Inc, 9(3), pp. 279–311. Available at: http://dl.acm.org/citation.cfm?id=565980.565981.

Jøsang, A., Hayward, R. and Pope, S. (2006) 'Trust Network Analysis with Subjective Logic', in *Proceedings of the 29th Australasian Computer Science Conference - Volume 48*. Hobart, Australia: Australian Computer Society, Inc (ACSC '06), pp. 85–94. Available at: http://dl.acm.org/citation.cfm?id=1151699.1151710.

Jøsang, A. and Lo Presti, S. (2004) 'Analysing the Relationship between Risk and Trust', in Jensen, C., Poslad, S., and Dimitrakos, T. (eds) *Trust Management*. Springer Berlin Heidelberg (Lecture Notes in Computer Science), pp. 135–145. doi: 10.1007/978-3-540-24747-0_11.

Kamvar, S. D., Schlosser, M. T. and Garcia-Molina, H. (2003) 'The Eigentrust algorithm for reputation management in P2P networks', *12th International Conference on World Wide Web (WWW)*, p. 640. doi: 10.1145/775240.775242.

Kim, H.-N. *et al.* (2007) 'User Preference Modeling from Positive Contents for Personalized Recommendation', in Corruble, V., Takeda, M., and Suzuki, E. (eds) *Discovery Science*. Intelligent E-Commerce Systems Laboratory, Department of Computer Science & Information Engineering, Inha University: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 116–126. Available at: http://dx.doi.org/10.1007/978-3-540-75488-6_12.

Kim, H. H. and Kim, H. H. (2012) 'Improving Recommendation based on Implicit Trust Relationships from Tags', in *2nd international conference on computers, networks, systems, and industrial applications*, pp. 25–30.

Kim, S. and Kwon, J. (2007) 'Effective Context-aware Recommendation on the Semantic Web', *International Journal of Computer Science and Network Security*, 7(8), pp. 154–159. Available at: http://paper.ijcsns.org/07_book/200708/20070822.pdf.

Kim, Y. A. and Song, H. S. (2011) 'Strategies for predicting local trust based on trust propagation in social networks', *Knowledge-Based Systems*. Elsevier B.V., 24(8), pp. 1360–1371. doi: 10.1016/j.knosys.2011.06.009.

Kleinberg, J. (2000) 'The Small-World Phenomenon: An Algorithmic Perspective', *in Proceedings of the 32nd ACM Symposium on Theory of Computing*, pp. 163–170. doi: 10.1.1.117.7097.

Koren, Y., Bell, R. and Volinsky, C. (2009) 'Matrix Factorization Techniques for Recommender Systems', *Computer*, 42(8), pp. 42–49. doi: 10.1109/MC.2009.263.

Kossinets, G. and Watts, D. J. (2009) 'Origins of Homophily in an Evolving Social Network', *American Journal of Sociology*, 115(2), pp. 405–450. doi: 10.1086/599247.

Kuter, U. and Golbeck, J. A. (2007) 'Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models', *AAAI*. Available at: http://www.aaai.org/Papers/AAAI/2007/AAAI07-218.pdf (Accessed: 25 September 2014).

Lam, S. K. and Riedl, J. (2004) 'Shilling recommender systems for fun and profit', *Proceedings of the 13th conference on World Wide Web WWW 04*, pp. 393–402. doi: 10.1145/988672.988726.

Laumann, E. O. (1966) *Prestige and association in an urban community; an analysis of an urban stratification system*, *An Advanced study in sociology*.

Lazarsfeld, P. F. and Merton, R. K. (1954) 'Frienship as a Social Process: A Substantive and Methodological Analysis', in *Freedom and Control in Modern Society*.

Lee, J., Sun, M. and Lebanon, G. (2012) 'PREA : Personalized Recommendation Algorithms Toolkit', *The Journal of Machine Learning Research*, 13, pp. 2699–2703.

Lehman, E. W. and Sztompka, P. (2001) 'Trust: A Sociological Theory', *Contemporary Sociology*, 30(4), p. 418. doi: 10.2307/3089802.

Levien, R. (2009) 'Attack-Resistent Trust Metrics', in *Computing with Social Trust*, pp. 121–132. doi: 10.1007/978-1-84800-356-9.

