RECOMMENDER SYSTEMS BASED ON ONLINE SOCIAL NETWORKS - AN IMPLICIT SOCIAL TRUST AND SENTIMENT ANALYSIS APPROACH

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
IN THE FACULTY OF SCIENCE AND ENGINEERING

2017

By
Dimah Hussain Alahamdi
School of Computer Science

Contents

Word Count: 45586

List of Tables

List of Figures

Abbreviations

RSs	Recommender Systems
OSNs	Online Social Networks
CB	Content-Based recommender
CF	Collaborative-Based recommender
TF-IDF	Term Frequency/Inverse Document Frequency
PCC	Pearson Correlation Coefficient
MF	Matrix Factorisation
STS	Social Tagging System
WOT	Web of Trust
SVR	support vector regression
ISTS	Implicit Social Trust and Sentiment based approach to recommender system
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
TIE	Twitter Interaction Extractor
$H-ISTS$	Hybrid implicit social trust and sentiment based recommender
$H-ISTS_{CF}$	the integrated of CF recommender with $H-ISTS$
GA	Genetic Algorithm
ILS	Intra-List Similarity

Notations

u_i	user u who rated item i
F	group of friends
f	a friend in the group F
u	the active user who needs recommendations
RT	re-tweeting action
$RT_{u,f}$	the number of re-tweets actions between f and u
$RT_{u,F}$	the number of re-tweets actions between f and u
$trust_{u,f}$	trust metric describes the trust level between user u and a friend f
$TRUST_{u,f}(T)$	trust metric between user u and friend f in a period T
L	list of friends
MR	a set of micro-reviews
SR	the sentiment ratings extracted from friends' micro-reviews
wc	class of ratings that a sentiment word belongs to
C	set of the used ratings class
TF	group of trusted friends
R	predicted matrix
$R_{u,i}$	the predicted rating for missing item i that the active user u needs
pr_u	the profile related to user u includes items ratings.

Abstract

RECOMMENDER SYSTEMS BASED ON ONLINE SOCIAL NETWORKS - AN IMPLICIT SOCIAL TRUST AND SENTIMENT ANALYSIS APPROACH
Dimah Hussain Alahamdi
A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy, 2017

Recommender systems (RSs) provide personalised suggestions of information or products relevant to user's needs. RSs are considered as powerful tools that help users to find interesting items matching their own taste. Although RSs have made substantial progress in theory and algorithm development and have achieved many commercial successes, how to utilise the widely available information on Online Social Networks (OSNs) has largely been overlooked. Noticing this gap in existing research on RSs and taking into account a user's selection being greatly influenced by his/her trusted friends and their opinions, this thesis proposes a novel personalised Recommender System framework, so-called Implicit Social Trust and Sentiment (ISTS) based RSs. The main motivation was to overcome the overlooked use of OSNs in Recommender Systems and to utilises the widely available information from such networks. This work also designs solutions to a number of challenges inherent to the RSs domain, such as accuracy, cold-start, diversity and coverage.

ISTS improves the existing recommendation approaches by exploring a new source of data from friends' short posts in microbloggings. In the case of new users who have no previous preferences, ISTS maps the suggested recommendations into numerical rating scales by applying the three main components. The first component is measuring the implicit trust between friends based on their intercommunication activities and behaviour. Owing to the need to adapt friends' opinions, the implicit social trust model is designed to include the trusted friends and give them the highest weight of contribution in recommendation encounter. The second component is inferring the sentiment rating to reflect the knowledge behind friends' short posts, so-called micro-reviews.

The sentiment behind *micro-reviews* is extracted using Sentiment Analysis (SA) techniques. To achieve the best sentiment representation, our approach considers the special natural environment in OSNs brief posts. Two Sentiment Analysis methodologies are used: a bag of words method and a probabilistic method. The third ISTS component is identifying the impact degree of friends' sentiments and their level of trust by using machine learning algorithms. Two types of machine learning algorithms are used: classification models and regressions models. The classification models include Naive Bayes, Logistic Regression and Decision Trees. Among the three classifications models, Decision Trees show the best Mean absolute error (MAE) at 0.836. Support Vector Regression performed the best among all models at 0.45 of MAE.

This thesis also proposes an approach with further improvement over ISTS, namely Hybrid Implicit Social Trust and Sentiment (H-ISTS). The enhanced approach applies improvements by optimising trust parameters to identify the impact of the features (re-tweets and followings/followers list) on recommendation results. Unlike the ISTS which allocates equal weight to trust features, H-ISTS provides different weights to determine the different effects of the two trust features. As a result, we found that H-ISTS improved the MAE to be 0.42 which is based on Support Vector Regression. Further, it increases the number of trust features from two to five features in order to include the influence of these features in rating predictions. The integration of the new approach H-ISTS with a Collaborative Filtering recommender system, in particular memory-based, is investigated next. Therefore, existing users with a history of ratings can receive recommendations by fusing their own tastes and their friends' preferences using the two type of memory-based methods: user-based and item-based. $\text{H-ISTS}_{\text{item}}$ is the integration of H-ISTS and item-based which provides the lowest error at 0.7091. The experiments show that diversity is better achieved using the $\text{H-ISTS}_{\text{user}}$ which is the integration of H-ISTS and user-based technique.

To evaluate the performance of these approaches, two real social datasets are collected from *Twitter*. To verify the proposed framework, the experiments are conducted and the results are compared against the most relevant baselines which confirmed that RSs have been successfully improved using OSNs. These enhancements demonstrate the effectiveness and promises of the proposed approach in RSs.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

First of all, I would like to thank and express my gratitude to Allah for all His blessings. I would not have completed my PhD without his will and guidance.

I would like to express my sincerest gratitude to my supervisor Dr.Xiao-Jun Zeng for his support, encouragement, and guidance to the right direction during this PhD research. His helpful discussions, useful suggestions, and comments on my work were valuable, from the starting date until submission.

I would like to acknowledge The University of King Abdulaziz for giving me the opportunity to complete my PhD studies and funding my research. I would also like to thank Saudi Ministry of Higher Education for their services and support. I would also acknowledges the University of Manchester and the entire staff in the School of Computer Science for all the help and support they provided during my PhD research.

My deepest gratitude and appreciation also go to my parents and my husband for their help and support in difficult times. Without your support, this PhD would have been much more difficult to complete.

Finally, I would wish to express my immense love to my children without whose motivation and love I would not have been able to achieve anything.

Chapter 1

Introduction

1.1 Recommender Systems Context and Motivation

The most important features of the current Online Social Networks (OSNs) can be presented in two main capabilities. The first is the free generated content produced by different users and the second is the advanced technological environment it works within. Users can easily approach the websites and share their opinions in available OSNs, such as Facebook, wikis and Twitter. Consequently, people face an exponential growth of information which is problematic for two reasons. The first problem is related to individuals and the difficulty they face in finding content that is relevant to their own interest amongst a vast amount of alternatives. The second is related to online businesses and the heavy demand they face to provide personalised product recommendations in the current environment of the World Wide Web.

Recommender systems (RSs) are defined as effective tools that provide personalised suggestions of information or products relevant to users needs. In fact, RSs have given privileges to both customers and online businesses in providing product recommendations when using platforms, such as Amazon, Netflix and Ebay. Traditionally, RSs can be categorised into two main techniques: collaborative filtering (CF) recommenders and content-cased (CB) recommenders [5, 6]. CF recommenders make use of users' rating information history and analyse the similarity between user's rating profiles to those of their *neighbours*. For users who have a high similarity in their history, there is a potential to share the same interests in the future based on the profiles shared between them [7, 8].

CB recommenders identify features that appear in item contents that users have experienced before, thereafter suggesting more items which contain these relevant features to the users. CB techniques are widely applied on textual component items where the system can derive the most important features/keywords included in a purchased book or an article and then provide more books/articles that contain similar keywords. These features and keywords used to build user-item profiles [9, 10]. In the past few years, recommendation researchers incorporated users' trust relations to improve traditional RSs by suggesting further products based on reliable connections [3, 11, 12, 13]. Recently, several approaches have been proposed to enrich recommendations based on online product reviews. These approaches utilise natural language processing techniques to identify the sentiment in reviews that users have written about products [14, 15, 16]. Therefore, in this thesis, it is suggested that it is more cost effective to consider implicit trust and sentiment from product reviews in an integrated manner when designing RSs as they are approached differently from those in the literature.

In order to develop approaches that utilise OSNs and underlying relations, these aforementioned approaches still have several serious drawbacks that need to be addressed as follows.

- In traditional recommendation techniques the ratings information used to build user-item profiles usually suffer from *sparsity* due to the fact that people's rating behaviour is very limited to a small portion of available items, only 1% of products in systems, and this affects the effectiveness of CF and CB algorithms as pointed out in [8]. As a result new users experience the so-called *cold-start* problem as traditional CF and CB algorithms provide less powerful recommendations for those who have not had any previous choices and ratings [5]. New source of data could be investigated to boost recommendations for new users based on their social connections on OSNs.

- Social trust in trust-based RSs assumes that users in systems are friends and also rate items. Although on the OSNs such as Facebook and Twitter people cooperate with their friends, relatives, and colleague in the real world, this aspect of social trust is not true in many cases in the current trust-based RSs. In all, the current individual online relations in RSs are not an accurate reflection of social friendships [17].

- Deriving users' opinions from written reviews about commercial products has not been fully applied using OSNs to harness more personalised suggestions. Friends' comments though hold many of opinions and interests about a diverse range of items and services, such as books, movies and restaurants, etc. Further, due to the inherent shortness of online comments and limited use of characters on OSNs environment, methods such as term frequency, used to index user's profiles in long standard reviews might not be the most appropriate algorithms. For example, review sites about restaurants include many headers to guide the customer how they write reviews such as: location, flavour, etc. This gives a more structured long text that allows customers to describe their opinions in detail. Moreover, in this thesis, we consider the informal use of language features on OSNs , such as emoticon signs as we believe that they hold a clear picture of emotions and opinions in the text. For example, in the tweet: *Jumanji ;) watching it since long time :D*, the signs *;)* and *:D* are indicative of highly positive opinion about the mentioned movie *Jumanji*. It is hoped that, in this way, we can comprehensively represent opinions in such an environment. In addition, it is more effective in RSs to measure the degree of sentiments and opinions with a multi-point scale of ratings than with the traditional classification to negative and positive polarities [18]. It is worth mentioning that studies in the domain of products reviews-based RSs differ from our work in that we propose to measure the sentiment in a multi-point scale, in brief and informal language features, and this is one of the challenging aspects raised in this thesis.

In this thesis, we focus on addressing the above weaknesses and model a recommender that can employ user's OSNs to derive the users' preferences even in the case where no rating information history is available.

1.2 Research Questions

The literature surrounding the RSs contains different types of proposed solutions. As we described in the previous section, most of these solutions have been designed based on ratings information. Such a solution can be the integration of users' social information existing on OSNs, upon which this thesis focuses. However, to date, there has been no work which aims to boost recommendations based on friends' opinions. Several research questions are therefore presented:

The first Question: *(1) How can we derive recommendations from OSNs about particular items?* In order to address this need, we wish to provide evidence for the availability of such information and implement suitable tools to collect them.

After retrieving the desired information, we need to know how friends' opinions can be utilised. This is a challenging task because in order to achieve this, we have to answer the followings questions. The second Question: *(2) How can the trust model be built to select friends that can provide recommendations?* Answering this question includes the identifications of trust features and criteria to construct the trust component that we applied in our approach to filter friends. The third Question: *(3) What is the most effective method to derive the correct sentiment describing friends' opinions from their posts?* In order to answer this question, we have to consider the environment nature of OSNs posts.

Once we have understood how the integration of OSNs in RSs can be performed, we need to answer the fourth Question: *(4) How can an effective method be designed to predict ratings for new users?* Attempting to answer this question has led to the investigation of machine learning methods to predict ratings of unseen items.

Then we wish to study the link between ordinary users who have a history of preferences in the system within their OSNs. Hence, we pose the fifth Question: *(5) What will the collaborative filtering performance be when we fuse ordinary users' preferences with friends' preferences derived from OSNs?* To this end, we ask one final research Question: *(6) How may the new approach contribute to the diversity challenge in CF users' profiles?*

1.3 Research Aim and Objectives

In this study, we challenge the assumption that friends' opinions have a huge impact on users' choices and interests even if they have different tastes, and argue that information on OSNs, microblogs in particular, can be considered as a new rich source of data that can be used to personalise recommendations. More precisely, our proposed approach aims to design a RS framework based on OSNs to allow users to receive personalised suggestions of items using two components:

- The friends' sentiment ratings that express friends' opinions about items are obtained from their short text posts, for example a post about a movie item (e.g.,*Jumanji ;) watching it since long time :D*). We try to estimate a rating that reflects the friend's opinion toward the movie, for example the estimated rating could be 4.5 in a rating scale of 5.

- The trust between an active user and his/her friends on OSNs based on the intercommunication between them such as re-tweet actions and followings/followers list. For example when an active user re-tweeted one friend more frequently than others, this friend could be considered as a trusted source to the active user in order to receive recommendations.

To the best of our knowledge, this is the first work of its kind which employs sentiment and trust from microblogs to generate personalised recommendations. More specifically, in order to achieve this aim we need to satisfy the following objectives:

1. To design and develop a new research framework of RSs based on OSNs. To achieve this objective, we specify and analyse the limitations and challenging issues in the existing recommendation literature in order to select and design the required features.

2. To develop a tool to collect a convenient datasets from OSNs that satisfy the requirements of our approach.

3. To develop trust and sentiment methodologies by utilising online social network data to improve and enhance the capability and quality of recommender systems as follows:

    - To identify a way to model sentiment ratings in short posts that can cope with the challenging environment of microblogs, taking into consideration that these posts are short and include informal use of language.
    - To infer multiple score ratings from friends' posts in microblogs and use these inferred sentiment ratings to represent friends' opinions.
    - To develop a realistic trust model based on intercommunication on OSNs.

4. To build efficient recommendation models using machine learning algorithms utilising the encountered trust and inferred sentiment ratings to predict items ratings.

5. To investigate the effect of the new research approach in a collaborative filtering systems on users who have previous preferences.

6. To investigate and analyse the impact of our approach on the diversity challenge in collaborative filtering recommenders.

7. To evaluate and analyse the performance of the research approach.

1.4 Research Contributions

The novelty of this research is that the research approach utilises trust and sentiment in providing recommendation. Furthermore, the developed recommendation framework does not require any previous ratings history or even any written reviews by users. Based on this study, our research contributions can be summarised as follows:

1. The design, development and evaluation of the novel research framework, **I**mplicit **S**ocial **T**rust and **S**entiment (ISTS) based RSs. Proposing the idea of providing recommendations based on OSNs as a new source of data in order to improve the performance of recommenders (Chapter 3). By contrast, the main data source in literature on recommenders are either items ratings feedback or trust ratings towards others (Chapter 3).

2. The proposal and development of methods that identify the trust relationship (Chapters 4 and 5).

3. The design and development of a sentiment analysis technique that can fit the domain of microblog in OSNs. In addition, rather than the polarities of negative and positive opinions in short posts we propose to express opinions in scalar ratings using two methodologies (Chapter 4):

   - A bag of words method to indicate the sentiment included in online posts.
   - A probabilistic method to support the illustration of the intensity of sentiment in posts into numerical rating scales.

4. The design, development and performance evaluation of machine learning methods to identify the most accurate prediction model of items ratings based on trust in friends and sentiment of their text posts, particularly for cold-start cases (Chapter 4).

   - Investigating and evaluating classification algorithms for rating predictions.
   - Improving and evaluating ratings prediction by using regression algorithms in order to adapt relative semantics between rating scores.

5. The design, development and evaluation of a novel framework Hybrid Implicit Social Trust and Sentiment (H-ISTS) based RSs. Although ISTS presents significant performance, the new H-ISTS approach can produce some enhancements in terms of qualifying and quantifying the trust model.

6. Analysis and evaluation of the efficiency and effectiveness of H-ISTS integrated to a collaborative filtering recommender (Chapter 6).

7. Investigation and analysis of the impact of the proposed approach in solving the diversity challenge and expanding users' profiles (Chapter 6).

Thorough description of these contributions will be provided in the conclusion of the thesis (Chapter 7).

1.5 Research Publications

The work presented in this thesis has resulted in several publications:

- D. H. Alahmadi and X.-J. Zeng, ISTS: Implicit social trust and sentiment based approach to recommender systems, Expert Systems with Applications, Vol. 42, no. 22, pp. 8840-8849, 2015.

- D. Alahmadi and X.-J. Zeng, Improving recommendation using trust and sentiment inference from OSNs, International Journal of Knowledge Engineering, Vol. 1, no. 1, pp. 9-17, 2015.

- Best research prize for the presentation titled Improving recommendation using trust and sentiment inference from OSNs, in the conference: International Conference of Knowledge, ICK, 2015.

- D. H. Alahmadi and X.-J. Zeng, Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis, in Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on, pp. 1045-1052, IEEE, 2015.

- D. H. Alahmadi and X.-J. Zeng, Enhancing collaborative recommendation performance by combing users' preferences and friends on microblogs, Knowledge-Based Systems, 2016 (in Review).

1.6 Thesis Structure

The structure of this thesis is organised as follows. In Chapter 2, we present the background of recommender systems. We cover the literature in the domain of recommendation and the existing approaches and solutions. Discussing gaps in the existing research in RSs has led to designing a set of desired features of RSs. A critical analysis of the current literature against the designed features is therefore presented in this chapter.

From the discussion of existing literature, Chapter 3 introduces the main ideas of the approach proposed by the current research. It starts by highlighting the possible recommendation scenarios. It then provides a description of the design of ISTS and its components.

In Chapter 4, we discuss the trust features that will be implemented in the ISTS. Furthermore, the chapter describes the sentiment analysis techniques that are developed for extracting opinions from short posts. Machine learning algorithms are used to predict ratings and deliver the correct recommendations are also discussed and evaluated.

In Chapter 5, the trust model is optimised by tuning trust features parameters. A genetic algorithm is used for the optimisation task. Extended trust features are described, analysed and evaluated.