Lewicki, R. J., Mcallister, D. J. and Bies, R. J. (1998) 'Trust and Distrust: New Relationships and Realities', *Academy of Management Review*, 23(3), pp. 438–458.

Li, C. *et al.* (2012) 'Multi-Layer network for influence propagation over microblog', *Intelligence and Security Informatics*, pp. 60–72. Available at: http://link.springer.com/chapter/10.1007/978-3-642-30428-6_5 (Accessed: 25 September 2014).

Li, Y.-M. and Kao, C.-P. (2009) 'TREPPS: A Trust-based Recommender System for Peer Production Services', *Expert Systems with Applications*, 36(2), pp. 3263–3277. doi: 10.1016/j.eswa.2008.01.078.

Liben-Nowell, D. and Kleinberg, J. (2003) 'The Link Prediction Problem for Social Networks', *Proceedings of the Twelfth Annual ACM International Conference on Information and Knowledge Management (CIKM)*, (November 2003), pp. 556–559. doi: 10.1002/asi.v58:7.

Liu, C. *et al.* (2004) 'Beyond concern: A privacy-trust-behavioral intention model of electronic commerce', *Information and Management*, 42(1), pp. 127–142. doi: 10.1016/j.im.2004.01.002.

Liu, Y., Cao, X. and Yu, Y. (2016) 'Are You Influenced by Others When Rating?', *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, pp. 269–272. doi: 10.1145/2959100.2959141.

Lou, T. *et al.* (2013) 'Learning to predict reciprocity and triadic closure in social networks', *ACM Transactions on Knowledge Discovery from Data*, 7(2), pp. 1–25. doi: 10.1145/2499907.2499908.

Lu, L. and Zhou, T. (2010) 'Link Prediction in Complex Networks: A Survey', *Physica A: Statistical Mechanics and its Applications*, 390(6), pp. 1150–1170. doi: 10.1016/j.physa.2010.11.027.

Lucas, J. P. *et al.* (2013) 'A hybrid recommendation approach for a tourism system', *Expert Systems with Applications*, 40(9), pp. 3532–3550. doi: 10.1016/j.eswa.2012.12.061.

Luhmann, N. (2000) 'Familiarity, Confidence, Trust: Problems and Alternatives', in *Trust: Making and Breaking Cooperative Relations*, pp. 94–107. doi: 10.1088/1751-8113/44/8/085201.

Ma, H., King, I. and Lyu, M. R. (2011) 'Learning to recommend with explicit and implicit social relations', *ACM Transactions on Intelligent Systems and Technology (TIST)*. New York, NY, USA: ACM, 2(3), p. 29 (1-19). doi: http://doi.acm.org/10.1145/1961189.1961201.

Marsden, P. V and Friedkin, N. E. (1993) 'Network Studies of Social Influence', *Sociological Methods & Research*, 22(1), pp. 127–151. doi: 10.1177/0049124193022001006.

Marsh, S. P. (1994) *Formalising trust as a computational concept*. University of Stirling.

Martín-Vicente, M. I., Gil-solla, A. and Ramos-Cabrer, M. (2012) 'Implicit Trust Networks: A Semantic Approach to Improve Collaborative Recommendations', *Recommender Systems for the Social Web*. Springer Berlin Heidelberg (Intelligent Systems Reference Library), 32, pp. 107–119. doi: 10.1007/978-3-642-25694-3_5.

Massa, P. and Avesani, P. (2004) 'Trust-Aware Collaborative Filtering for Recommender Systems', in Meersman, R. and Tari, Z. (eds) *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*. ITC-iRST, Via Sommarive 14, I-38050 Povo, TN, Italy: Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 492–508. Available at: http://dx.doi.org/10.1007/978-3-540-30468-5_31.

Massa, P. and Avesani, P. (2005) 'Controversial users demand local trust metrics: An experimental study on epinions. com community', in *Proceedings of the National Conference on Artificial Intelligence*. Pittsburgh, Pennsylvania: AAAI Press (AAAI'05), p. 121. Available at: http://dl.acm.org/citation.cfm?id=1619332.1619354.

Massa, P. and Avesani, P. (2007) 'Trust-aware recommender systems', *Proceedings of the 2007 ACM conference on Recommender systems RecSys 07*, 20, pp. 17–24. doi: 10.1145/1297231.1297235.

Massa, P. and Avesani, P. (2009) 'Trust metrics in recommender systems', *Computing with social trust*, pp. 1–27. Available at: http://link.springer.com/chapter/10.1007/978-1-84800-356-9_10 (Accessed: 25 September 2014).

Maurer, U. (1996) 'Modelling a Public-Key Infrastructure', *Lncs*, pp. 1–26. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.4069&rep=rep1&type=pdf.

Mayer, R. C., Davis, J. H. and Schoorman, F. D. (1995) 'An Integrative Model of Organizational Trust', *The Academy of Management Review*, 20(3), pp. 709–734. doi: 10.2307/258792.

McGrath, M. (2008) 'Employing 'Social Network Analysis' to Influence Tourism Events Decision-Making: A Pilot Study', in *Information and Communication Technologies in Tourism 2008*, pp. 556–567. doi: 10.1007/978-3-211-77280-5_49.

McKnight, D. H. and Chervany, N. L. (2001) ‘Trust and distrust definitions: One bite at a time’, *Trust in Cyber-societies*. Springer, pp. 27–54. Available at: http://link.springer.com/chapter/10.1007/3-540-45547-7_3 (Accessed: 25 September 2014).

McPherson, J. M. and Smith-Lovin, L. (1987) ‘Homophily in Voluntary Organizations: Status Distance and the Composition of Face-to-Face Groups’, *American Sociological Review*, 52(3), p. 370. doi: 10.2307/2095356.

McPherson, M., Smith-Lovin, L. and Cook, J. M. (2001) ‘Birds of a Feather: Homophily in Social Networks’, *Annual Review of Sociology*. McPherson, M (Reprint Author), Univ Arizona, Dept Sociol, Tucson, AZ 85721 USA. Univ Arizona, Dept Sociol, Tucson, AZ 85721 USA. Duke Univ, Dept Sociol, Durham, NC 27708 USA.: ANNUAL REVIEWS, 27(1), pp. 415–444. doi: 10.1146/annurev.soc.27.1.415.

Mehta, H. *et al.* (2011) ‘Collaborative Personalized Web Recommender System using Entropy based Similarity Measure’, *International Journal of Computer Science Issues (IJCSI)*, 8(6 No 3), pp. 231–240.

Mehta, S. and Banati, H. (2012) ‘Trust aware social context filtering using Shuffled frog leaping algorithm’, in *12th International Conference on Hybrid Intelligent Systems, {HIS} 2012, Pune, India, December 4-7, 2012*. IEEE, pp. 342–347. doi: 10.1109/HIS.2012.6421358.

Mei, J. *et al.* (2017) ‘A social influence based trust model for recommender systems’, *Intelligent Data Analysis*, 21(2), pp. 263–277. doi: 10.3233/IDA-150479.

Meyffret, S. *et al.* (2012) *RED : a Rich Epinions Dataset for Recommender Systems*. Available at: https://hal.archives-ouvertes.fr/hal-01010246.

Milgram, S. (1967) ‘The small world problem’, *Psychology today*, 1(May), pp. 61–67. doi: 10.1007/BF02717530.

Mobasher, B. *et al.* (2005) ‘Effective Attack Models for Shilling Item-Based Collaborative Filtering System’, *Proceedings of the seventh International Workshop on Knowledge Discovery from the Web (WEBKDD 2005)*, pp. 13–23. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.7759&rep=rep1&type=pdf#page=23.

Mobasher, B., Jin, X. and Zhou, Y. (2004) ‘Semantically enhanced collaborative filtering on the web’, *Lecture Notes in Computer Science*, 3209(49), pp. 57–76. doi: 10.1007/978-3-540-30123-3_4.

Mooney, R. J., Bennett, P. N. and Roy, L. (1998) ‘Book Recommending Using Text Categorization with Extracted Information’, *Proceedings of AAAI-98 Workshop on Recommender Systems*, (July), pp. 49–54.

Mui, L. (2002) *Computational Models of Trust and Reputation : Agents , Evolutionary Games , and Social Networks*. Massachusetts Institute of Technology.