In Chapter 6, we study the effect of ISTS integrated with the collaborative filtering framework. We investigate the impact of ISTS on the diversity challenge in users' profiles.

In Chapter 7, we conclude the thesis by reviewing the main findings that contribute to the field of recommender systems and suggest several avenues for future work.

Chapter 2

Background and Literature Review

2.1 Chapter Introduction

In this chapter, a background and related literature of Recommender Systems research are surveyed. In detail, Section 2.2 defines recommender systems and describes their importance for users and businesses. Section 2.3 describes the main elements used in recommendation process. Section 2.4 presents a review of the main existing recommendation methods and reports the advantages and issues associated with each method. Section 2.5 introduces the use of the trust perspective in recommendation research. Section 2.6 presents recommendation approaches based on item reviews. Section 2.7 analyses and summarises the required features to design a successful recommendation solution against the missing features in the surveyed related literature. Section 2.8 introduces our vision of the way forward to address the missing parts in the existing proposed recommendation solutions. Finally, Section 2.9 provides a conclusion for the chapter .

2.2 Recommender Systems

Recommender Systems (RSs) are powerful tools and techniques that process the resources of online businesses to generate useful suggestions of items that match users' interests and needs. Items/products can be any of the resources of businesses that are consumed by users online, such as books, CDs or news, etc. RSs produce certain decisions to make suggestions of which items to purchase or what news to read online.

Generally speaking, the mechanism of a recommender system is based on the analysis of historical data about each user's feedback about items used previously in order to recommend a list of estimated preferred items that the user may like in his next purchasing operations. Users are overwhelmed by a huge number of choices of items that websites offer; hence, RSs serve to filter these alternatives in a personalised manner and recommend the items that will most likely attract user's consideration.

Recommender systems have appeared as an independent and important research area since the publication of the first paper on recommendation methods in the mid-1990s [7]. In the past decade, retailers have integrated the service of providing personalised recommendations as an important part of their systems. These services include the popular e-commerce retailers, Amazon.com and eBay (ebay.com), as well as the movie and entertainment industry, MovieLens (movielens.org), Netflix (netflix.com), FilmTrust (trust.mindswap.org/FilmTrust), and the music industry, such as CDNOW (cdnow.com), Ringo (ringo.com) [19], LastFm (last.fm) and Pandora (pandora.com), in addition to pictures and photo sharing systems, such as (flickr.com), expertise finder systems, such as Linkedin (linkedin.com), and news recommendation sites, such as Google news [20].

Nowadays, modern e-commerce systems highly require recommender systems as a useful tool to increase their profit, similar to the model followed by Amazon.com. The next section outlines business motivations that encourage commercial platforms to exploit recommender systems.

2.2.1 Importance for Online Businesses

Online businesses may utilise recommendation techniques for the following reasons [6]:

- *To increase the number of items sold.* This is probably the ultimate function of RSs that attracts online businesses. Commercial websites that are supported by RSs are able to sell a relatively larger set of products compared to those that do not employ any recommendation techniques [21]. Even for non-profit applications, these have same interest in increasing the number of items experienced and consumed by users. For example, using recommendation techniques in scientific journal websites aim to increase the number of articles read on the site.

- *To sell a more diverse range of items.* Another importance function of a RS is to allow the user to explore new items he/she may like. For instance, movie rental systems may rent the most popular movies easily, but with RSs techniques, less popular movies will also be suggested to the right users when this is likely to suit their own tastes.

- *To increase user satisfaction*. Systems that include a combination of (1) a well designed friendly user interface and (2) an interesting recommendations output, are more likely to improve the user's subjective judgement of the system function.

- *To increase user fidelity*. The more frequently the user visits the commercial website, the more closely the RS will understand the user's preferences and tastes. Since RSs analyse the interactions between the user and the system as the user's profile becomes more refined, consequently, long-term users who would have visited the website regularly are treated as valuable customers served according to their needs and tastes, and this in turn increases user fidelity.

- *To better understand what the user wants*. By collecting user information and feedback logs, this makes the service provider able to re-use this knowledge for other different goals such as enhancing production criteria to match users' tastes or improving the management of items stock accordingly. For instance, in the travel sector, analysing the data collected by the RS helps the travel agents decide to have a particular promotion to new destinations based on information derived from customer choices.

However, users also may utilise RSs to achieve a different set of goals as explained next.

2.2.2 Importance for Users

Herlocker et al. [22] explained that users can also benefit from different functions when a RS is implemented. They reported serval functions of RS as follows:

- *To find certain items*. The core function of RS is to offer set of recommendations that suit the user's needs. Providing the user with a ranked list associated with an estimation of how much the user would be interested in them in the style of a rating scale. However, some RSs choose to hide the predicted rating of items.

- *To find all good items*. In medical and financial applications, these domain are *mission-critical* with a small number of items. The user should benefit from all well examined possibilities. Moreover, providing users with ranked items and stating additional explanations may also be required in such situations.

- *Annotation in context*. Providing a list of items with emphasis on certain items based on information derived from long-term preference profiles is another option that provides useful annotations to customers.

- *Recommending a sequence*. The idea behind this task is to provide a sequence of items matching a user's interests instead of one recommendation. For example focusing on generating series of books on a certain topic, which a user has experienced before.

- *Recommending a bundle*. The customer receives a set of items that fits well together. For instance, when providing users with a travel plan, the recommendations may encompass several attractions, restaurants, destinations, and accommodation services or even some special children activities. From a customer's point of view, these alternatives should be presented as a single travel suggestion.

- *Just browsing*. Some users' ultimate goal is to browse the catalogue without any intention of purchasing items. Hence, the recommender helps in allocating items relevant to the scope of the user's search.

- *Finding a credible recommender*. Some customers do not believe in recommender systems; however, they try to test these systems to assess their ability to generate credible recommendations. Hence, some systems offer this service to allow the users to verify their performance.

- *Improving customer profiles*. In order to achieve the goal of personalisation, users feed the systems with information about their likes and dislikes. Otherwise, some RSs provide active user with what delivered to average users.

- *Self-Expressing*. Some users intend to contribute with their ratings only to express their opinions and beliefs. They, however, do not pay attention to the recommendations.

- *Helping others*. The key motivation for some users is to help other customers with their experience and benefit the community with their contributions. For example, in a car retail RS, a user who has bought a car knows that ratings will support the plans of other buyers rather than their own for the next time he/she buys a car.

- *Influencing others*. Some users have the ultimate goal of influencing others' opinion towards buying certain items they love. On the other hand, some RSs suffer from malicious users that may exploit the RS just to promote or even penalise particular items.

As a matter of fact, a well designed RS must maintain a balance between the needs of users and businesses while trying to satisfy both. The section below discusses the challenge associated with formalising the recommender system problem.

2.3 Formalising RSs Problem

A Recommender System usually has three input elements: (1) users, (2) items, and (3) feedback, where users provide their feedback about items. Feedback can be in the form of ratings, which is the most common way in which users describe their opinions about items. Let us assume that users of a system are denoted by set U and user $u \in U$, and N is the total number of users in the set U. Let the set of items I where item $i \in I$, and M is the total number of items in the set I. We denote the item that is rated by one user using this notation i_u, and the user who rated an item as u_i. Every user and item are represented in the systems by a unique index to identify them clearly in the system.

The system's input elements are then transferred to a user-item ratings matrix, which has the space of $N \times M$. We assume that g is the utility function that computes the level of interest of user set U in item set I:

$$g : U \times I \rightarrow \hat{R}. \tag{2.1}$$

where $\hat{R}$ is a recommended set of items in a specified numerical range. More specifically, we move now to find the set of items $i \in I$ that maximises the user's utility for each user $u \in U$ [5], as follows:

$$i_u = argMax_{i \in I} g(i,u), \qquad \forall u \in U \tag{2.2}$$

Most recommender systems' core engines try to extrapolate the utility g to build a model that can estimate a user's judgment towards unknown items using already known feedback on items' used before according to certain criteria to test the performance of RS, such as measuring the error in the estimated ratings. After the model has been built, unknown ratings can be estimated and provided to users in an ordered set. Researchers use machine learning algorithms, approximation theory, and some heuristics for the prediction step. Next, we discuss how users' profiles and items' profiles can be built and we also describe the types of feedback that can be considered within recommender systems.

2.3.1 Feedback Acquisition in RSs

Recommender Systems require extracting information about users. Figure 2.1 shows the different possible actions and feedback from users towards the items (e.g., articles, products, videos, etc.) in the system. Metadata stand for all types of feedback supplied by the user. Either direct or implicit feedback require extended analysis to extract information that help in qualifying an item [1].

Normally, RSs store users' history, their transactions with the system and feedback about items in order to extract enough information on users' interests. For instance, a transaction log can record a reference and a description of the item that was used by certain users, including purchase history, ratings, product reviews, etc. In fact, the most widely used transaction data collected by RSs are ratings. Rating as an evaluation of an item given by the user can be in two forms, explicit and implicit. In the explicit form, the system asks the user to evaluate an item on a rating scale and to express their opinions about it. However, implicit rating is used in cases when the system infers these ratings. Therefore, computing rating scores to represent quality can facilitate judging users evaluation of items in a simple manner. According to [22], ratings can be represented in different style and scales, as described below:

- Numerical rating, such as as a score the form of from 1 to 5 in the form of stars.
- Ordinal ratings, for example (strongly like, like, neutral, dislike, strongly dislike), where the system asks the user to select which option closely reflects his/her opinion from these terms.
- Binary ratings where the system asks the user to indicate if a particular item is good or bad.
- Unary ratings to find out if a user purchased an item or rated the item positively.

More formally, in order to build users' profiles pr_u as $PR = \{pr_{u_1}, pr_{u_2}, ..., pr_{u_N}\}$ in the system, users are asked to rate items to allow the recommender engine to recommend new items to an active user. Every profile of a user u includes a set of ratings

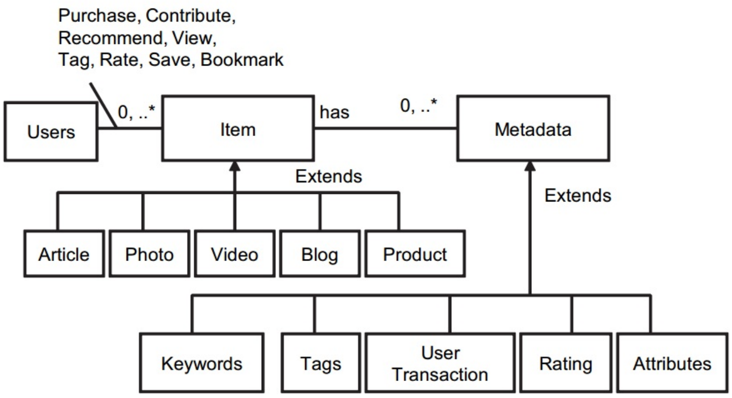


Figure 2.1: A user's different interactions with items on the system [1]

r toward items i denoted as $r_{u,i}$, hence $pr_u = \{r_{i,u} | i, u \subset U \times I\}$. Typically, the total space of the user-item ratings matrix R is $S = N \times M$, and usually $PR \subset S$.

Ratings information is an example of explicit feedback which can be used directly in the system. On the other side, implicit feedback can also be extracted from the system and can take different forms; for example, writing reviews and comments about the used items, clicks on certain items, the time spent on a certain web page, purchased and returned items, the watch list on sites such as YouTube, editing a wish list on retailer websites, and finally, tagging items. Using implicit feedback require the system to analyse the users' behaviour and observe the history of interactions. Both implicit and explicit feedback can reflect a range of different opinions either positive or negative. However, implicit information is noisy and might cause many error, although they afford the system the chance to collect a large set of training data. By contrast, explicit information is noise free although users tend to rate only a small number of items, which leads to a limited amount of data. Indeed, a method is required to extract users' opinions from reviews and text analysis steps should be applied to ensure the correct qualification of an item [1]. This is explored in more detail in Chapter 4.

Nowadays, modern and efficient recommender systems are required to intelligently balance the use of both types of user's feedback. To develop a successful Recommender System, the design of a suitable method to implement both implicit and explicit information must be taken in account, whether these are used separately or in an integrated/combined manner to arrive to the correct recommendations; however, this point, is beyond the scope of this thesis.

Recommender systems require to perform tasks using a range of different techniques and methods to achieve their different goals. The next section will discuss the classification of RSs techniques.

2.4 Classification of Recommendation Methods

Recommendation problems are addressed in the literature under two main categories: content-based recommendation, discussed in Section 2.4.1, and collaborative filtering recommendation, explored in Section 2.4.2. A third subcategory, Hybrid recommenders, is presented in Section 2.4.3.

2.4.1 Content-Based Recommendation

Content-based (CB) recommendation methods use the description of items to build item representations and user profiles. CB methods analyse the content/description in user profiles based on the items he/she liked before, then they match up this content with similar content of unseen items. In other words, these methods generate recommendations by analysing the features of items, such as the item textual descriptive information, and testing regularities that may occur in this content in order to recommend similar items in the future.

Many current CB recommenders build recommendations on items that contain textual information, including websites (URLs), books, documents, news, etc. For example, in a book recommendation application, in order to suggest books to a particular user, the CB recommender tries to analyse the regularities among the books that the active user has rated highly in the past (specific authors, types, subject matters, etc.). Then, only books that have high degree of similarity to what the user preferred would be suggested. The CB approach has its origins from information retrieval [23]. One step which has been enhanced over traditional information retrieval approaches is the use of user *profiles* that contain information about items a user has preferred and liked. The user's profile can be built by observing the users' transaction behaviour related to certain items [5].

More formally, let pr_i be an item profile, i.e., a set of characteristics of an item i described in pr_i. This set of features is extracted to be incorporated in the recommendations process as appropriate. As mentioned before, content-based systems are designed to ultimately recommend text-based items; the features in these systems are usually described in *keywords*. For example, if a system recommends web pages to users, it may represent the web pages content with the most important keywords. The importance of word k in document d is defined as a weight $w_{d,k}$ that can be measured by several methods.

One of the widely used measures for assigning keyword weights is *Term Frequency/Inverse Document Frequency* (TF-IDF) [5]. Assuming that $freq_{k,d}$ is the number of times a keyword k_i occurs in document d_j, then the frequency term $TF_{k,d}$, the term frequency or the normalised frequency of keyword k_i in document d_j will be as follows:

$$TF_{k,d} = \frac{freq_{k,d}}{maxfreq_{K,d}} \tag{2.3}$$

where the $maxfreq_{K,d}$ is calculated over all the frequencies in d of all keywords K. In fact, term frequency (TF) in many documents tends not to be a good indication of relevant documents and non-relevant ones as it treats all words equally in term of importance. It is obvious that some keywords appear more commonly in certain topics; for example, the word *programme* is included in almost every document about programming, so this word has a little effect when determining relevance of certain documents. Therefore, the measure of inverse document frequency (IDF) is often used together with simple term frequency to reduce the effect of the frequency of terms which have less importance when computing the weights. The inverse frequency of keyword k_i in a document will be:

$$IDF_k = log\frac{N}{n_k} \tag{2.4}$$

Where N is the total number of documents that can be provided to a user as a recommendation while n_k is a subset of N where the keyword k appears. Therefore, to compute the weight for keyword k_i in document d_j, the calculation should combine both $TF_{k,d}$ and IDF_k as follows:

$$w_{k,d} = TF_{k,d} \times IDF_k \tag{2.5}$$

Then, the profile of document d_j is defined as:

$$Pr_d = (w_1, w_2, \cdots, w_K) \tag{2.6}$$

As mentioned earlier, recommenders which use the CB approach suggest items similar to those that a user liked previously. Hence, the level of similarity is computed between two types of profiles; the first, item profiles which contain item features as weights of keywords, and the second, user profiles which contain weights of keywords of items seen or rated by the user in the past. For instance, if a user has read many online articles on the topic of *networks*, then the CB techniques will be able to provide other articles related to *networks* to that user. This is the case because these articles will contain terms related to *networks* (e.g, router, protocol, and ”wireless”) as opposed to articles on other topics. Therefore, recommenders use similarity measures to identify higher similarity values between a user's profile and those articles profiles which have network terms with higher weights.

To assign similarity weights between users' profiles and items' profiles, CB recommenders apply similarity techniques, such as a vector cosine similarity measure, defined as follows:

$$sim(u,d) = cos(\vec{w}_u, \vec{w}_d) = \frac{\vec{w}_u . \vec{w}_d}{||\vec{w}_u||_2 . ||\vec{w}_d||_2} \tag{2.7}$$

In vector cosine similarity algorithm, the profile of user u and document d will be treated as two vectors $\vec{w}_u$ and $\vec{w}_d$. Then, in order to compute similarity between the two vectors the method will measure the cosine of the angle between them.

Several techniques are based on a *model* learned from data related to users and items rather than the use of heuristic approaches. Various machine learning algorithms are used for CB recommenders, such as classification and clustering algorithms presented in [24, 25]. For example, the authors of [9] rated a set of web pages in two categorises: *relevant* and *irrelevant* by the user. They used users' profiles to learn their interest in different pages by determining 128 informative words as features. They then used a Naive Bayesian (NB) classifier to classify unrated web pages by estimating the probability that a page p_j belongs to a particular class C_i (relevant or irrelevant) given the set of keywords $k_{1,j},...,k_{n,j}$ using the probability function $P(C_i|k_{1,j}\&...\&k_{n,j})$. In order to overcome the drawback of poor prediction in the case when some users are unwilling to rate many pages, some initial knowledge about the user's interests can be used.

In general, we can summarise the CB recommendation main steps as follows:

- Building items' profiles: this is carried out by analysing the content and includes pre-processing steps to extract features; for example, converting web pages original text representation into keyword vectors.
- Building users' profiles: this step analyses user's history of rated items and

builds the user's profile based upon his/her previous interests.

- Measuring similarity: the system measures the similarity between an item's and user's profile to indicate the weight of the importance of an item to an active user.

- Filtering recommendations: the system provides the active user with a list of the most relevant unseen items with predicted binary rating.

The next section highlights the challenging issues in CB recommendation methods.