Mui, L., Mohtashemi, M. and Halberstadt, A. (2002) ‘A Computational Model of Trust and Reputation for E-businesses’, in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)-Volume 7*. Washington, DC, USA: IEEE Computer Society (HICSS ’02), p. 188. Available at: http://dl.acm.org/citation.cfm?id=820745.821158.

Nageswara Rao, K. and Talwar G., V. (2008) 'Application domain and functional classification of recommender systems—a survey', *DESIDOC Journal of Library & Information Technology*, 28(3), pp. 17–35.

Nanas, N., De Roeck, A. and Vavalis, M. (2009) 'What Happened to Content-Based Information Filtering?', in Azzopardi, L. et al. (eds) *Advances in Information Retrieval Theory*. Springer Berlin / Heidelberg (Lecture Notes in Computer Science), pp. 249–256. Available at: http://dx.doi.org/10.1007/978-3-642-04417-5_23.

Newman, M. E. J. (2001) 'Clustering and preferential attachment in growing networks', *Physical Review E*, 64(2), p. 25102. doi: 10.1103/PhysRevE.64.025102.

Newmann, M. E. J. (2003) 'The structure and function of complex networks', *SIAM Review*, 45(2), p. 58. doi: 10.1137/S003614450342480.

Nielsen (2015) *Global Trust in Advertising Winning Strategies for an Evolving Media Landscape*.

Nolen-Hoeksema, S., Fredrickson, B. and Loftus, G. (2009) *Atkinson & Hilgard's Introduction to Psychology*, *Cengage Learning*. doi: 9781844807284.

O'Connor, P. (2008) 'Information and Communication Technologies in Tourism 2008; User-Generated Content and Travel: A Case Study on Tripadvisor.Com', pp. 47–58. Available at: http://dx.doi.org/10.1007/978-3-211-77280-5_5.

O'Doherty, D., Jouili, S. and Roy, P. Van (2012) 'Towards trust inference from bipartite social networks', *… on Databases and Social Networks*, pp. 13–18. doi: 10.1145/2304536.2304539.

O'Doherty, D., Jouili, S. and Roy, P. Van (2012) 'Trust-based recommendation: an empirical analysis', in *Sixth ACM Workshop on Social Network Mining and Analysis (SNA-KDD 2012)*. Beijing, China. Available at: http://wan.poly.edu/KDD2012/forms/workshop/SNAKDD2012/doc/a4-doherty.pdf (Accessed: 25 September 2014).

O'Mahony, M. P. and Hurley, N. (2004) 'Collaborative recommendation: A robustness analysis', *ACM Transactions on Internet Technology (TOIT)*, 4(4), pp. 344–377. doi: 10.1145/1031114.1031116.

Oufaida, H. and Nouali, O. (2009) 'Exploiting Semantic Web Technologies for Recommender Systems: A Multi View Recommendation Engine (Short Paper)', in Sarabjot, S. A. et al. (eds) *Proceedings of the 7th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems (ITWP'09), in conjunction with the 21st International Joint Conference on Artificial Intelligence*. Pasadena, California USA. Available at: http://ceur-ws.org/Vol-528/paper10.pdf.

Park, S. T., Seo, K. and Jang, D. (2005) 'Expert system based on artificial neural networks for content-based image retrieval', *Expert Systems with Applications*, 29(3), pp. 589–597. doi: 10.1016/j.eswa.2005.04.027.

Patel, J. *et al.* (2005) 'A Probabilistic Trust Model for Handling Inaccurate Reputation Sources', *Trust Management*. Edited by P. Herrmann, V. Issarny, and S. Shiu. Springer Berlin Heidelberg (Lecture Notes in Computer Science), 3477, pp. 193–209. doi: 10.1007/11429760_14.

Pathak, B. *et al.* (2010) 'Empirical Analysis of the Impact of Recommender Systems on Sales', *Journal of Management Information Systems*, 27(2), pp. 159–188. doi: 10.2753/MIS0742-1222270205.

Pazzani, M. and Billsus, D. (1997) 'Learning and Revising User Profiles: The Identification of Interesting Web Sites', *Machine Learning*, 27(3), pp. 313–331. doi: 10.1023/A:1007369909943.

Pazzani, M. J. (1999) 'A framework for collaborative, content-based and demographic filtering', *Artificial Intelligence Review*, 13(5), pp. 393–408. doi: 10.1023/A:1006544522159.

Pazzani, M., Muramatsu, J. and Billsus, D. (1996) 'Syskill & Webert: Identifying interesting web sites', *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1*, pp. 54–61. doi: citeulike-article-id:1188705.

Polat, H. and Du, W. (2003) 'Privacy-preserving collaborative filtering using randomized perturbation techniques', *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pp. 625–628. doi: 10.1109/ICDM.2003.1250993.

Polatidis, N. *et al.* (2017) 'Privacy-preserving collaborative recommendations based on random perturbations', *Expert Systems with Applications*. Elsevier Ltd, 71, pp. 18–25. doi: 10.1016/j.eswa.2016.11.018.

Portes, A. and Sensenbrenner, J. (1993) 'Embeddedness and Immigration: Notes on the Social Determinants of Economic Action', *American Journal of Sociology*, 98(6), pp. 1320–1350. doi: 10.1086/230191.

Powell, M. J. D. (1981) *Approximation theory and methods*. Cambridge University Press. Available at: http://books.google.gr/books?id=ODZ1OYR3w4cC.

Quercia, D., Hailes, S. and Capra, L. (2007) 'Lightweight distributed trust propagation', *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 282–291. doi: 10.1109/ICDM.2007.64.

Quinlan, J. R. (1984) 'Learning Efficient Classification Procedures and their Application to Chess End Games', *Machine Learning: An Artificial Intelligence Approach*, pp. 463–482. doi: 10.1007/978-3-662-12405-5_15.

Rafailidis, D. (2016) 'Modeling trust and distrust information in recommender systems via joint matrix factorization with signed graphs', *Proceedings of the 31st Annual ACM Symposium on Applied Computing - SAC '16*, pp. 1060–1065. doi: 10.1145/2851613.2851697.

Rafailidis, D. and Crestani, F. (2017) 'Learning to Rank with Trust and Distrust in Recommender Systems', *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17*, pp. 5–13. doi: 10.1145/3109859.3109879.

Rapoport, A. (1953) 'Spread of information through a population with socio-structural bias: I. Assumption of transitivity', *The Bulletin of mathematical biophysics*, 15(4), pp. 523–533. doi: 10.1007/BF02476440".

Ray, S. and Mahanti, A. (2010) 'Improving Prediction Accuracy in Trust-Aware Recommender Systems', in *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*. Washington, DC, USA: IEEE Computer Society (HICSS

’10), pp. 1–9. doi: 10.1109/HICSS.2010.225.

Rennie, J. D. M. and Srebro, N. (2005) ‘Fast Maximum Margin Matrix Factorization for Collaborative Prediction’, *Proceedings of the 22Nd International Conference on Machine Learning*, pp. 713–719. doi: 10.1145/1102351.1102441.

Resnick, P. *et al.* (1994) ‘GroupLens: An Open Architecture for Collaborative Filtering of Netnews’, in *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. New York, NY, USA: ACM (CSCW ’94), pp. 175–186. doi: 10.1145/192844.192905.

Ricci, F. *et al.* (2011) *Recommender Systems Handbook*. Available at: http://dx.doi.org/10.1007/978-0-387-85820-3.

Rich, E. (1979) ‘User modeling via stereotypes’, *Cognitive Science*, 3(4), pp. 329–354. Available at: http://www.sciencedirect.com/science/article/B6W48-4FWF9GC-9/1/26c2b8da340100c45590cc04bc7b8a18.

Richardson, M., Agrawal, R. and Domingos, P. (2003) ‘Trust Management for the Semantic Web’, *Interpretation A Journal Of Bible And Theology*, 3(April), pp. 351–368. doi: 10.1007/978-3-540-39718-2_23.

Rosenquist, J. N. *et al.* (2010) ‘The spread of alcohol consumption behavior in a large social network’, *Annals of Internal Medicine*, 152(7), pp. 426–433. doi: 10.7326/0003-4819-152-7-201004060-00007 [doi].

Roy, P. Van, Jouili, S. and Skhiri, S. (2012) *Structural trust inference for social recommendation*.