2.4.1.1 Challenges in CB Recommendation

CB techniques have many advantages: first, these methods are based only on users' preferences presented in ratings to build users' profiles and does not require input from other users' ratings to build these profiles. Second, the explanation advantage allows the system to exploit the list of *keywords* to justify why a specific item has been recommended. In addition, systems based on CB are able to recommend *new items* which have not received any ratings yet, a feature which is not applicable to other techniques. Despite all these advantages, CB techniques still exhibit several limitations [5, 26]:

- **Limited content analysis**: Automatic extracting of features to describe items in a system will be much harder to implement when the data are graphical images, audio and video streams. Information retrieval techniques achieve good results in extracting features from text documents; however, when describing documents by their most important keywords, CB techniques cannot distinguish between high quality articles and badly-written ones when the same keywords are used [19].

- **Overspecialisation**: The recommended items will be limited to what the user has rated in the past. In other terms, only items matching the user's built profile of preferences with a high score of similarity will be recommended. For example, using a recommender system for restaurants, a user with no past experience of *Mexican* cuisine would never receive a recommendation for even the greatest Mexican restaurant in the city. Moreover, this overspecialisation drawback is not only seen with CB recommenders in cases where they cannot recommend items that are different from what the user has experienced before, but in certain cases, very similar items should not be recommended by the system, for example news articles covering the same event.

2.4.2 Collaborative Filtering Recommendation

Collaborative filtering (CF) is one of the earliest and most successful recommendation technologies, including such tools as the Tapestry system [27], GroupLens system [7, 28] and Video Recommender [29]. CF techniques work by building profiles of customers' preferences [30]. In other words, CF systems recommend products to a particular user by deriving common opinions from other customers. These systems include a certain sort of algorithms and techniques in order to identify a set of customers known as *neighbours* to users. Neighbours are defined as the customers who have relevant products ratings and they share a similar style of judgement with an active user. For instance, in a movie-related collaborative recommendation system, the system tries to search for the *neighbours* who have similar tastes to an active user based on similar movies they have rated before. Then, only the movies that are the most preferred by the *peers* of the active user would be provided as recommendations. CF recommender systems are based on the assumption that people who had a matching taste in the past will show the same pattern in the future.

The following discussion is related to Table 2.1 below, which shows five users' profiles represented as rows in a user-item ratings matrix U_1, U_2, U_3, U_4, U_5. The columns represent items I_1, I_2, I_3, I_4. Ratings profiles of users are $pr_1 = [4, \phi, 5, 5]$, $pr_2 = [4, 2, 1, \phi]$, $pr_3 = [5, \phi, \phi, 4]$,$pr_4 = [5, 3, \phi, \phi]$ and $pr_5 = [\phi, 3, 2, 1]$. For example, let us consider U_1 as an active user and by looking at his/her profile, we can see that U_1 did not rate I_2 and the system's objective is to decide whether I_2 is a convenient suggestion to U_1. Only three users U_2, U_4 and U_5 have used I_2 and they have certain opinions about it. However, U_2 and U_4 have close tastes in some items to U_1. Therefore, it is better to consult their tastes to estimate whether item I_2 is a good or bad suggestion for U_1. By contrast, U_5 has dissimilar tastes to U_1 and hence, the rating by U_5 will have a very negligible effect on the recommendation decision, or rather, the opinion of U_5 will be ignored. In addition, U_3 will be excluded from the recommendation process as this user has not used and rated I_2 although he/she has similar tastes to U_1.

Collaborative filtering algorithms can be classified into two main sub-categories: namely, memory-based and model-based algorithms. The former will be discussed in Section 2.4.2.1 while the latter will be covered in Section 2.4.2.2.

Table 2.1: An example of a user-item matrix

	I_1	I_2	I_3	I_4
U_1	4	**?**	5	5
U_2	4	2	1	ϕ
U_3	5	ϕ	ϕ	4
U_4	5	3	ϕ	ϕ
U_5	ϕ	3	2	1

2.4.2.1 Memory-based Collaborative Filtering Algorithms

Memory-based or neighbourhood techniques [7, 28, 31] use the entire user-item matrix to calculate recommendation predictions for unseen items. These systems identify like-minded users, or so-called *neighbours or nearest-neighbours*. Generally, memory-based recommenders work according to the following steps:

- First, memory-based recommenders ask users to rate some items in order to be able to recognise their taste.
- Second, statistical techniques are applied to define the user's neighbours who shared similar prior preferences.
- Third, after defining like-minded users, the systems provide predictions of ratings to those items have not been seen by the user, then provide recommendations accordingly.

In step two, a measurement is used to compute the similarity between two profiles. The close neighbours can be allocated by: (1) defining a certain number of neighbours N, (2) predefining a suitable threshold hence only neighbours whose similarity level exceeds the threshold are incorporated in the recommendation encounter, and (3) excluding neighbours who have highly dissimilar tastes reflected by a negative similarity degree. In step three, the systems can choose to provide the recommendations as a score of rating and the ratings values will be expressed using the same scale used to express opinions, for example, a scale from 1 to 5. Another different way of recommendation would be to generate a list of *Top-N* recommended items.

In memory-based techniques prediction for an active u about an item i denoted as $r_{(u,i)}$ is computed as aggregation of ratings of neighbours who have previously rated i

$$r_{u,i} = aggr_{a \in Neig} \quad (r_{a,i}) \tag{2.8}$$

where a is the neighbour of u who rated i and $Neig$ refers to the set of neighbours and where $Neig \subset U$ and $|Neig| = N$. Examples of some aggregation functions are given below:

$$r_{u,i} = \frac{1}{N} \quad \Sigma_{a \in Neig} \quad (r_{a,i}) \tag{2.9}$$

This simple average is divided by N the number of neighbours or $Neig$. The most common aggregation method is to compute the prediction as a weighted sum as shown below:

$$r_{u,i} = \frac{\Sigma_{a \in Neig} \quad sim(u,a)(r_{a,i})}{\Sigma_{a \in Neig}|sim(u,a)|} \tag{2.10}$$

sim here refers to the similarity degree between an active user u who needs recommendations and another user a. Therefore, a large weight of sim indicates that u and a are very similar to each other and consequently the rating $(r_{a,i})$ will participate more in the prediction of $r_{u,i}$. From Equation 2.10, the problem is that this computation dose not consider the fact that users may express their ratings using the rating scale in different way. The adjusted weighted sum, discussed in Chapter 6, is widely used to address this limitation. In this approach, instead of using absolute values of ratings, the weighted sum uses their deviations from an average rating of the corresponding user.

Similarity computation is a primary element in memory-based methods to compute different levels of similarity between a particular user and other users; a heuristic utility of similarity should be defined sim. To measure this similarity between two users an active user u and user a denoted as the utility $sim(u,a)$, this function illustrates the distance between the two users. The closer the users a and u are, the more weight will be given to the predicted ratings for user u in the prediction process. One popular approach to compute the similarity weight is the Pearson correlation coefficient (PCC) [31, 32]:

$$sim(u,a) = \frac{\Sigma_{i \in I_{a,u}} \quad (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\Sigma_{i \in I_{a,u}} \quad (r_{a,i} - \bar{r}_a)^2 \quad \Sigma_i \in I_{a,u}(r_{u,i} - \bar{r}_u)^2}} \tag{2.11}$$

where $r_{u,i}$ is the rating value given by an active user u to item i, while $\bar{r}_u$ is the average of all ratings given by user u. Similarly, the rating given by the neighbour a to the same item i is $r_{a,i}$ and the average of all ratings given by a is $\bar{r}_a$. Sim will be computed over the set of the co-rated items represented by $I_{u,a}$.

Furthermore, certain heuristic similarity measures to overcome the cold-start problems are proposed in [33], while [34] presented a similarity metric using a prior stage, in which a genetic algorithm generated weights which are dependent mainly on the nature of the dataset provided from each recommender system.

2.4.2.2 Model-based Collaborative Filtering Algorithms

Collaborative filtering also use machine learning techniques in order to make intelligent predictions. Building models algorithms includes three main steps:

- The designed models first learn the pattern of collected users' ratings in a training dataset.
- Testing the designed models and tuning all the needed parameters to satisfy problem requirements; for example, minimising the absolute squared error.
- After building reliable and tested models, ratings can be predicted and recommendations can be provided to the desired users using these models.

Researchers in the recommendation area have introduced different learning models based on *machine learning* algorithms, for example:

Classification: The authors in [35] used simple Bayesian CF algorithms in a collaborative filtering prediction process. Using a Naive Bayesian (NB) strategy, they assumed that the features are independent, given the class, to compute the probability of a class with all the features. The predicted class will be classified when the class has a high probability as follow:

$$r_{(u,i)} = E(r_{(u,i)}) = \Sigma_c \in Classset \quad P(r_{(u,i)} = c | r_{(u,j)}, j \in I_u) \tag{2.12}$$

where c represents the rating class from the scale used $Classset$, and j is an item that has been seen and rated by u and belongs to the set of all items I_u that are rated by u. In the case of a class that has missing features, the model computes the probability and classification over observed data. In the ratings matrix, they converted multiclass data to binary class data as a boolean feature vector for the purpose of simplicity. However, this method caused limited scalability and loss of multiclass data. In fact, real world problems include multiclass data; therefore, researchers tried to improve Bayesian CF algorithm to adapt to multiclass data. Authors in [36] conducted empirical experiments that showed that Bayesian CF can be better scalable and less time-consuming in the process of prediction; however, these can have worse predictive accuracy than the Pearson correlation based CF algorithm.

Clustering: Clustering-based CF algorithms have also been used to improve prediction quality. The basic idea behind clustering techniques is to assign a cluster to every similar group of data, such as group of users who share similar tastes. Cluster members can be seen as like minded neighbours. Sarwar et al. [8, 30] discussed the limitations of CF techniques and presented clustering based algorithms to enhance speed performance. They applied two phase algorithms: (a) clustering the user-item ratings database to N partitions, and (b) using memory-based CF algorithms to estimate recommendations for every users within the clusters based on only preferences from cluster members. Experimental study showed that making recommendations predictions within smaller clusters, using *k-means* algorithms, improve scalability in clustering techniques when compared with classical CF techniques; they reduce the number of neighbourhoods to be tested due to the static pre-computed clusters, and as a result, the online prediction process becomes much faster [37]. The experiment in Sarwars clustering methods presented two observations. First, clustering algorithms showed lower prediction quality in comparison with the basic CF approach. Furthermore, it was evident that as the number of clusters increased, the prediction error also increased. An explanation of this can be that the increased number of clusters may lead to smaller cluster sizes, and therefore, resulting in an insufficient number of neighbours to create an representative opinion about a particular item. In addition, clustering techniques have the limitation that users clustered to a single group may not receive recommendations out side the cluster's taste trend, and this often causes less-personal recommendations and most often worse accuracy than memory-based algorithms [31].

Regressions: Regressions-based CF algorithms are also presented in the literature to approximate users' ratings. Due to the numerical nature of the ratings data in the real world, regressions models can contribute in predicting numerical values. Assuming $X = (x_1, x_2, ..., x_m)$ is a set of random ratings where $x \subset I_u$,

$$Y = AX + E \tag{2.13}$$

where A is $m \times k$ matrix, k is the k-dimensional rating space where E represents the noise in user preferences. The predicted matrix Y is $m \times n$ where $Y_{u,i}$ is the rating of user u to item i. The study reported in [38] proposed a collections of linear models to search for similarities between items. This regression approach succeeded in combining linear models to compute rating predictions for an active user. In order to estimate the parameters of the linear regression function, the authors used *least squares* metric. They showed that their approach offers good performance in addressing *sparsity*, a common problem in CF methods.

Matrix factorisation models: Recently, several matrix factorisation (MF) approaches have also been proposed [39, 40, 41]. In fact, the idea of implementing MF models has widely attracted researchers because of two properties: (1) an attractive level of accuracy, and (2) sufficient scalability over large datasets. The general idea behind MF is modelling both users and items as an inner product and producing a joint space of latent features space of a specific level of dimensionality. MF models infer latent features underlying the ratings interaction between users and items in the user-item ratings matrix. These features explain how one user may rate an item and the model uses these features to approximate the ratings matrix into a low-rank one. In the movies recommender, features may measure how much a given user is interested in a given movie. More formally, the rating is computed by the following perdition formula:

$$\hat{r}_{u,i} = \mu + b_i + b_u + q_i^T p_u \tag{2.14}$$

The model parameters b_i and b_u are learned by minimising the squared error. And μ is the average overall ratings. For example, the approach in [39] achieved the goal of approximation of the original user-item ratings matrix using maximum margin MF (MMMF) that minimises the sum of the squared error between actual ratings and predicted ratings. Bell and Koren [42] presented a solution for the *Netflix prize* [43] by combining several linear combinations of prediction models and managed to win the prize; the challenge goal for them was to achieve 10% reduction in the Root Mean Squared Error (RMSE). Several approaches have appeared as extensions to the MMMF model; for example, the study in [44] proposed to avoid overfitting of the regularisation parameters, a fully Bayesian treatment of the probabilistic matric factorisation (PMF). The authors in [41] proposed several matrix factorisation MF approaches such as an incremental variant of MF that efficiently handles new users' ratings.

These algorithms are different from our goal in this thesis because of the total main focus in the aforementioned models is to improve performance in terms of accuracy based on the ratings information while they ignore more personalised data sources, such as OSNs. Next, we discuss the advantages and challenges related to CF techniques.

2.4.2.3 Challenges in CF Recommendation Methods

CF recommenders have several useful points when they are applied. One of the most important point is that they can provide recommendations using only the direct data source which is the ratings profiles. Furthermore, the more the users respond to the recommendations, the better the systems can adapt to users' tastes over the time; i.e., rich users' profiles imply better quality recommendations.

The most challenging aspects in using CF approaches, which the literature pointed out and efforts have been made to try and overcome, include:

The level of prediction accuracy: Most of recommender models in the literature seek high levels of prediction accuracy. Researchers use the accuracy aspect in order to provide proof of how successful their proposed recommendation approaches are in providing predictions that are close to users' tastes, as stated above in [42].

Cold start problem: Both CB and CF recommender systems suffer from the so-called cold-start problem. To be able to provide accurate recommendations, users' preferences should be analysed by collecting ratings which is not the case for new users who have recently entered the system; hence, this may lead to poor recommendation results. Similarly, the problem of cold-start also exists in the case of new items. Recommender systems are updated regularly with new items. To incorporate these new items in the recommendation process, these items must be rated first by substantial number of users.

The problem of sparsity: A common observation that the number of items that people rate in any recommender system is usually very small in comparison to the total number of items that need rating prediction. Ultimately, the total number of users' ratings affects the success of collaborative recommenders. For instance, in a movies recommendation collaborative system, when a movie has been rated by only few people even if rated highly, this movie would not be recommended extensively. Furthermore, a common issue with users who have unusual taste compared to the application community is that the systems may not be able to allocate similar users [5, 45].

Below we briefly discuss additional important challenges [6]:

- Scalability of the algorithms with massive and dynamic sets of data. Several recommendation techniques in real systems deal with large interactions between users with items such as ratings, preferences and reviews, etc. These solutions should be introduced to ensure that off-line tests are sufficient when running a real application. In [46], the authors designed incremental and parallel versions of co-clustering algorithms and utilised them to build an efficient real-time CF framework. Their empirical evaluation showed better results in terms of accuracy when compared with classic correlation and matrix factorisation with much lower computational cost.

- The majority of the current developed recommender systems generates recommendations when users explicitly request them. However, nowadays, different scenarios emerge due to the fact that ubiquitous computer means that users are always connected. Therefore, there is a need for a recommender system to be able to detect implicit requests. Also, the new generation of recommender systems should predict when and how to generate recommendations to be proactive rather than reactive [47].

- Diversity is another challenge for an active user. Researchers agree that a recommendation list should include a certain level of diversity in the recommended items. In early stages of the recommendation process, users may need to explore different tastes and preferences. In other words, users may use the recommender as a knowledge discovery tool. There are some efforts to shape the nature of this ”diversity ” and how to balance the goal of accuracy of recommendation to achieve successful personalisation of users’ preferences as a constraint, with the goal of diversity. More discussion about diversity is in Chapter 6.

- Recommender systems and users’ privacy is another issue. In order to provide users with personalised recommendations, recommender systems collect as much user data as possible. These actions may inconvenience some users when the system seems to highly know and understand their true preferences. Therefore, it is important to provide solutions with confidential and sensible use of user’s data. Importantly, these solutions should ensure protection of users’ profiles against malicious users attacks.

- The methods used by recommender systems can be divided into two approaches: long-term profiling when the systems aggregate user behaviour data, e.g., collaborative filtering, and short-term preferences when the systems capture temporary preferences of users, e.g., case-based methods. It is clear that both approaches are important. In fact, new efforts should be made to combine both approaches in order to build hybrid type models, which precisely use the content of users’ preferences. Also, efforts should be made to make explanations of the underlying relation between short-term and long-term preferences.

- The ability to create generic models and cross domain recommender systems to act through different applications and domains. The core of current recommender systems can be built upon hybrid approaches but cannot easily utilise users' preferences profiles to provide recommendations in another different domain.

- The recent scenario of grid or cloud computing is a powerful opportunity to provide more robust and flexible RS models that can be distributed and operate in open networks. The traditional RS models are based on client-server architecture, where user-client asks for recommendation from a server-recommender and then provides the client with suitable suggestions. The new open networks may help in solving classical problems in the centralised systems which are the vast majority of RSs. In study reported in [48], the authors proposed a distributed hash table based technique to gain efficient user database management and retrieval in decentralised CF systems. They proposed a heuristic algorithm to allocate similar users from a distributed hash table overlay network and to provide recommendations locally.

- Recommenders that can suggest optimised sequence of recommendations. This represents an attempt of conversational RSs to enhance the quality of recommendations based on the approach of one-time request/response. These type of systems can be improved by adding a learning skill that can optimise the recommended items and also the dialogue between users and the system must be used in all possible scenarios.

- Mobile computing for more personalised recommendation. Designing recommenders in the most compatible platform in mobile devices is a crucial demand. People may require recommendation when they are on the move, e.g., at shops, at the movies or on vacation. This environment needs to deal with different user interface and computational model constraints. In other words, solutions should consider the limitations in computational power and screen size. For example, the study in [49] illustrated a prototype design of a software agent that can make recommendations about travel-related services according to the context of the user in a mobile environment. Another effort published in [50], the authors presented an integrated approach of recommendations with electronic map technologies. They implemented a map-based mobile recommender system that can intuitively generate personalised suggestions to mobile users.

2.4.3 Other Types of Recommendation Methods

In this section, hybrid recommendation solutions are briefly introduced along with other recommendations approaches. Hybrid approaches combine two different approaches together to integrate the advantages of different approaches and to overcome some of their challenges. To address the limitations in CB and CF techniques, several studies [51, 52] compared the performance of both CB and CF recommenders and demonstrated that hybrid approaches can generate recommendations with higher accuracy compared to using either CB or CF methods. Combining CB and CF methods can be proposed in different ways [5]:

1. Combining the outcome from the prediction process after applying CF and CB separately.

2. Adding some CB features to improve Collaborative recommenders.

3. Adding some CF features to improve Content-Based system.

4. Defining a general model that includes both CB and CF features.

Recently, different recommender system approaches have appeared to propose solutions to the inherent limitations mentioned earlier. Due to the growth of Web2.0 applications, new areas are emerging and need to be explored; for instance, social tagging system (STS). Social tagging facilities appear to allow ordinary users to publish and edit contents and share free keywords. Thus, RSs are implemented to support users in finding relevant topic and some commercial STS are starting to generate recommendations such as Deliciuis.com. This new direction of RS requires to investigate novel facets, approaches and algorithms. STS deals with a third dimension, which is the tag in addition to the user and the item dimensions and this may affect the complexity of the algorithms being used [53, 54].

Next we introduce in detail two perspectives to improve collaborative recommender systems related to the scope of this thesis, which are the adaption of trust in RSs in Section 2.5 and item reviews-based recommender system in Section 2.6.

2.5 Trust in Recommender Systems

This section introduces *trust* in RSs. Trust in recommender system is a notion that correlates with similarity of preferences between users. In the study [55], Guo has considered trust as one's belief towards the ability of others in providing valuable ratings and evaluations of items.

In general, by looking at trust in computer science, it is used as follows. First, trust models can be used as policies such as to ensure confidence in systems; examples include enforcing access policies, management of credentials identifying information sources and reputation-based trust models in peer-to-peer (P2P) networks [56]. The second set of trust models includes using the history of user behaviour and trust in relationships within social networks [57]. The trust methods used in this study belong to the latter group.

More precisely, trust perspective in recommender systems has a major role in overcoming limitations and issues. Consuming trust increases systems productivity and supports reliable influence of online behaviour to produce recommendations [58, 59]. Particularly, there are two different uses of trust in the recommendation domain: (1) trust relationships between users, and (2) trust in the system's output/recommendations. In the former use, recommender systems attempt to deliver more useful suggestions derived from a user's profile and his/her social connections. Trust-based recommender systems utilise the relationship that exists between users to implement recommendation algorithms. The latter use of trust deals with the system as one of the system's roles, such as effectiveness (supporting users to make the right decisions) and persuasiveness (convincing customers to try or buy items). The former use is within the scope of this thesis.

If we look at the three users in Figure 2.2, users a, b, and c; when mimicking a real world scenario, user a may ask friend b about his/her experience using item i. If the friend b does not have experience using this item then b can ask friend c about his/her opinion about item i. The recommendation by friend c can be reliable to some extent based on the available trust information. It can be observed that in this way, the system can personalise recommendations, where more desirably recommended items can be approached based on reliable users.

2.5.1 Classification of Trust Models

Probabilistic and gradual approaches: Trust models can be classified into probabilistic approaches and gradual approaches. Probabilistic approaches maintain trust values in a black and white style. In this case, a higher probability indicates that a source can be trusted. For instance, in the study [60], the focus was on calculating trust values in systems involving semantic information, such as a bibliography server.

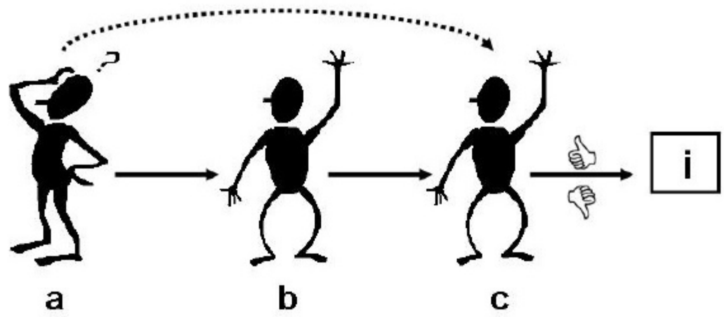


Figure 2.2: A user needs reliable recommendations from others in the system [2]

On the other hand, gradual trust approaches deal with prediction of trust values when the outcome of an encounter can be positive to some degree; for example, some information cannot be completely right or wrong, but rather, it can be correct to some extent or incorrect to some degree [58, 61]. Indeed, in real life, trust is managed as a gradual phenomenon; humans tend to highly trust some people or trust them more or less than others.

Trust and distrust: The last decade has witnessed an increase in research on gradual trust as explained above; however, most of these studies focused on computing only trust and ignored distrust; this is because modelling of distrust is considered a relatively unexplored area because there is a growing opinion that distrust cannot be defined as the lack of trust [62]. Many gradual trust models including both trust and distrust are carried out, for instance, Guha et al. [63] implemented trust as a pair of values (t, d), denoting trust and distrust, with degree taking values in the interval [0,1] for both t and d. They computed the final suggested trust judgement by subtracting d from t. Potential information may be lost in the case of merging the two values. For example, in the case of (0.3, 0), where the net trust value is 0.3, the same result will be obtained in a different case, say (0.7, 0.4), which is not a true representation of the real scenario. Victor et al. [64] proposed solution to the issues raised in Guha's model and applied a bilattice structure to deal with the situation of distrust or lack of knowledge, while further explaining the required knowledge in their approach.

Global and local trust: Trust can be implemented as a global or local parameter. Local trust metrics compute trust according to the subjective beliefs of an active user in other users' opinions; hence, the local trust score will vary among users as they have different points of views towards each other [58, 59, 65]. By contrast, global trust can be managed as a community-wide reputation level and then this also can be classed as an objective trust value; examples including reputation systems, which use global trust values [66], and the PageRank global algorithm used by the search engine Google.com [67]. Global trust scores are computed as the community's general point of view regarding a specific user. It is worth mentioning that local trust requires to be computed precisely to suit every single user's opinions, which is computationally expensive, whereas, global trust algorithms run once for all users; i.e., global trust models may compute trust in relationships among all users in a particular community while local trust metrics estimate a personalised trust score based on user point of view. However, objective trust estimation may utilise subjective trust, which occurs when aggregating weighted subjective trust values. Another concept, *credibility*, is also used in the literature to indicate global trust [68, 69].

2.5.2 Trust Computation

Trust can be computed as an estimation of how much one user trust another user by understanding their shared connections and behaviours within the networks. In the literature of trust-based recommenders, two strategies are used in building trust metrics p̂ropagation and âggregation [2]:

Propagation: The assumption behind propagation is that trust considered transitive to some degree. For example, when user u_1 trusts user u_2, u_2 is called trusted third party (TTP), and if TTP u_2 trusts user u_3, then it is possible to assume that u_1 may trust u_3 to a certain degree. This basic strategy is called atomic direct propagation. There may be several paths between two users and of course these paths may have different lengths. For instance, if there are two paths between the source user u_1 to the target user u_3, different actions may be taken, which will rely on the total length of the path, or a choice may be made to follow the shortest path links between two users.

Aggregation. A trust metric may also use an aggregation strategy. To illustrate this technique, let us consider that several paths linking to an active user, for whom the system is trying to predict a trust score in a large network. In this case, the trust prediction may be generated via different propagation paths, which must be integrated into one aggregated estimation.

Combining both strategies propagation and aggregation is often used, and the final trust evaluation might depend on the way they are used together. Classical aggregation and propagation can be used as weighted operators in a weighted sum, an average or a weighted average in the recommendations process.

Trust metrics in recommender systems research can be viewed as two main categories:

- Explicit trust data used to build trust network in rating-based systems. Classical examples includes TidalTrust presented by Golbeck in her PhD thesis [58] and MoleTrust proposed by Massa and Avesani in [59] and [70]. This will be further explained in Section 2.5.2.1.
- Inferred or implicit trust in rating-based systems, such as that developed by O'Donovan [71]. This will be discussed in detail in Section 2.5.2.2.

Furthermore, another new category of trust data is introduced in this thesis based on the idea that trust can be inferred between people; what is new is that this approach uses the user's online social networks to capture this implicit trust. In other words, recommender systems can use trust between friends in popular social media networks in recommendations. This type of trust in our proposed approach will be discussed in more detail in Chapter 3.

2.5.2.1 Using Explicit Trust

Trust-based recommender systems are based on the estimation of opinions from reliable users instead of similarity between two users. In the literature, the most common strategies aim to establish trust networks, which are created individually by users to assign a degree of trust towards other users. Studying trust by mining users' web of trust (WOT) has attracted the attention of many researchers over the past decade [58, 59, 70].

For example, Figure 2.3 illustrates how trust networks can give the opportunity to reach more users and items. As mentioned before, user a and user b gave a similar rating to item i_1 and so they were linked together through their ratings. As a result we can say that user a may have an interest in i_2. However, there is no way to find out whether user a would like item i_3. This situation may be solved when a trust network is adapted among users. The solid lines indicate the trust relations between users as it is assumed, where user a expresses a certain level of trust in user b, and user b shows a similar trend in trust to user c. By issuing trust statements, a WOT can be built; therefore, more accurate recommendations can be provided. Based on the idea of propagating trust, this relation can be true $(a \Rightarrow b \Rightarrow c) \Rightarrow (a \Rightarrow c)$, and then we can recommend item i_3 to user a.

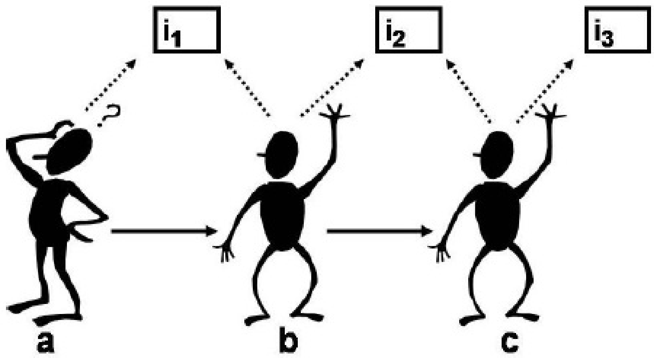


Figure 2.3: Recommending item i to user a [2]

In [58], the study focused on the path strength that connects user a and user b. Golbeck proposed a method to indicate similarity using a trust metric called TidalTrust. She deployed TrustFilm as a social network website, where users rate movies and add friends and rate explicitly how much they trust them about movies selections on a particular scale. On TrustFilm, trust network inference algorithm TidalTrust was used to predict trust ratings [72, 73]. The main formula used by Golbeck, namely classical weighted average, to predict a rating R for an active user u about an item i is:

$$R_{u,i} = \frac{\sum_{a \in R^T} \quad t_{u,a} r_{a,i}}{\sum_{a \in R^T} \quad t_{u,a}} \tag{2.15}$$

where the trust value $t_{u,a}$ reflects how much user u trusts rater a and R^T represents the group of users who rated i. The trust value $t_{u,a}$ should exceed a given threshold based on a trust scale from 1 to 10. The trust value $t_{u,a}$ is computed based on *TidalTrust*, which includes all friends v of user u in the set WOT^+ if v's trust score exceeds a predefined threshold:

$$t_{u,a} = \frac{\sum_{v \in WOT^+} t_{u,v} t_{v,a}}{\sum_{v \in WOT^+} t_{u,v}} \tag{2.16}$$

The authors observed that shorter prorogated paths with higher trust values produce better recommendation results. A problem seemed to occur in Golbeck's work when recommendations were generated only when there is a connection; i.e., a recommendation could not be provided when no path from the target user to any another user was found. They asked users in the FilmTrust network to establish at least one social connection path in order to increase the ability of system to provide recommendations. This limitation may affect the accuracy of the resulted recommendation due to the problem of trust sparsity and new users who did not issue any trust evaluations statements.

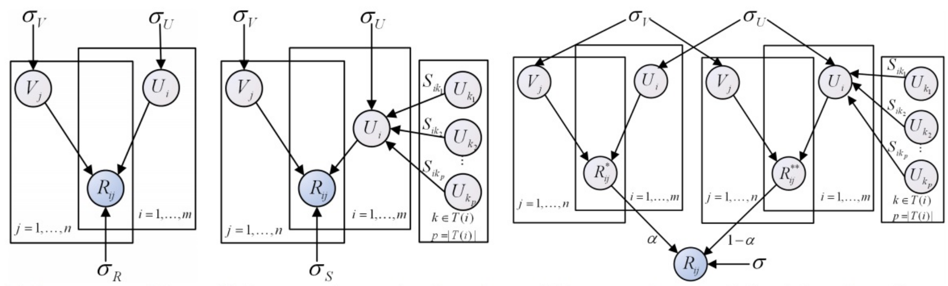
In another study, Massa and Avesani [59, 74] built a trust model from direct explicit trust data obtained from users on the popular *Epinions* service, which is a website that allows users to rate different items, such as cars, books, movies, etc. In addition, users have the ability to rate other users' reliability from a personal point of view. The authors argued that the collected trust information can solve the sparsity problem found in traditional collaborative filtering techniques. The sparsity problem happens when two users rate few items so it is unlikely to have many co-rated items, and as a result, computing similarity between two users becomes difficult. The study in [59] implemented trust in a close manner to the collaborative filtering algorithm; however instead of using Pearson correlation similarity weights $w_{u,a}$ to define the close neighbours, these are changed to trust values $t_{u,a}$ as shown in Equation 2.17. Also, they based their model on the fact that trust and similarity are correlated.

$$R_{u,i} = \bar{r}_u + \frac{\Sigma_{a \in R^T} t_{u,a}(r_{a,i} - \bar{r}_a)}{\Sigma_{a \in R^T} t_{a,u}} \tag{2.17}$$

where $\bar{r}_u$ and $\bar{r}_a$ are the rating average of user u and neighbour a, and R^T represents the group of trusted users who rated i. Therefore, the above formula can be called *trust-based collaborative filtering*. The authors arrived to the conclusion that because of trust prorogation operator, farther away users can be approached and compared. However, ratings from users at one step distance from the active user provide more precise and useful recommendations than those users at two or more steps.

Within the explicit trust context, Ma et al. [3] used *social trust* to introduce the idea of a realistic trust-based recommender system to enhance existing methods which mostly suffer from sparsity and scalability problems. They tried to explain in much more depth the relationship between the user-item ratings matrix and the user's WOT. In order to have a more accurate and realistic recommender system, they introduced a novel probabilistic matrix factorisation framework to fuse the users ratings and their trusted users ratings together based on three steps.

First, they learned the characteristics or features of users by applying matrix factorisation machine learning techniques to factorise the user-item ratings matrix. The idea behind this technique is to derive a high quality representation of low-dimensional features for the users-items matrix as represented in Figure 2.4(a). Second, they proposed a MF to generate recommendations but based only on the trust weights and the corresponding ratings by the trusted friends of unseen items. Trusted friends can be defined as the systems's users who were rated by a given user with a trust score that indicate their reliability.



(a) Factorization of User-Item Matrix (b) Recommendations by Trusted Friends (c) Recommendations with Social Trust Ensemble

Figure 2.4: Graphical models used in social trust recommender [3]

Hence, the decision of recommending an item is obtained as an estimation guided by friends' ratings weighted by the trust score related to the friend. Assuming that there is a directed social trust, and between a pair of users, the edge weight is $S = S_{u_i,u_j}$ which represents how much user u_i trusts user u_j, and T is the group of trusted friends, then the prediction of item k rating for the user i is described as follows:

$$R_{u_i,k} = \Sigma_{j \in T_i} R_{j,k} S_{u_i,u_j} \tag{2.18}$$

Since $R_{j,k}$ is the rating given by trusted friends, this assumption generates recommendation purely based on the trusted friends' tastes. The graphical model is explained in Figure 2.4(b); the authors defined a conditional distribution over the observed ratings and applied a Bayesian inference. At this stage, the observed ratings depended totally on the group of trusted friends. In the third step, in order to satisfy their assumption that in the real world people have their own tastes but they are also influenced by friends opinions. The authors defined MF model reflects both sets of tastes as the recommendations can be partially influenced by users' tastes and their friends' tastes. They used Epinion as a dataset because every member of Epinion edits a trust list, and therefore they can build a trust network and apply the three proposed Matrix factorisation models. Figure 2.4(c) illustrates the third step. By tuning models parameters, they concluded that only using the user-item ratings matrix or relying solely on users social trust network for recommendations could not provide superior results to fusing these two models together.

Using a different context, some studies [75, 76] assumed that people naturally use trust linguistic expressions rather than numerical ratings; therefore, their proposed solutions were based on a fuzzy set. These authors introduced a fuzzy trust network among users and expressed trust as seven triangular fuzzy numbers. They used the trust results to enlarge the neighbourhood size to overcome the high sparsity level which always affects recommendation quality. After increasing the size of the trust neighborhood, more similarity values could be computed. At last, the standard Resnick's formula [7] is used for the final prediction step. In fact, they used trust as a key factor to filter users but they did not include the trust values when computing predictions. In their experiments, they used MovieLens dataset, and because this dataset does not contains trust statements, they randomly generated a trust matrix consisting the corresponding ratings of to linguistic expressions.

We can outline the main limitations of these works as the following:

- The trust network data are very limited in comparison with the number of people available; i.e., sparsity of trust data also appears in these approaches and the systems need the users to express some trust scores towards others in order to generate recommendations.
- The assumption that users evaluate other users explicitly is not practical and does not reflect real world cases.
- These approaches incorporate only subscribers in a commercial website in the context of trust, and this is not a true representation of the claimed social trust connections from social networks.