Ruef, M. *et al.* (2003) ‘The Structure of Founding Teams: Homophily, Strong Ties, and Isolation among U.S. Entrepreneurs’, *AMERICAN SOCIOLOGICAL REVIEW*, 68(2), pp. 195–222. Available at: http://www.jstor.org/stable/1519766.

Salakhutdinov, R. and Mnih, A. (2007) ‘Probabilistic Matrix Factorization.’, *Proc. Advances in Neural Information Processing Systems 20 (NIPS 07)*, pp. 1257–1264. doi: 10.1145/1390156.1390267.

Salakhutdinov, R. and Mnih, A. (2008) ‘Bayesian probabilistic matrix factorization using Markov chain Monte Carlo’, *Proceedings of the 25th international conference on Machine learning - ICML ’08*, pp. 880–887. doi: 10.1145/1390156.1390267.

Salton, G. (1989) *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Sarwar, B. M. *et al.* (1998) ‘Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system’, in *Proceedings of the 1998 ACM conference on Computer supported cooperative work - CSCW ’98*. New York, New York, USA: ACM Press, pp. 345–354. doi: 10.1145/289444.289509.

Sarwar, B. M. *et al.* (2000) ‘Application of Dimensionality Reduction in Recommender System - A Case Study’, *Architecture*, 1625, pp. 264–8. doi: 10.1.1.38.744.

Sarwar, B. M. *et al.* (2002) ‘Recommender Systems for Large-scale E-Commerce : Scalable Neighborhood Formation Using Clustering’, *Communications*, 50(12), pp. 158–167. doi:

10.1.1.4.6985.

Schwartz, B. (2004) *The paradox of choice: why more is less*. ECCO. Available at: http://books.google.com/books?id=zutxr7rGc_QC.

Selvaraj, C. and Anand, S. (2012) 'Peer profile based trust model for P2P systems using genetic algorithm', *Peer-to-Peer Networking and Applications*. Springer US, 5(1), pp. 92–103. doi: 10.1007/s12083-011-0111-9.

Shang, M.-S. *et al.* (2009) 'Relevance is more significant than correlation: Information filtering on sparse data', *Europhysics Letters*, 88(6), p. 68008. doi: 10.1209/0295-5075/88/68008.

Shani, G. and Gunawardana, A. (2011) 'Evaluating recommendation systems', *Recommender systems handbook*, pp. 257–298. doi: 10.1007/978-0-387-85820-3_8.

Sherchan, W., Nepal, S. and Paris, C. (2013) 'A Survey of Trust in Social Networks', *ACM Comput.Surv.* New York, NY, USA: ACM, 45(4), p. 47:1-47:33. doi: 10.1145/2501654.2501661.

Sivapalan, S. *et al.* (2014) 'Recommender systems in e-commerce', in *2014 World Automation Congress (WAC)*. IEEE, pp. 179–184. doi: 10.1109/WAC.2014.6935763.

Smith, D., Menon, S. and Sivakumar, K. (2005) 'Online peer and editorial recommendations, trust, and choice in virtual markets', *Journal of Interactive Marketing*. Wiley Subscription Services, Inc., A Wiley Company, 19(3), pp. 15–37. Available at: http://dx.doi.org/10.1002/dir.20041.

Soboroff, I. and Nicholas, C. (1999) 'Combining content and collaboration in text filtering', *International Joint Conferences on Artificial Intelligence*, 99, pp. 86--91. doi: 10.3115/1118935.1118938.

Song, W., Phoha, V. V and Xu, X. (2004) 'The HMM-Based Model for Evaluating Recommender's Reputation', in *Proceedings of the E-Commerce Technology for Dynamic E-Business, IEEE International Conference*. Washington, DC, USA: IEEE Computer Society (CEC-EAST '04), pp. 209–215. doi: 10.1109/CEC-EAST.2004.64.

Symeonidis, P. and Tiakas, E. (2014) 'Transitive node similarity: Predicting and recommending links in signed social networks', *World Wide Web*, 17(4), pp. 743–776. doi: 10.1007/s11280-013-0228-2.

Tang, J. *et al.* (2012) 'eTrust: Understanding Trust Evolution in an Online World', in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM (KDD '12), pp. 253–261. doi: 10.1145/2339530.2339574.