2.5.2.2 Using Implicit Trust

To overcome the identified weaknesses in the approaches that rely on explicit trust, implicit social trust-based RSs have emerged to estimate and infer trust values. Most commonly, algorithms based on automatic generation of trust use past ratings history of users in the system. More specifically, assigning high trust values for a particular user depends on his/her past profile of delivering accurate recommendations. Another consideration related to trust, the study reported in [77] analysed users' opinions in relation to the correlation between similarity of interest and user trust for making recommendation based upon opinions from trusted peers rather than the most similar ones in an automated online recommender. The finding showed a strong positive relationship in the assumed hypothesis of the correlation between the trust and similarity of interest between users.

Within this context, O'Donovan et al. [71, 78] also considered inferred or implicit trust in recommendation approaches. To be able to allocate users with the highest trustworthiness levels, O'Donovan introduced two trust metrics: profile-level (average trust for the profile overall) and item-level (average trust for a particular profile when making recommendations for a specific item). These two models measured the correctness of recommended items from *producer* user to the target user u; i.e., recommendations coming from user a is considered true only if it is within user u's actual ratings profile $r_{u,i}$, and this will take a binary success/fail score. The full set of recommendations that a given producer has been involved in, $RecSet_{producer}$, is given as:

$$RecSet_{producer} = \{r_{(u_1,i_1)}, ..., r_{(u_n,i_n)})\} \tag{2.19}$$

$$CorrectSet_{producer} = \{r_{u,i} \in RecSet_{producer} : r_i \in Pr_u\} \tag{2.20}$$

Then trust of a producer profile is computed as a percentage given by:

$$trust_{producer} = \frac{|CorrectSet_{producer}|}{|RecSet_{producer}|} \tag{2.21}$$

$trust_{producer}$ represents the profile-level trust, the more successful in making many accurate recommendations in the past, the higher the trustworthiness weight for a producer profile has in comparison to a poor recommendation profile history. For instance, if a producer has been involved in the recommendations process 100 times, and 40 of these results contributed to correct ratings predictions, the profile level trust score for this user is calculated as 0.4. Obviously, the profile-level trust computes the trust of the whole profile in general. It is worth noting that O'Donovan's methods are considered as global trust metrics, and to compute the final rating, O'Donovan et al. incorporated their proposed trust metrics into a standard collaborative filtering algorithm [7] based on the strategy of trust-based filtering described by Equation in 2.17. The results showed that the use of trust values has a positive impact on overall prediction error rates. This works focused on reducing the average error prediction in order to achieve high accuracy. Other problems such as data sparsity, have not been discussed in this model. This work did not show whether or not incorporating trust values can increase the number of the overlapping items between two users. Furthermore, the method reflected a global trust understanding while a local trust metric is needed for the goal of personalisation and customisation of suggested items for a particular user. In addition, this method requires rich profile ratings history; otherwise, the system would not be able to provide results.

Kim et al. [12], presented a computational framework for estimating trust level between a pair of users in ratingbased online communities (e.g., reviews) without providing an explicit WOT. In their framework, they combined a reputation concept by developing *content providers' expertise* and a trust concept when defining *content user preference*. They used mainly feedback ratings on contents rather than the WOT as they argued that ratings on items are much more frequently provided by users in comparison with task of rating other users. A disadvantage of their study is that they still based their model only on the ratings information and ignored the opinions included in the feedback. However, they did not refer to analysing implicit relationships between users in these networks.

In another study that explored implicit trust and distrust in the global context. Victor et al. [64], highlighted that many studies based on WOT explicitly using either binary or continuous trust values, which have used global or local trust cannot be applied for rating-based online businesses unless users evaluate other users trustworthiness. Furthermore, in the study [79], Victor discussed the weaknesses of trust-based recommendation, and suggested that to tackle such limitation, it is important to explore other ways of analysing trust relations when the needed information cannot be provided by the user. For instance, one way forward is exploiting online friends' connections networks; e.g., Facebook, Twitter, LinkedIn, e-mail and reputation systems, such as eBay.

In study [11], the authors explained how to employ data from friends in social networks to estimate missing values in the user-item rating matrix. However, they illustrated that trust-based research which has been explored in the literature, has three main weaknesses as follows:

1. The *trust relationships* are different from *social friendships* in many aspects. Typically, on the web, a user *a* maybe add a user *b* to a trust list, when a user *a* likes and shows some interest in user *b*'s opinion. To this end, the trust generation process is a unilateral action. That means, this process does not demand that user *b* should confirm the relationship by adding user *a* to his/her trust list. What is more, this does not imply that user *a* knows user *b* in real life. On the other hand, "social friendships" indicate the cooperative relationships and the connections around us; for example, classmates, relatives or even our favourite writers.

Many of the social networking websites such as Facebook, and Twitter, are designed to establish online connections between friends in the real world. Based on this discussion, we can arrive to an understanding that trust-based recommender systems do not involve the meaning of "social recommendation" because this concept implies a social friend network to enhance recommender systems. Nevertheless, the difference in meaning may cause a remarkable difference in building trust models.

2. The assumption of trust-based recommender systems is that trust can be built when two users *a* and *b* have similar tastes. In fact, this is not always the case in social recommendation since the tastes of one user are not necessarily shared by all of their friends. In other words, a person may have common tastes with some friends, while his/her other friends can have completely different preferences. Hence, the algorithms of traditional trust-based recommenders cannot be directly used in providing recommendations in social recommender systems due to the different hypothesis of similarity and trust.

3. The vast growth of Web 2.0 applications has a huge impact on people. People spend more and more time on online social networks which is currently the most attractive activity on the web. On the other hand, there are few online systems which consider the trust concept such as Epinions.com. Thus, to provide personalised recommendations to the online users, researchers should broaden their studies to encompass social recommendation.

To do so, in their solution, the authors [11] selected the low-rank matrix Factorisation to factorise the user-item rating matrix and used the factorised user-specific and item-specific matrices to predict more missing values. The proposed model was based on the assumption that in order to make a reasonable decision, people tend to ask their friends for valuable suggestions. They defined two sets of friends: F+ is to denote outlink friends and F- is to denote inlink friends and sometimes these two sets were equal. For example, on facebook, F+ and F- are equal since if user *a* is in user *b*'s friend list then user *b* must be included in user *a*'s friend list. On the other hand, in some trust network like Epinions, F+ is not equal to F-, which means there is no bidirectional relation. They worked to minimise the taste difference between a particular user and friends. A similarity function is used to indicate whether the distance between the features vectors for one user and the feature vectors for his/her friends is small or large in order to treat users' friends differently. They applied very popular methods in the literature in computing similarity, which are vector space similarity (VCC) and Pearson correlation coefficient (PCC).

However, there are some disadvantages related to this work. First, their models were based on the user-item rating matrix, and therefore, these models could work only when a sufficient amount of ratings information is provided. Furthermore, they assumed that friends in social networks rate items as well, and this is not the case all the time because friends in these networks express opinion by writing comments. In other words, the model did not use any external sources of data related to friends' connections, such as Twitter. In addition, it focused ultimately on improving predictions accuracy and ignored the evaluations of cold-start profiles problem and the diversity challenge.

We can conclude important remarks about the gaps and limitations in the above mentioned solutions that use implicit trust in recommender systems. Most of mentioned studies limited the term 'social recommendation' to only the subscribers to RSs and ignored connections with friends on OSNs. Most of these models used ratings data to provide recommendation results and did not involve the use of other source of data. Furthermore, trust was interpreted as similarity between users and did not reflect users' communications behaviour as another indication of trust relationships.

2.5.3 Challenges in Trust-based Recommenders

This section illustrates issues and limitations encountered when incorporating trust for recommendations tasks. These difficulties will be described as follows:

1. Regarding the cold-start problem, Massa and Avessani [74] explained that when this issue occurs in classical recommenders, it can be alleviated by establishing a trust network among the users of recommender. To do so, new users should issue some trust statements to obtain suggestions from the system. However, in another study Victor and Cornelis [80] discussed that the new users very often face the cold-start challenge in WOT. Hence, in order to have a trust network new users should establish connections with other users and expand their trust network as soon as they entered the system.

2. Supporting visualisation: how to start relations in a trust network seems to be a problem. Using visualisation approach may address this problem by introducing a framework to visualise trust-based collaborative filtering. The system visualises both information coming from the classical similarity computation (PCC)

and information from trust values obtained from rating data. Based on the interactive interface to represent users in the system, this approach allows new users to indicate their tastes and obtain real-time trust information [71].

3. Using online social media: when information cannot be provided explicitly by the users, a new source of data should be explored. Divers sources of social data can be investigated, such as online friends networks (e.g., facebook or LinkedIn). Recommender systems rely on different behaviour theories, such as cognitive similarity between people and preferences, social capital in reputation systems and ties strength. All these social data may contribute successfully to trust-based recommenders; however, the number of studies carried out in this area is still limited. Research should investigate further to identify how much of these data are useful and to evaluate the performance of the new systems and those of the traditional trust-based recommendation approaches [11, 79, 81].

4. Exploring the potential of distrust: few efforts have been done in modelling distrust [64], and this is because of the limited availability of datasets that include distrust information, which is a major issue. Also, there is no general agreement about how to incorporate this type od data into recommenders. For example, some strategies tried to involve distrust as a user filter [2].

The study reported in [77], showed a strong positive relationship between trust and interest of similarity between users. However, to broaden the trust use in RSs and simulate real life interactions, new methods should be implemented to compute implicit trust values. To this end, practical trust must be obtained implicitly and from people who have meaningful relations to the users.

The next section discusses reviews feedback that can contribute to the domain of recommendation and is related to this thesis.

2.6 Reviews in Recommender Systems

Because of the collaborative characteristics of the current World Wide Web, people gain benefits from these features in two ways. First, they can publish their opinions about trending topics or even describe their experience about some products or services. Second, they can seek for this information and get benefit from different forms of product feedback, such as ratings or text reviews. According to a survey by Pang et al. [82], more and more people are willing to share their opinions with both friends and strangers. In this survey, results showed that users are eager to consume online reviews and recommendations. Therefore, there is vast amount of research in this field to aid consumers in accessing information about opinions and provide tools to assess sentiment and evaluation [15, 16, 83, 84]. Furthermore, online industries also have more interest in leveraging platforms on Web 2.0 such as blogs and social networks, and can track opinions about products by accessing beneficial information about purchasing decisions.

The first appearance of the term *opinion mining* was in a paper by Dave et al. [85] in 2003, in which they defined opinion mining as processing a set of search results of a given item to generate a list of product attributes and features and aggregate opinions about each of them. This describes the main role of opinions mining in extracting and analysing judgments on items features. On the other hand, the term *sentiment analysis* (AS) is the analysis of sentiment in text, and this term first appeared in the work of Das and Chen in 2001 [84], who were interested in analysing market sentiment. A sizeable number of studies use the phrase sentiment analysis in applications concerned with classifying reviews to either a positive or a negative polarity. Nowadays, the term is used more broadly to indicate the computational treatment of opinions and sentiment in text.

Generally speaking, a review can be seen as a set of words but only a few of these terms hold a sentiment or represent the reviewer's opinion. Owning to the huge popularity of online commerce, a body of work tried to answer the question of whether the polarity of online reviews has any influence on actual purchasing decisions. It is obvious that obtaining a good reputation occurs by acquiring many positive reviews. Usually, the orientation of a review is determined from explicit ratings defined by metadata such as the number of stars.

In a related vein, the authors in [86] suggested that the apparent success of feedback mechanisms to facilitate transactions among strangers does not mainly come from their crude numerical ratings, but rather from their rich feedback text comments. In their report [87], the authors, inspired by the idea of average probability of subjectivity in a sentence in a review, proposed the concept of using the standard deviation for a particular sentence within a review in order to test whether the review encompasses a diversity of subjective and objective sentences. This is mainly because reviews have different effects when they are purely subjective, purely objective or mixed between the two types.

2.6.1 Standard Review-based Recommenders

Various studies have been done to exploit sentiment in textual reviews in order to augment ratings in collaborative recommenders [15, 16, 83, 88, 89]. For example, Ganu et al. [15] enhanced RSs by manipulating topic and sentiment information at the sentences level. They estimated ratings from text comments written by users about restaurants on a multi-point rating scale instead of the two bipolar classes: positive or negative. They used a regression model to estimate scaled sentiment points from written reviews. They are the first to integrate sentiment information from reviews into RSs. In addition, Leung et al. [16] presented a framework based on probabilistic sentiment inference, which they named Probabilistic Rating infErence Framework, PREF. They applied natural language techniques to compute sentiment orientation in reviews. In their model, they used a Naive Bayesian classifier to infer ratings, then, they integrated the inference ratings from reviews with a collaborative filtering algorithm to improve the accuracy of suggested items to customers. Peleja et al. [83] inferred ratings from reviews by applying sentiment knowledge to improve the overall quality of RSs. They implemented a multiple Bernoulli classification algorithm to compute probabilistic ratings from reviews. To obtain recommendations, they applied a matrix factorisation approach by adapting users' reviews as a factor to regularise probabilistic estimation.

2.6.2 Microblog-based Recommenders

Due to the exponential growth of information on the Internet, users can easily approach websites and share their opinions in online microblogs. A vast amount of alternatives have become available to users, therefore, it is a new challenge that attracts researchers in the field of RS to investigate the possibilities of personalising recommendations using OSNs features and environments to enhance RSs [4, 34, 90, 91].

Esparza et al. [92, 93] provide an example of incorporating OSNs in the foundation of recommendations. The authors investigated how to obtain recommendations from online microblog services. They presented a solution to exploit the content of short posts written by users as product reviews to over come the insufficient meta-data about items in CB recommenders and ratings in CF recommenders. These posts are used to index the most frequently used and the most important terms to create user and item profiles using the TF-IDF technique described in section 2.4.1. Then a query search algorithm was applied to retrieve relevant item profiles using data from a Twitter-like

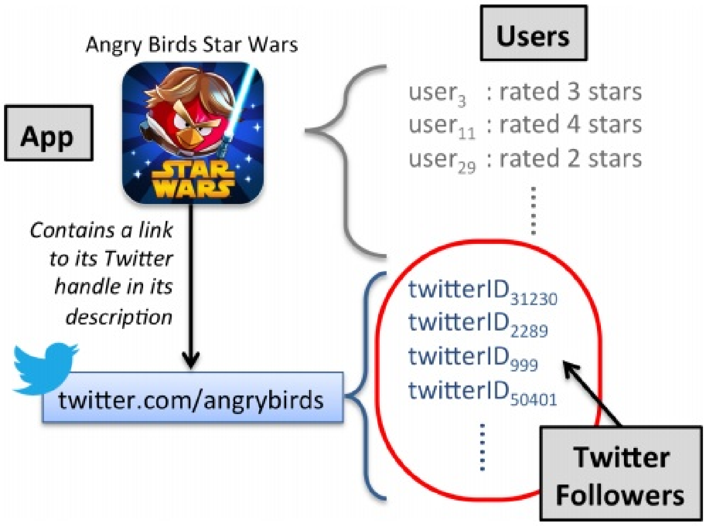


Figure 2.5: Using official accounts of Apps on Twitter in recommendation [4]

review service called *blipper*. This service gave users ability to write short reviews and to rate movies at the same time. This study is similar to our work in terms of utilising microblog's short posts to generate recommendations; however, people using microblogs do not usually rate items, they mainly express opinions. Also, items profiles are built based on a global point of view not local and personalised as all reviews written about an item are combined to construct its profile, and hence the profile is fixed overall community. Another weakness in this study is that they assume that users have rich profiles and they introduce short reviews accompanied with ratings. Therefore, the usual cold-start problem, the case of new users, was not addressed in these papers.

Interestingly, in [4], authors proposed a solution to the cold-start problem in mobile software applications for smartphones (Apps). They used social networking services (SNS), such as Twitter to recognise signals about the new release of these Apps by using Apps' accounts on Twitter. In the example in Figure 2.5, the Angry Birds Star Wars App has an official Twitter account with the handle **@angrybirds5**. They explored the followers list of these accounts, *Twitter-followers*. Their solution was based on a simple averaging technique where the probability of how likely a given user will like a particular App is computed by observing how the Twitter-followers like this App. Given a set of Twitter-followers, the probability that user u likes App a

is computed using the following formula:

$$\begin{aligned} p(+|a,u) &= \Sigma_{t \in T(a)} p(+,t|a,u) \\ &= \Sigma_{t \in T(a)} p(+|t,u) P(t,a) \end{aligned} \tag{2.22}$$

The sign $+$ shows positive interest in the App a, and $T(a)$ represents the set of people who follow the account of App a. However, the main limitation in this work is that users need to follow item Apps on social media, which is not always the case. Another problem with this approach is that the products require official accounts on Twitter so that people can follow these accounts. However, no attempt was made to quantify the trust association between users based on their interactions. In contrast to this study, the scenario in this thesis challenges the more general case for any item, especially when no product account exists on Twitter in very personalised and subjective manner.

2.6.3 Challenges in Sentiment Analysis of Microblogging

OSNs, in particular microblogs, are the focus of this study. Many challenges can be faced when designing frameworks to extract opinions from OSNs, and specifically in the microblog domains, because they do not provide much contextual information and contain implicit knowledge. Another problem is ambiguity since it is difficult to utilise text information as it is in blog posts and comments. In addition, language variation can be an issue since posts include less grammatical and structured text than longer comments. Examples of these irregularities include non-standard capitalisation, different spelling, the use of emoticons, abbreviations and hashtags. Furthermore, the extensive use of humour, irony and sarcasm increases the difficulty for a machine learning algorithms to detect such meanings. On the other hand, these short posts focus more explicitly in the topics and usually the single tweet belongs to one topic [94]. Maynard et al. [94] discussed some more challenges that may be imposed by social media on an opinion mining systems as follows:

- **Relevance**: in social media, discussion may spread over a diversity of topics. Therefore, there is no guarantee that every comment will be relevant to the page topic.

- **Target identification**: this problem happens in sentiment analysis when the retrieved text does not contain any sentiment. What is more, the opinion obtained from the text may be completely different from the search keyword.