Tang, J. *et al.* (2013) 'Exploiting homophily effect for trust prediction', *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 53–62. doi: 10.1145/2433396.2433405.

Tang, J. (2015) *Computing Distrust in Social Media*. Available at: http://repository.asu.edu/attachments/146459/content/Tang_asu_0010E_14639.pdf.

Ungar, L. H. and Foster, D. P. (1998) 'Clustering methods for collaborative filtering', *AAAI Workshop on Recommendation Systems*, pp. 114–129. doi: 10.1.1.33.4026.

Victor, P. *et al.* (2013) ‘Enhancing the Trust-Based Recommendation Process with Explicit Trust’, 7(2), pp. 1–19.

Victor, P., Cornelis, C. and DeCock, M. (2011) *Trust Networks for Recommender Systems*, *Trust Networks for Recommender Systems*. Edited by C. Cornelis, M. de Cock, and S. (Online Service). Atlantis Press. doi: 10.2991/978-94-91216-08-4.

Virzi, R. a. (1992) ‘Refining the test phase of usability evaluation: how many subjects is enough?’, *Human Factors*, 34(4), pp. 457–468. doi: 10.1177/001872089203400407.

Walter, F. E., Battiston, S. and Schweitzer, F. (2008) ‘A model of a trust-based recommendation system on a social network’, *Autonomous Agents and Multi-Agent Systems*. Springer Netherlands, 16(1), pp. 57–74. doi: 10.1007/s10458-007-9021-x.

Wan, Y.-H. and Chen, C. C. (2011) ‘An Effective Cold Start Recommendaton Method Using a Web Of Trust’, in Seddon, P. B. and Gregor, S. (eds) *Pacific Asia Conference on Information Systems, {PACIS} 2011: Quality Research in Pacific Asia, Brisbane, Queensland, Australia, 7-11 July 2011*. Queensland University of Technology, p. 205. Available at: http://aisel.aisnet.org/pacis2011/205.

Wang, J., Vries, A. P. De and Reinders, M. J. T. (2006) ‘Unifying User-based and Item-based Collaborative Filtering Approaches by Similarity Fusion Categories and Subject Descriptors’, *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 501–508. doi: 10.1145/1148170.1148257.

Watts, D. J. (1999) ‘Networks, Dynamics, and the Small‐World Phenomenon’, *American Journal of Sociology*, pp. 493–527. doi: 10.1086/210318.

Watts, D. J. and Strogatz, S. H. (1998) ‘Collective dynamics of “small-world” networks.’, *Nature*, 393(6684), pp. 440–442. doi: 10.1038/30918.

Wieringa, R. J. and Heerkens, J. M. G. (2006) ‘The methodological soundness of requirements engineering papers: A conceptual framework and two case studies’, *Requirements Engineering*, 11(4), pp. 295–307. doi: 10.1007/s00766-006-0037-6.

Wierzbicki, A. (2010) *Trust and fairness in open, distributed systems*, *Studies in Computational Intelligence*. doi: 10.1007/978-3-642-13451-7_1.

Wilks, Y. and Brewster, C. (2009) ‘Natural Language Processing as a Foundation of the Semantic Web’, *Found.Trends Web Sci.* Hanover, MA, USA: Now Publishers Inc, 1(3–4), pp. 199–327. doi: http://dx.doi.org/10.1561/1800000002.

De Wit, J. (2005) *Evaluating Recommender Systems*, *Thesis*. University of Twente. doi: 10.1109/AXMEDIS.2008.21.

Xue, G.-R. *et al.* (2005) ‘Scalable collaborative filtering using cluster-based smoothing’, *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’05*, p. 114. doi: 10.1145/1076034.1076056.

Yang, X., Steck, H. and Liu, Y. (2012) ‘Circle-based recommendation in online social networks’, in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD ’12)*. New York, NY, USA: ACM Press, pp. 1267–1275. Available at: http://dl.acm.org/citation.cfm?id=2339728 (Accessed: 25 September 2014).

Yu, B. and Singh, M. P. (2002) ‘An Evidential Model of Distributed Reputation Management’, in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 1*. Bologna, Italy: ACM (AAMAS ’02), pp. 294–301. doi: 10.1145/544741.544809.