- **Negation**: sentiment classifiers which use the simple technique bag-of-words have the weakness that they do not extract negation well since phrases such as "not good" and "good" may be ignored when using unigram models.
- **Contextual information**: social media posts, and specifically tweets, contain a high level of contextual information and level of knowledge that users can obtain compared to standard text. This type of information is not easy to detect.
- **Volatility over Time**: OSNs have a rapidly and dynamic temporal models. Opinions may differ radically over time.

2.7 What is Missing

Through the literature survey, we show in this section our remarks about certain desired features/components which are necessary when designing efficient collaborative recommender systems. These features are specified in Table 2.2 and defined as follows:

- F1: Preserving accuracy level, this requirement comes as a common demand in RSs.
- F2: Supporting user/item cold-start solution.
- F3: Addressing sparsity and increasing the coverage of user-item ratings matrix.
- F4: Using ratings information as the main data source.
- F5: Using SA techniques to infer implicit opinion from products reviews to provide relevant recommendation to a given user.
- F6: Implementing trust-based approaches using WOT or implicit trust based on users.
- F7: Using OSNs as a source of data, such as supporting the idea of using microbloggers from OSNs to augment recommendations.
- F8: Integrating people interconnections from OSNs in recommendations.

Existing work in the literature paid most efforts to include the above features. From Table 2.2 and the discussion of algorithms surveyed in the aforementioned sections, we conclude the following summarised remarks:

Table 2.2: State-of-the-art collaborative recommendation related approaches

Algorithms \ Required Features	F1	F2	F3	F4	F5	F6	F7	F8
Sarwar's approach [8]	√	×	×	√	×	×	×	×
MF-based recommender [42]	√	×	√	√	×	×	×	×
TidalTrust [72, 73]	√	×	√	√	×	√	×	×
MoleTrust [59, 74]	√	×	√	√	×	√	×	×
O'Donovan approaches[71, 78]	√	√	×	√	×	×	×	×
Social trust-based methods[11]	√	√	√	√	×	×	×	×
PREF [16]	√	×	√	×	√	×	×	×
The approach of Peleja et al. [83]	√	×	√	√	√	×	×	×
Microblog-based approaches [92, 93]	√	×	√	×	×	×	√	×
Twitter-based approaches[4]	√	×	√	×	×	×	√	×

- From Table 2.2, it is evident that none of proposed recommendation solutions have addressed all the issues specified in the above features.

- Solutions proposed in memory-based collaborative filtering, such those presented by Sarwar et al. [8, 30], are based on a user-item ratings matrix as the main source of data. This feature, F4, implies the ability to draw recommendations immediately without the need for the complexity of integrating context data. These models support the accuracy feature, F1, but they lack the ability to adapt a solution for the cold-start problem, F2. In addition, these methods are based only on similarity of previous ratings profiles, which ignores large amount of information coming from friends, colleagues and acquaintances on OSNs, feature F7. Is is worth saying that the authors compared rating profiles between people who are strangers to each other, and therefore any explanation of the provided recommendations is absent.

- Most of the proposed algorithms in Section 2.5.2.1 adapt the required feature F6 using explicit trust. They focus on preserving accuracy level, feature F1, by using direct evaluations between users to approach reliable users and mining trust graph to increase coverage, feature F3, in the user-item ratings matrix. However, they did not consider the new users or new items, feature F2. Hence, users need to provide two types of ratings they included in feature F4: (1)item ratings to make predictions, and (2) people trust ratings to build a trust network WOT.

These requirements are highly sparse; therefore this might make algorithms difficult to apply in RS environment. They require exploring new data sources and adapting features F7 and F8.

- Implicit trust can be used to provide more realistic recommendation systems, feature F6, as proposed in Section 2.5.2.2 to compute the hidden trust level within the social connections. To allocate trust values, these methods analyse past rating history stored in the user-item ratings matrix; hence, they support the use of feature F4. They assume that the rating style between users will indicate how much two users are close in their taste by minimising features vectors. These models did not investigate hidden opinions in items feedback, such as item reviews, feature F5, while considering trust.

- Most matrix factorisation-based approaches, Section 2.4.2.2, such as the one reported in [42], focus on acquiring high accuracy, therefore, they satisfy feature F1. Most of the MF extension models target this feature while implementing ratings as the only source of information and thus satisfy feature F4 too. However, they do not consider features F7 and F8 to explore new data sources.

- Most of the work on implicit social trust, such as Ma et al. [11], do not support real world social connections and they claim to investigate OSNs in their design, features F7 and F8. In fact, in their framework, they consider that friends on social networks rate products, and hence, their models are tested on datasets that include direct trust information and direct items ratings by friends. Obviously, this is not the situation in real online communities as people exchange opinions and broadcast messages on different topics and they do not rate products. Consequently, we see that features F7 and F8 of utilising OSNs and exploiting friends' connections are not satisfied by these models.

- Most methods in Section 2.6.1 such as PREF [16], which are based on standard text reviews include sentiment analysis, feature F5, and ratings information, feature F4; however, they cannot be applied to accomplish feature F7. Techniques used to extract opinions from standard reviews are difficult to fit the nature of short posts on OSNs. Online social network providers specify the length of text on these posts, as a result, brief sentences and common social abbreviations should be interpreted to better capture opinions in such environment. On the other hand, in the example paper [83] within the same context, the authors attempted to increase the review-based accuracy, feature F1, by implementing MF models and they assumed that users provide ratings and write reviews; however, these assumptions work against solutions to apply feature F2. Predicting a rating scale from standards reviews was investigated, however, microblogs posts represent more challenging environment to exploring benefits when implementing features F7 and F8.

- In research related to microblog-based recommendation approaches [92, 93], the authors apparently used real time web text and explored microblogger environment in the Twitter-alike service *Bliper*. Therefore, they integrated the feature F7. However, limitations in this work appear in two ways; they applied a term frequency technique to capture relevance to users' tastes, which is not applicable to the nature of short messages on OSNs. Another limitation is that they did not support solutions for feature F2, and their assumptions were based on a rich profile history of ratings.

- Using microblogs such as Twitter in the recommendation context has attracted the attention of a few researchers. The developed solution in [4] was based on finding the list of followers of a certain App product on Twitter. Such models support the use of OSNs in finding out followers who like a particular App in order to extract information that can be used for a given user who follows the account of the App; therefore, this scenario supports incorporating feature F7. The main limitation in this work is that the authors considered that items have unique accounts on Twitter and they ignored user sentiment analysis, feature F5.

- None of the presented solutions propose to boost user preferences in collaborative recommenders using friends from OSNs and hence these do not support feature F8.

- None of the solutions manage both F7 and F8 features, using implicit interactions between users while extracting sentiment from reviews. It is worth noting that it may be more effective to consider implicit trust and sentiment from product reviews in an integrated manner when designing RSs, which are managed separately in literature.

The following section highlights our vision in personalising recommendations using OSNs.

2.8 Vision for the Way Forward

This section presents our point of view to improve the current methods and overcome the existing limitations when designing an efficient collaborative recommender system. Improvements that should be taken into account are drawn as follows:

1. Exploiting the explosive growth of the use of the Internet, in particular, online social networks since these networks hold a vast amount of information and opinions that may guide users' preferences effectively.

2. Reducing the dependence on similarity of rating profiles in recommendation in order to achieve higher diversity levels. However, diversity in suggestions should be based on possible and relevant frameworks to users.

3. Making use of OSNs to eliminate the cold-starts problem and generate recommendations of more relevant items. Focusing on solving the cold-start problem associated with new users is an important concern, more specifically, for users who have not yet rated any items or established any path connections with any other users. These users should be provided with reliable suggestions based on their friendships on OSNs.

4. In order to capture trust, systems should infer personal subjective beliefs towards others rather than asking users to assign direct evaluation to others' reliability.

5. Defining trust between users on OSNs should investigate the implicit behaviour and communications available in these environments.

6. Producing results based on the use of OSNs requires a suitable dataset that represents real world scenarios and includes information about friends' opinions and their connections with users.

7. Leveraging opinions occurring in friends' posts on OSNs needs to be integrated in RSs; for example, considering microblogs' messages as micro reviews is an important factor to enrich user's preferences.

8. Ratings should be extracted from sentiment information held in online posts into scale ratings to be integrated in collaborative recommender systems. Thus, solutions should be applied to fit with the characteristics of the microblog domain.

Overall, these remarks support the view that industries in online business should consider more effective connections with OSNs to boost recommendations. For example, retailers websites may offer their customers the opportunities to tell friends about certain products they have experienced. However, these online industries should find a convenient framework to provide customers with the ability to retrieve their friends' opinions and experience about products. This effort will simulate the real world *word of mouth* scenario in an online basis. Therefore, customers should be able to use the experience of their friends about items, as required, and in particular, to answer the question: *what is the opinion of a friend on OSNs about an item?*

2.9 Chapter Summary

This chapter presented a detailed background on recommendation and provided a thorough survey about existing recommendations algorithms. The chapter also offered an analysis of the advantages and limitations of approaches in related literature. This chapter has described two types of recommender systems which are related to this work; trust-based recommender and review-based recommender. Challenges of these types were also presented and discussed. Then, the chapter highlighted the required features that are needed when designing an effective recommendation approach. Finally, based on the literature survey of state-of-the-art approaches and the identified required features, new ideas that have the potential to help to achieve efficient RSs using OSNs were proposed. This approach is further discussed in Chapter 3.

Chapter 3

Using OSNs in Recommendation: Problem Analysis and Main Ideas

3.1 Chapter Introduction

In this chapter, we describe the main ideas that underline and support our proposed research approach. We first present the current recommendation scenarios, and then based on these scenarios, we introduce our research framework and general characteristics of the proposed approach. We describe every component in the framework including trust modelling, sentiment rating foundation and ratings prediction. The ideas used to build every component in the research framework is identified and discussed.

In detail, the chapter is organised as follows. Section 3.2 describes real world recommendation scenarios. Section 3.3 illustrates the overall research framework. Section 3.3.1 introduces our target micrpblog. Section 3.3.2 discuses our vision in building the trust model. Section 3.3.3 describes sentiment extraction methods, especially from a microblog. Sections 3.3.4 describes the ratings perdition step. Section 3.4 summaries the chapter.

3.2 Recommendation Scenarios

Before explaining our approach, it is important to have an overview about three existing scenarios in providing recommendations:

- In real life, people used to ask for advice from their friends and colleagues. These personal connections provide the required recommendations in *word of* *mouth* basis.

- Over the Internet, RSs provide users with items they may like; these recommenders mainly use information within the system itself, such as the similarity between profiles of registered users or items; e.g., Amazon.com or explicit web of trust in the system e.g., Epionion.com, as illustrated in Figure 3.1. These systems adopt many different approaches as detailed in Chapter 2. For example, a system may utilise the similarity between u_1 and u_2 or trust between u_3 and u_4. However, all users u1, u2, u3, u4 and u5 are registered users within the system. Systems may exploit diversity of information starting from purchasing activity, products ratings, products reviews, ending by communication behaviour to generate personal recommendations.

- Recently, some online businesses started offering a service that registered users can inform friends on other popular OSNs about products they liked or purchased. They use the ultimate feature of Web 2.0, which is free generated content by users, and leverage the ability to share their opinions on microblogs. To this end, we can observe that diverse of information about items and services on social networks can come from what friends have experienced when they tried the products or visited a commercial website. As a result, the existence of this service increases the amount of information exchanged between friends and this may be exploited among friends as a valuable means of recommendations.

However, a different scenario to obtain recommendations can be approached as follows. Instead of consulting similarity between strangers and unknown people registered on the recommender, users can turn to their real friends for help in choosing items among alternatives. More specifically, finding products or services that may gain users' attention can be carried out based on credible sources from the selections users' friends. Investigating friends' selections and deriving the degree of their satisfaction about commercial products require analysis of what friends write to describe their feelings about these items. Microblogs provide easy and simple platforms to allow friends to generate and post instant brief messages describing their opinion about almost everything in the real world. For example, in order to provide the active user in Figure 3.2 with interesting choices, similar to the normal procedure that people follow when they lack of relevant information about items he/she might be interested in may be found in his/her online friends' comments.

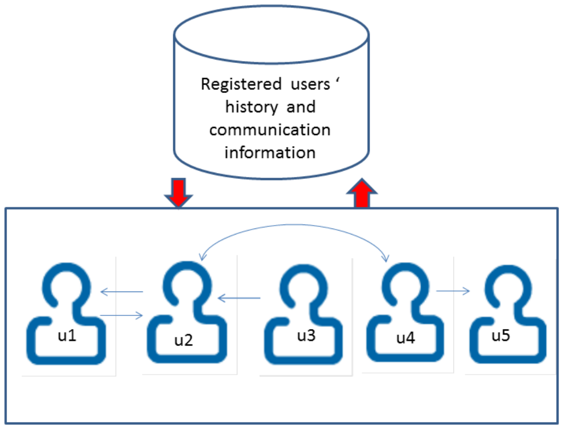


Figure 3.1: Current recommenders consult users' profiles and history

Assuming the domain of movies recommender, as shown in Figure 3.2, Friend A and Friend B can suggest two movies, The Lion King and Cars2, to the active user. Trust relationships are different between the active user and his/her friends A and B. Taking intercommunications activities into account can be used to show who is a more trusted friend, such as the action of re-sending messages received from his/her friends, saving his/her friends' posts in a favourite list or the total number of friends appearing in the friend list. Another challenge is to analyse the level of sentiment held in friends' written posts towards items. For example, if the trust relationship between the user and his/her Friend A is low, and Friend A shows a high positive opinion about a particular movie, while Friend B has a stronger trust relationship with the active user, then friend B's opinion will be highly considered and suggested.

Recommender systems may use their users' friends to help in the recommendation process. An apparent consequence of the exponential growth of information available to users is that recommender systems face two cases; the first is the difficulty experienced by users to find contents that are relevant to their own interests amongst a vast amount of alternatives and the other is the demand for a modern technique which can provide personalised recommendations by exploiting information in the current environment of World Wide Web.

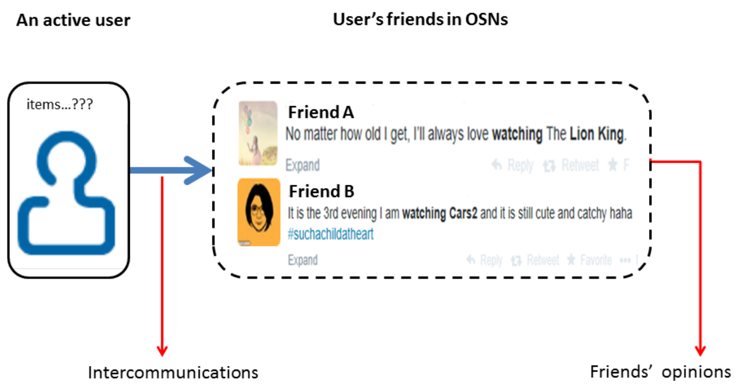


Figure 3.2: Example of user's friends intercommunications in OSNs

3.3 The Proposed Approach: ISTS

From the previous section, we understand the need to explore OSNs to augment users' selections. Friends have great potential to provide valuable items suggestions that may trigger users' attention. From Chapter 2 and features in Section 2.7, we have discussed the limitations in the current RSs literature and the overlooked use of OSNs. Bearing this in mind, we design our novel approach to overcome such gap and support the desired features.

In this thesis, our aim is to model a recommender that can employ a user's OSNs to derive the user's preferences even in the case where no rating information history is available. We therefore propose **I**mplicit **S**ocial **T**rust and **S**entiment (ISTS) based RSs.

General characteristics of the proposed approach (ISTS):

- The proposed approach goes a step further beyond the commercial website and is based on connecting users with friends to consult them about information and experience in order to gain more knowledge that can improve the recommendation process. This is based on that people are willing to receive recommendations from their acquaintances and can be influenced by their opinions even if they have different interests.

- We consider the intercommunications rate as a trust indicator. The strength of a relationship will indicate the strength of trust between peers; i.e., the more frequent communications between friends will be assigned high trust values between them. The higher trust value, the more power will be given to the candidate item in the recommendation process.

- The solution we provide adapts friends' opinions. We attempt to convert friends' opinions into a scalar ratings range, for example from 1 to 5, higher scores indicate higher positive opinion in order to adopt the opinion ratings into the recommendation process.

- Identifying friends' opinions on items and assigning the level of trust between friends shape the final personalised suggested items for an active user.

- In general, we will manage a user's friends list as his/her personal web of trust (WOT). In contrast to the literature, in our approach, the user will not be asked to rate people to prepare the WOT.

- The trust metric and opinion analysis are explored in the context of subjectivity and evaluated using movie recommendation domains.

It is important to mention that our approach is quite general and can be applied in many online businesses along with different OSNs. These networks design different styles of communication behaviour in order to allow users to post messages, re-send posts they like or even save certain posts in a favourite list. Of course, there are more types of interaction between friends, such private messages between them. For example, on the platform of the microblog *Twitter*, re-sending is the actions of re-tweeting and favouring tweets, while on Facebook these activities can be the act of liking others' comments and the action of sharing pictures with a list of friends.

The overall framework we propose is illustrated in Figure 3.3, which shows the steps involved in the model. First, information is extracted from microblog service, as described in Section 3.3.1. Second, after collecting the required data, trust is computed from intercommunication behaviour of users, Section 3.3.2. Third, a sentiment analysis technique is used to infer friends' opinion in the form of scaled rating about items, Section 3.3.3. The final steps are discussed in Section 3.3.4.

3.3.1 The Target OSN: Twitter

To begin with, we shed some light on our target social network *Twitter*. Since Twitter was launched in July 2006, the number of Twitter's users has exceeded 140 million,

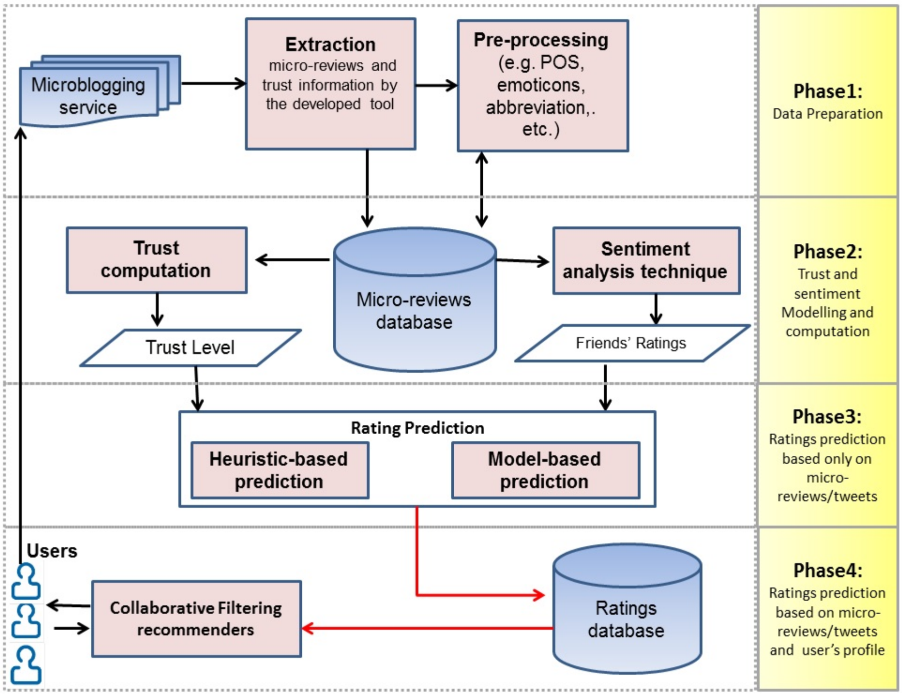


Figure 3.3: ISTS: The Proposed Framework

indicating its huge popularity. In general, it has the facility of allowing users to post brief comments with a maximum of 140 characters, also known as *tweets* about a variety of interests [95]. For instance, one user can describe briefly his/her personal experience in using a particular mobile phone or purchasing any product. We choose Twitter as the OSN for our framework because of the following features:

- Users can manage relationships in the form of lists called followings/followers. These lists show posts from others which reflect their opinions and ideas.

- People can publish their opinions on diverse topics, such as political issues, books, movies, shopping and celebrities, etc; in brief tweets.

- Twitter has a public API supported by many different languages. The availability of this API gives developers the ability to develop software to access the Twitter server and utilise the contained information.

- The wide spread of advanced technological systems like phones and other smart devices allows instant access to Twitter. This enriches the service as it has a wide range of current information available from its users.

In our approach we require to measure the degree of implicit trust between friends. In fact, this type of hidden trust is a subjective aspect that differs from one user to another. However, gathering this type of social information is not an easy task. Therefore, we developed a tool to automatically collect social network data by using Twitter API. This tool extracts the required interactions between friends is called Twitter Interaction Extractor (TIE); further details can be found in Section 4.6. A summary of the difficulties faced during data collection from microblogs can be found below:

- Rules: there are different laws and regulations that many microblog providers apply to protect their users.
- Dynamics: data from real time web is changing dynamically in terms of relations and contents and this is a great challenge.
- Rate limit: many microblog providers apply a rate limit to enforce developers to limit their access and requests within a short period of time.
- Privacy: some users add more privacy constraints to protect their accounts anf forbid public access.

It is important to highlight that we only accessed accounts that are available to the public and did not access any protected accounts for privacy reasons.

3.3.2 Implicit Social Trust

This research presents a novel approach to boost recommendations for users based on their social connections on OSNs. As discussed in Chapter 2, the aspect of social trust is not realistic and not true in many cases in the current trust-based RSs. Eventually, online relations of individuals as represented in RSs are not an accurate reflection of real social friendships [17].

It is worth-noting that the trust aspect in most of the reviewed literature is based either on: (1) analysing common ratings between users towards certain items, or (2) direct trust evaluation produced by users towards each others. However, these assumptions may be correct for some items but not true for other items. For example, having two users *a* and *b*, if user *a* provides a trust score towards user *b* because of common interests in a thriller movie, that dose not guarantee that user *b*'s choices of action movies can also be interested to user *a*. Trust-based RSs provide user *a* by new suggestions from user *b*'s likes, but user *a* decision and rating about the new suggestions can totally be different. In order to present a new perspective for a more effective trust metric, this thesis introduces a metric which is based on how an active user may perceive people in his/her online social networks. The reason beyond is that people tend to ask their relatives and friends about new experience even if they have different tastes, therefore, people are usually influenced by their social connections despite the existence of diversity in preferences. Using this trust perspective, new users to RSs may gain several benefits as follows. Trusted people based on the proposed trust metric may trigger new users' choices and provide interesting items in a better way than anonymous people within the system, which is what trust-based RSs normally do. This trust metric might solve major problems in RSs, such as overspecialisation coming from common ratings. It may also propose new methods in the future to address the diversity challenges in RSs. Hence, we measure trust between people in a new way, different from trust-based RSs.

As we have mentioned before, intercommunications between friends on OSNs can be used to express the amount of trust they perceive about their peers, such as re-tweeting, mentioning other users, favouring others' posts and the number of followers users have [96]. Examples of intercommunication behaviours that we study in the *ISTS* based on Twitter platform include:

- Re-tweeting, which means that a user re-sends a tweet to all his/her friends to show interest, and this is denoted as *RT*.
- There are two types of friends list on Twitter; people who follow a user called *followers* and people who are followed by a user called *followings*. People who have a large number of followers are certainly concerned about sharing their opinions and showing quality in their published comments. As a result, having a friend with a higher number of followers in comparison to his/her followings can draw more trust to friend's tweets.

In study [97], the authors showed that users tend to have a high level of satisfaction when receiving suggestions from friends compared to those coming from strangers in their neighbourhood in traditional RSs. Other works identify users' profile through the identification of the *influence* [98]. For example, Kawak et al. [99] assigned three different influence metrics in Twitter: the number of followers, PageRank, and the number of re-tweets. They found that different levels of influence can be determined depending on the used metric. In another study [100], different metrics were used to define influence: number of followers, the number of re-tweets, and the number of mentions, and again these metrics showed different perspective of influence. In general, we observe that the Twitter features followers/followings and re-tweets information are used as popular influence metrics. Here we design the following goals regarding the trust component in the research proposed framework, ISTS:

- To collect a dataset that includes the required intercommunication behaviours between active users and their friends.
- To select a suitable representative of behaviours in OSNs that indicate trust.
- To define the an appropriate method to model the trust metric.
- To demonstrate how this trust metric can contribute to the final prediction of recommendations and ratings in the case of users who have ratings profiles as well as the new users.

3.3.3 Extracting Sentiment

In this section, we introduces an important ISTS component: sentiment held in friends' brief posts, as shown from Figure 3.3. As discussed in Chapter 2, recommendations extracted from users' product reviews do not employ any advantageous metrics that can be harnessed to personalise recommendations from friends on OSNs such as Twitter.

In the task of text processing, the piece of text should be converted into feature vectors, and this step is intended to make important features available to use. There is an extensive body of research that solve the problem of feature selection for machine learning solutions. In the following, we will illustrate some features engineering in sentiment analysis.

Term presence vs. frequency: Term frequency has classically been emphasised in standard information retrieval, such as the popular technique TF-IDF weighting, which presents reasonable results. However, Pang et al. [14] showed better performance using term *presence* in comparison with term frequency, where the input values increase with the occurrence frequency of such keywords. This means that binary-valued feature vectors in which the input indicates whether a keyword occurs (value 1) or does not occur (value 0) showed more effectiveness in the domain of review polarity classification than real-valued features vectors. This finding sheds the light on the difference between topic-based document categorisation and polarity classification. Determining a topic is more likely to be affected by frequent occurrences of certain terms; however, the overall sentiment may not often be detected through the number of occurrences of the same terms.

Terms positions: The position of a token within a document can have an effect on the overall sentiment. Hence, position information is sometimes encoded into the feature vectors [14, 101]. Within the debate of the usefulness of higher-order n-grams, Dave et al [85] found that in some cases, bigrams and trigrams perform better in product-review polarity classification, while Pang et al. [14] found out that unigrams outperform bigrams in classifying movie reviews in terms of sentiment polarity.

Part of speech (POS): This is a type of information that can be useful in the domain of sentiment analysis and opinion mining. Adjectives have been used as a feature by several researchers. For example, work on determining subjectivity showed a strong correlation between the presence of particular adjectives and sentence subjectivity [102]. This result explained that certain adjectives are effective in defining sentiment. Other work compared the performance of adjectives, verbs and adverbs where more sub-categorisation may play a significant role [103].

Negation: Considering negation is a major concern in opinion and sentiment analysis. It is clear that negation tokens like (no, not) when added to a phrase, can change the overall sentiment to the opposite class. Some work coded negations directly into the feature vectors. For example, in the sentence "I don't like cooking" the token like will be changed into a new token "like-not". Another problem which may appear in modelling negation is that sometimes negation is expressed within a context of irony and sarcasm. These expressions are not easy to detect [104].

It is worth mentioning that studies in the domain of products review RSs differ from our work in the following aspects. Deriving users' opinions from written reviews about commercial products has not been fully applied using OSNs to harness more personalised suggestions from friends comments. Even though these networks hold a huge amount of opinions and interests about a diverse range of items and services, such as books, movies and restaurants, etc. Further, due to the inherent shortness of online comments and the restricted use of characters in the OSNs environment, methods like term frequency to index user's profiles in standard long reviews might not be the most appropriate algorithms. For example, reviews sites for restaurants include many headers to guide the customer when they write reviews, which include location, flavour, menu, etc.; this gives a more structured long text that allows the customer to describe their opinions in detail. Moreover, in this thesis, we considered the informal use of language features on OSNs , such as emoticons signs as we believe that they hold a very clear picture about opinions in the text. For example, in the tweet: *Jumanji ;) watching it since long time :D*, the signs *;) and :D* in such tweets contain a highly positive opinion about the mentioned movie, *Jumanji*. It is hoped that, by this way, we can represent opinions in a more profound manner in such an environment. In addition, it may be more effective for RSs to measure the degree of sentiment and opinions with a multi-point scale of ratings than the traditional classification to negative or positive polarities [18]. Measuring sentiment on a multi-scale point, in brief and informal language features is one of the most challenging aspect raised in this research.

Here, we design specific goals regarding the component of sentiment of short posts in the ISTS as follows:

- To provide proof that using such microblogs posts is achievable and to collect suitable tweets regarding recommendation application in this study.
- To define the ultimate features that build up short posts.
- To implement an appropriate method that converts these posts into scalar ratings and this will represent the friends' posts in the form of numerical evaluation.
- To use this inferred numerical evaluation to support the final decision about unknown items to active users in both new users and users with a history of ratings.

3.3.4 Ratings Prediction

From previous sections, we described how trust is intended to be modeled and how friends' sentiment ratings will be computed. Now, we move to the prediction task in the proposed research framework, illustrated in Figure 3.3. The crucial step in the function of RSs is to generate rating predictions for unknown items. The research approach proposed by this thesis predicts ratings for two types of users.

For new users without any history of ratings: new users will receive recommendations based mainly on their connections on OSNs. Users will be recommended personalised items derived from their trusted friends' opinions. Heuristic methods and machine learning models are developed for this task. Experiments are conducted to evaluate their performance in terms of accuracy of recommendations; this is described in more detail in Chapter 4 and 5.

For users who already have a history of rating profiles: in this case, users have known tastes and preferences towards some items. Consequently, users' items ratings stored in the system should not be ignored and should be used. Users will receive recommendations based on a combination of their known tastes and their trusted friends' preferences. In other words, we need to fuse users' own preferences and their friends' opinions using collaborative filtering techniques, including user-based and item-based methods. Evaluations and results include coverage and diversity analysis discussed in Chapter 6.

Here, we develop specific goals to satisfy the component *ratings prediction* in our ISTS framework as follows:

- To predict ratings for missing items employing OSNs as a new source of data in the proposed ISTS recommendation approach.
- To improve the prediction performance using different machine learning algorithms.
- To provide new users, with no profile of preferences, with relevance predicted items based on their friends on OSNs.
- To integrate OSNs with users' rating profiles in collaborative filtering recommendations domain.
- To perform the integration and demonstrate the performance by evaluating the accuracy, the coverage and diversity challenge foundation.

3.4 Chapter Summary

This chapter has given an overview of the proposed ISTS approach. We have pointed out how to identify user's preferences derived from reliable friends' opinions in real world recommendation scenarios. We have also described the overall framework of our approach and general characteristics related to ISTS. Then, the main idea of every component in the framework were explained. Starting by presenting our target microblog, and some motivations from Twitter services related to our research context were described. Then, moving to the trust component, in which implicit social trust is modelled and inferred from intercommunications with friends on microblog platforms. Furthermore, we have described how the sentiment component can be extracted from friends' short posts, tweets. Finally, we have highlighted the ratings prediction component for new users and for users with ratings history. The goals to be satisfied for every component were designed in this chapter. The next chapter highlights the technical aspects of the proposed framework.

Chapter 4

ISTS: Implicit Social Trust and Sentiment based Approach to Recommender Systems

4.1 Introduction

This chapter provides a technical description of our novel approach ISTS, which is designed to integrate users' OSNs with recommender systems. The chapter demonstrates how the novel ISTS framework supports trust features and analysis of sentiment identified from users' friends posts.

The chapter starts by highlighting the preliminary requirements, in Section 4.2. Section 4.3 shows how we consider and compute trust in our approach based on one feature (re-tweet) and then two features (re-tweet and followings/followers). Section 4.4 describes sentiment analysis methods, bag of words and probabilistic techniques.

Section 4.5 illustrates how item rating prediction was computed. Section 4.5.1 describes our proposed ISTS framework considered from a heuristic perspective. Section 4.5.2 discusses the ISTS framework based on classification models. Section 4.5.3 discusses our novel ISTS by implementing regressions models. Section 4.6 demonstrates the experimental settings in terms of dataset description and the metrics used in the evaluations. Section 4.7 evaluates the results and compares performance with the most related literature work. Section 4.8 presents the needed improvements to the proposed ISTS. Finally, Section 4.9 provides a summary of the chapter.

4.2 Preliminary Considerations

This section presents the main research assumptions and notations used in modelling the ISTS method.

4.2.1 Main Assumptions

In this section, we introduce the main required assumptions that were used in ISTS. Let us assume that new users are recently registered. These new users usually answer some directed questions during the registration steps. Nowadays, most people normally have OSNs accounts. Information about users' Twitter accounts are collected in the registration process. Normally, the system will have the ability to link and access their accounts and obtain the desired information. Consequently, systems can store this information and record these logs regularly, for example every week or every month. The allowable length of access time this information and the update frequency are all controlled by business service providers and OSN providers, such as Twitter.

In the ISTS, systems have to search two types of information:

- Friends' tweets that include opinions about missing items.
- Intercommunications actions between the active user and his/her friends who post tweets about the missing items.

4.2.2 Notations

The notations used in this chapter are summarised below.

- $List_{in}$: the number of followers of the active user's account.
- $List_{out}$: the number of followings of the active user's account.
- mr: the micro-review.
- sw: a sentiment word.
- MR: a set of micro-reviews..
- $P_{(f,mr)_i}$: positive words in a mr written by friend f who writes mr.
- $N_{(f,mr)_i}$: negative words in a mr written by friend f who writes mr.

- *wc*: class of ratings that a sentiment word belongs to.
- C: the class set of the used ratings.
- $senti_{wc_j}$: the mapped weight of words from SentiWordNet [105].
- TF: total group of trusted friends.

4.3 Implicit Trust

4.3.1 Implicit Trust Method: RT

We believe that interactions between friends on OSNs can indicate the level of trust they may hold towards one another [106]. We start with the belief that the strongest indication of showing an interest in friends' tweets is the action of re-tweeting which means that a user re-sends a tweet to all his/her friends to emphasise the interest, this action will be denoted as RT. Formally, we want to compute the trust relation between user u and one friend f among the group of friends F, with $f \in F$. Intuitively, trust is identified as a normalised average:

$$trust_{u,f} = \frac{RT_{u,f}}{RT_{u,F}} \tag{4.1}$$

where we denote trust between user u and friend f as $trust_{u,f}$, whereas $RT_{u,f}$ is the number of messages re-tweeted by user u to friend f in a given period of time and $RT_{u,F}$ is the total number messages re-tweeted by user u to all friends in the group F in the same period of time.

Due to the fact that people's interactions vary over time and relations are not static, we define the periods of time as $T = \{t_1, t_2, ..., t_W\}$ and then the same computation of trust in Equation 4.6 is applied for each time period $t_j \in T$ as follows:

$$trust_{u,f}(t_j) = \frac{RT_{u,f}(t_j)}{RT_{u,f}(t_j)} \tag{4.2}$$

Based on the above equation, we can detect the level of trust between user u and friend f over the entire period of time T as follows:

$$TRUST_{u,f}(T) = \frac{1}{T}\sum_{j=1}^{T} trust_{u,f}(t_j) \tag{4.3}$$

where $TRUST_{u,f}$ refers to the total level of trust over certain periods of time between u and f. In fact, deciding the number of time periods to test logs for trust is a domain specific decision. This decision is influenced by many criteria, for example, the efficiency of the database engine used by the online business and the amount of information allowed to be exchange between the business and OSNs services.

4.3.2 Implicit Trust Method: *RT* and *L*

As we have mentioned before, intercommunications between friends on OSNs express the amount of trust they perceive about their peers. Intercommunication behaviours that ISTS integrates on the Twitter platform are:

- Re-tweeting action (RT) which means that a user re-sends a tweet to all his/her friends to show interest. We compute trust relation between user u and one friend f among the group of friends F, $f \in F$. In this approach, we compute the re-tweet action RT in the next equation for every time period $t_j \in T$:

$$RT(t_j) = \frac{RT_{u,f}(t_j)}{RT_{u,F}(t_j)} \tag{4.4}$$

- There are two types of friends lists on Twitter. As a result, having a friend with a higher number of followers compared to his/her followings can draw more trust to micro-reviews written by such a friend. This ratio between the number of followers and followings is denoted as L. We can obtain the value of L at time t_j as follows:

$$L(t_j) = \frac{List_{in}(t_j)}{List_{in}(t_j) + List_{out}(t_j)} \tag{4.5}$$

  where $List_{in}$ indicates number of followers and $List_{out}$ indicates number of followings. L demonstrates the percentage of the number in $List_{in}$ in relation to the total number of both lists. Every relation between user u and friend f can be affected by the value of L.