Yu, K. *et al.* (2009) ‘Fast Nonparametric Matrix Factorization for Large-scale Collaborative Filtering Categories and Subject Descriptors’, *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 4, pp. 211–218. doi: 10.1145/1571941.1571979.

Yuan, W. *et al.* (2010) ‘iTARS: trust-aware recommender system using implicit trust networks’, *Communications, IET*. IET, 4(14), pp. 1709–1721.

Zafarani, R. and Liu, H. (2009) ‘Social Computing Data Repository at {ASU}’. Available at: http://socialcomputing.asu.edu.

Zarghami, A. *et al.* (2009) ‘Social Trust-Aware Recommendation System: A T-Index Approach’, *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. Ieee, pp. 85–90. doi: 10.1109/WI-IAT.2009.237.

Zhan, J. *et al.* (2010) ‘Privacy-preserving collaborative recommender systems’, *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, 40(4), pp. 472–476. doi: 10.1109/TSMCC.2010.2040275.

Zhang, F. (2009) ‘A Survey of Shilling Attacks in Collaborative Filtering Recommender Systems’, in *2009 International Conference on Computational Intelligence and Software Engineering*. IEEE, pp. 1–4. doi: 10.1109/CISE.2009.5365077.

Zhang, F., Bai, L. and Gao, F. (2009) ‘A User Trust-Based Collaborative Filtering Recommendation Algorithm’, in, pp. 411–424.

Zhang, H., Song, Y. and Song, H.-T. (2007) ‘Construction of Ontology-Based User Model for Web Personalization’, in *Proceedings of the 11th international conference on User Modeling*. Corfu, Greece: Springer-Verlag (UM ’07), pp. 67–76. doi: http://dx.doi.org/10.1007/978-3-540-73078-1_10.

Zhao, M. and Smith, S. W. (2006) ‘Modeling and Evaluation of Certification Path Discovery in the Emerging Global PKI’, *EuroPKI 2006*, pp. 16–30.

Zhou, X. *et al.* (2012) ‘The state-of-the-art in personalized recommender systems for social networking’, *Artificial Intelligence Review*. Springer Netherlands, 37(2), pp. 119–132. doi: 10.1007/s10462-011-9222-1.

Zhu, J., Hong, J. and Hughes, J. (2002) ‘Using Markov models for web site link prediction’, *Hypertext 2002*, p. 169. doi: 10.1145/513378.513381.

Ziegler (2004) ‘Semantic web recommender systems’, *Current Trends in Database TechnologyEDBT 2004 Workshops*, 3268, pp. 78–89. Available at: http://www.springerlink.com/index/BVUY163JEH4FH6KH.pdf.

Ziegler, C.-N. and Golbeck, J. A. (2007) ‘Investigating interactions of trust and interest similarity’, *Decision Support Systems*, 43(2), pp. 460–475. doi: 10.1016/j.dss.2006.11.003.

Ziegler, C.-N. and Golbeck, J. A. (2015) ‘Models for Trust Inference in Social Networks’, in

Dariusz, K., Fay, D., and Gabryś, B. (eds) *Propagation Phenomena in Real World Networks*. Springer International Publishing, pp. 53–89. doi: 10.1007/978-3-319-15916-4_3.

Ziegler, C.-N. and Lausen, G. (2004) 'Spreading activation models for trust propagation', in *IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004. EEE '04. 2004*. IEEE, pp. 83–97. doi: 10.1109/EEE.2004.1287293.

Ziegler, C.-N. and Lausen, G. (2005) 'Propagation Models for Trust and Distrust in Social Networks', *Information Systems Frontiers*, 7(4–5), pp. 337–358. doi: 10.1007/s10796-005-4807-3.

Zigomitros, A., Papageorgiou, A. and Patsakis, C. (2016) 'A practical k-anonymous recommender system', in *IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications*. doi: 10.1109/IISA.2016.7785379.

Zolfaghar, K. and Aghaie, A. (2012) 'A syntactical approach for interpersonal trust prediction in social web applications: Combining contextual and structural data', *Knowledge-Based Systems*. Elsevier B.V., 26, pp. 93–102. doi: 10.1016/j.knosys.2010.10.007.