Trust is then identified as a normalised average between the two factors RT and L at a ceratin time t_j as following:

$$trust_{u,f}(t_j) = \frac{1}{2}(RT(t_j) + L(t_j)) \tag{4.6}$$

where we denote trust between user u and friend f as $trust_{u,f}$ and RT is the parameter defined in Equation 4.4. It is important to mention that the RT and L are given the same power impact, however, more investigation of their different degree of impact will explained later on Chapter5. Computing $TRUST$ at entire period of time T can be obtained based on Equation 4.3.

4.4 Sentiment Analysis

It is important to mention that in this thesis we introduce a sentiment analysis algorithm that is different from other sentiment algorithms as the following. Most of the existing methods consider sentiment analysis as a classification problem in positive or negative classes [18]. In our approach, we need to provide sentiment in a scalar rating format in order to be used in the RSs application. Rosenthal et al in [18] indicated that studies should focus on identifying sentiment on a rating scale in future studies to be used on popular websites such as Amazon, TripAdvisor, Yelp, etc. They explained that obtaining sentiment in a rating scale should move from a binary classification to regression. In addition, the challenge encountered in relation to sentiment is to apply sentiment analysis to short and informal posts on OSNs, which is different from long standard reviews of products which is rich by words that hold sentiment. We need to consider all the informal components of these short posts such as emoticons. The results of the proposed sentiment analysis are integrated with trust values to predict the friends' ratings about items, and hence the sentiment analysis methods are evaluated implicitly when evaluating the predicted ratings. However, we have evaluated our approach against a method that is based on the probabilistic theory to infer multi-point scale of ratings from standard reviews for RS presented in [16].

4.4.1 Sentiment Analysis: a Bag of Words Approach

On Twitter, we deal with *tweets* as *micro-reviews* denoted as mr and any mr consists of sentiment words sw, expressed as $mr = \{sw_1, sw_2, ..., sw_m\}$. In addition to these sentiment words, people on OSNs widely use special language features and signs to reflect their emotions and interest in any topic, such as emoticons, hashtags and capitalisation. Hence, every special Twitter's language feature is considered as either negative or positive sw. Table 4.1 explains the special language features we used. In addition, we consider negative and positive opinion words listed in [14] that were applied successfully in standard movie reviews. These words are: *love, best, great, still , bad , boring, !, ?* and the word watching to guide the search to movie items. However, we eliminated any word that has not appeared in our collected tweets.

Table 4.1: A list of negative and positive Twitter features

Positive Features(+)	Negative Features (-)
+ Intensifier: e.g., cooool	- Intensifier: e.g., sleeeepy
+ Emoticon: e.g., :)	- Emoticon: e.g., :(
+ Hashtag:e.g., #Priorities	- Hashtag: e.g., #sadness
+ Abbreviation:e.g., LOL	- Abbreviation: e.g, NSB
URL	Negation

In order to extract the described features from friends'*tweets* and build a bag of words, we applied some necessary preprocessing steps, as demonstrated in Figure 4.1, (1) Tokenisation stage is where we segment tweets into separate words using punctuation marks "," and spaces based on the feature extraction unigram model because this type of model has proven to perform better than bigram ones particulary in movie reviews [14]. (2) Normalisation includes removal of stop words which are the articles, such as "a" and "the" as well as removing user-names from tweets. Negation is also tackled at this stage, and is marked by the word "NOT" to indicate the presence of negation. Any referral link will be indicated by a "URL". We keep the name of the movie to help with understanding the sentiment polarity of words. In addition, we use an online lexicon to identify the polarity in emoticons and abbreviations, such as the emoticon ":)" and abbreviation "OMG". For a more comprehensive emoticon list, see en.wikipedia.org/wiki/EmoticonPositive and negative intensifiers are also accounted for, such as "cooool". (3) Binary vectors of features are implemented manually for every micro-review and existence of the *sw* will be given a value 1, whether positive or negative, otherwise it will take the value zero.

In contrast to binary sentiment methods that provide only one polarity of reviews either as negative or positive, the ISTS requires to obtain more precise sentiment analysis that describes more than a simple like or dislike. Both the natural shortness of tweets and the requirement for finer-grain of sentiment analysis draw a challenging task. For a given set of micro-reviews $MR = \{mr_1, mr_2, ..., mr_n\}$, every mr_j is represented by a set of sentiment words $mr_j = \{sw_{j1}, sw_{j2}, ..., sw_{jm}\}$. ISTS needs to infer a sentiment rating $\hat{sr}$ that holds a class of ratings $\hat{sr} = \{sr_1, sr_2, ..., sr_S\}$, for example

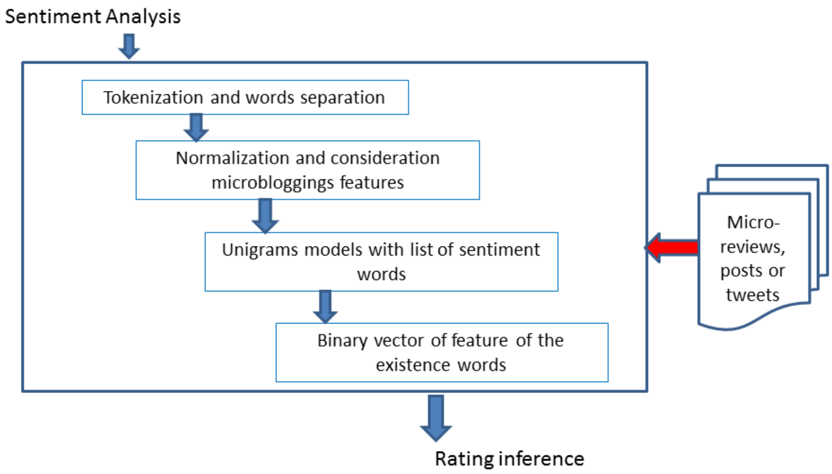


Figure 4.1: Sentiment Analysis preprocessing steps

$\hat{sr} = \{1,2,3,4,5\}$, since the goal is to allocate a rating to describe the strength of an opinion in micro-reviews. Now, sentiment ratings $\hat{sr}$ can be inferred by aggregating all existing of positive sw's normalised by the total number of existing features previously mentioned, which can be either positive or negative, which is in line with a similar method used the authors of a previous study [15]; however, they worked only at sentence level while the proposed ISTS works at the word level due to the shortness of micro-reviews and its ability to support the special language features of OSNs. The following equation illustrates how we compute the inferred sentiment rating sr from mr:

$$sr_{f,i} = \frac{P_{(f,mr)_i}}{P_{(f,mr)_i} + N_{(f,mr)_i}} * S \tag{4.7}$$

where $P_{(f,mr)_i}$ and $N_{(f,mr)_i}$ are the positive and negative features regarding item i presented in friend f' mr_j. S is the number of the class categories used in the recommendation; for example, some systems base score ratings on a five or ten-point scale. This can be further explained by the equation:

$$P_{(f,mr)_i} = |P_{sw \in mr_j}| \tag{4.8}$$

where $|P_{sw \in mr_j}|$ is the number of positive sentiment words sw appearing in mr_j, and similarly:

$$N_{(f,mr)_i} = |N_{sw \in mr_j}| \tag{4.9}$$

where $|N_{sw \in mr_j}|$ is the number of negative sentiment words sw in mr_j.

From Equation 4.7, in the ISTS, S should be 5 in order to measure the strength of an opinion in the popular five-point scale.

As shown in Figure 4.2 sample of the tweets about movies are given ratings, and bolded terms are the opinion words that we considered. The reason behind this finding may be because people express sentiment in standard reviews differently and use informal language when using OSNs.

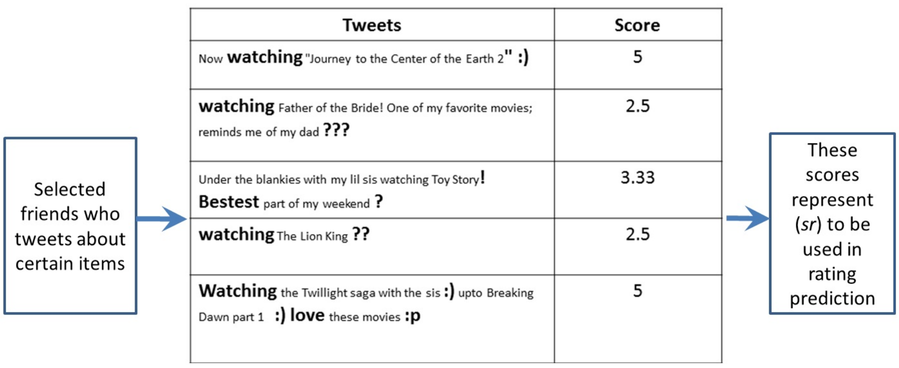


Figure 4.2: A sample of tweets and the calculated *sr* based on bags of words

4.4.2 Sentiment Analysis: Probabilistic Approach

In this study, we present the use of friends' short posts on OSNs, in particular; friends' *tweets* on Twitter to consult friends about their interests and choices as people do in the real world. It is particularly challenging to extract sentiment from tweets because : (1) tweets are inherently short and use more informal language compared with long standard reviews of products; (2) these are user free generated-contents text style. Again we deal with widely used special language features and symbols on OSNs that people add to their comments to reflect emotions about topics, including emoticons, intensifiers and capitalization, etc.

In order to extract and measure sentiment from friends'*tweets* we applied some necessary feature extraction pre-processing steps:

- Tokenisation step, where tweets are segmented into separate words using punctuation marks ”,” and spaces based on a feature extraction unigram model with proven effectiveness as stated above.

- Normalisation process was done by removing stopwords, including articles and removing user-names from tweets as stated above.

- Word-families that we consider in this model are adjectives, verbs and adverbs, to allocate strength. This is the **POS tagging** step.

- Negation is also tackled, and is marked by the word ”NOT”. Negation tagging is important, and therefore, we add the tag -NOT to other words that appear in the context of this negation within the same sentence. For example, *good* in sentence includes *it is not good*, will mark as good-Not. Words denoting negating also include words, such as no, never, and neither, this method follows the model proposed in [14].

- In order to know the intensity of opinion that words can hold we use SentiWordNet [105]. This well-known lexical resource assigns weights to words in three polarities: positive, negative, neutral.

- We use the online lexicon to identify the sentiment in emoticons and abbreviation (en.wikipedia.org/wiki/Emoticon) such as the emoticon ”:)”. This can also be applied to abbreviation ”OMG”. Positive and negative intensifiers are also counted such as ”coooool”.

- We considered synonym words of emoticons to allocate sentiment weights. These informal shortcuts and emoticons hold a vast amount of sentiment and information about opinions.

- For capitalisation and intensifiers, which denote emphasis, we double the allocated SentiWordNet weight of these feature words.

- We also consider the meaning of abbreviations that are used based on online lexicon.

- Words used in hashtag are considered if they are not or an item name; i.e.; only if the hashtag consists of a verb, an adjective, or an adverb.

- Any referral links in micro-reviews will be indicated by a ”URL”, as stated above. A URL is also considered a positive indication of expressing opinion about watching a movie or doing an activity.

Formally, for a given set of micro-reviews $MR = \{mr_1, mr_2, ..., mr_n\}$, every mr_j would contain a set of sentiment words $mr_j = \{sw_{j1}, sw_{j2}, ..., sw_{jm}\}$. These words are defined through the feature extraction steps. We represent every sw by a weight illustrating the sentiment intense $senti$ derived from SentiWordNet [105]. There are two reasons for using these weights and for labelling opinion orientation manually: (1) SentiWordNet has been widely used in several previous studies such as [83], and (2) the focus of our work is not on extensive linguistic processing. We want to infer sentiment rating SR to hold a rating describing the overall strength of an opinion expressed in a micro-review where $SR \in C$ and $C = \{c_1, c_2, ..., c_m\}$. Let us assume that C=$\{1,2,3,4,5\}$, where five represents the most positive rating and one is the most negative sentiment.

Based on SentiWordNet there are three different weights for three polarities: positive, negative and neutral. First, we decided whether a particular word's polarity was either positive, negative or neutral. Thereafter, we choose the weight under the selected polarity. However, there are only three sentiment class polarities so we need to further extend these sentiment word classes, denoted as $wc \in WC$, where $WC = \{wc_1, wc_2, ..., wc_k\}$. In our case WC=$\{1,2,3,4,5\}$ so we map whether the word belong to a particular sentiment word class wc_j based on SentiWordNet weights and the predefined polarity of the word as follows:

$$wc_j = \begin{cases} 5, & \text{if } PositivePolarity \quad \& \quad senti_{wc_j} >= 0.5 \\ 4, & \text{if } PositivePolarity \quad \& \quad senti_{wc_j} < 0.5 \\ 2, & \text{if } NegativePolarity \quad \& \quad senti_{wc_j} < 0.5 \\ 1, & \text{if } NegativePolarity \quad \& \quad senti_{wc_j} >= 0.5 \\ 3, & \text{otherwise} \end{cases} \tag{4.10}$$

where $senti_{wc_j}$ is the weight from SentiWordNet dictionary that belongs to the class wc_j.

Secondly, inspired by the Naive Bayes (NB) classifier used in [16], we infer a probabilistic-based sentiment rating SR as follows: for every mr we compute the total intensive of a particular wc_j; in other words, we sum up all the weights that belong to a particular wc_j as follows:

$$SI_{(mr,c_j)} = \Sigma_{c_j=wc_j}(senti_{wc_j}) \tag{4.11}$$

Here, $SI_{(mr,c_j)}$ is the summation of all sentiment intensity weights that belong to a particular wc_j, where $senti_{wc_j}$ is the intensity weight of a word with respect to its sentiment class.

We assign an estimated value to mr, denoted as $E(mr,c_j)$, according to wc_j as follows:

$$E(mr,c_j) = P(c_j)SI_{mr,c_j} \tag{4.12}$$

where $P(c_j)$ is the prior probability of class c_j in the dataset, $E(mr,c_j)$ is an approximate estimated value that shows how likely mr is to belongs to sentiment class c_j. And SI_{mr,c_j} is the strength of a certain sentiment class c_j in mr. We calculate the prior probability of each class category c_j from the training examples as follows:

$$p(c_j) = \frac{N(c_j)}{N} \tag{4.13}$$

where $N(c_j)$ is number of examples belonging to class c_j relative to the total number of examples N in the dataset.

Thirdly, after computing $E(mr,c_j)$ for every $c_j \in C$, we need to show the strength of each class in relation to other classes by normalising $E(mr,c_j)$ so their sum will be equal 1:

$$E'_{mr,c_j} = \frac{E(mr,c_j)}{\Sigma_{c_i \in C} E(mr,c_i)} \tag{4.14}$$

At this point, we can predict the overall friend's opinion by computing the sentiment rating SR of mr, denoted as $SR(mr)$, as follows:

$$SR(mr) = \Sigma_{c_j \in C} \quad c_j * E'_{mr,c_j} \tag{4.15}$$

where c_j is the numerical label j'th class of the set C. Similarly to the probability theory we consider E'_{mr,c_j} to be the probability that mr has a rating c_j in order to compute the Expected Value. This consideration allows the predicted rating in our framework to hold a continuous class value which demonstrates a more continuous natural relation between class labels. In other words, SR assigned to mr can fall within two adjacent classes in the set C; for example, SR may be predicted to be 3.75.

4.5 Item Rating Prediction

Having a dataset, $SD = \{trust_{u,f}, SR_{f,i}\}$, which includes trust between user u and friend f, $trust_{u,f}$, and the friend's opinion about item i, $SR_{f,i}$ is required to identify the relations between user-rating and feature terms, denoted by IR which illustrates the impact of each friend's opinion $SR_{f,i}$ on a user-rating about item i, denoted as $IR_{(u,f,i)}$ and is computed as followings:

$$IR_{(u,f,i)} = g(trust_{u,f}, SR_{f,i}) \tag{4.16}$$

As mentioned before g is an unknown function that we need to identify. Let the set TF refers to a set of friends who have a trust relationship with user u and an opinion about item i, $TF = \{(f,i)_1, (f,i)_2, ..(f,i)_m\}$. We aggregate the impact function IR to obtain the final user ratings as:

$$R_{u,i} = \frac{\Sigma_{f \in F} IR_{(u,f,i)}}{|TF|} \tag{4.17}$$

The above equation can be simplified as:

$$R_{u,i} = averageIR_{(u,f,i)} \tag{4.18}$$

We can see that Equation 4.19 is a special case of Equation 4.16 when it is defined as follows:

$$g(trust_{u,f}, sr_{f,i}) = trust_{u,f} * sr_{f,i} \tag{4.19}$$

4.5.1 Heuristic-Based Model

Initially, we wish to generate rating predictions heuristically for the unseen items by new users. More formally, let us assume that the system needs to identify a rating R for an active user u about an item i, denoted as $R_{u,i}$. The trust values will act as weights to vote for the derived sentiment ratings sr. Let set TF refers to the set of friends who have a trust relationship with user u and opinion about item i, $TF = \{(f,i)_1, (f,i)_2, ..(f,i)_m\}$ and hence $TF_i \subset F$. The following formula shows how the ISTS would predict the rating $R_{u,i}$ as a weighted average:

$$R_{u,i} = \Sigma_{f,i \in TF} \quad \frac{trust_{u,f}}{|TF|} * sr_{f,i} \tag{4.20}$$

TF_i represents the trusted group of friends who have expressed opinions about item i among all friends F. And $sr_{f,i}$ is the extracted sentiment rating from friend f about item i. And $trust_{u,f}$ refers to the trust value between a user and a friend who posted information about item i.

We have also the following properties:

1. Trust between one user u and his/her friend f may not be bidirectional, hence:

   if $u \to f$ and $f \to u$ then $trust_{u \to f} \nLeftrightarrow trust_{f \to u}$

   this means it does not necessarily implied that the relative trust will be at the same level between user u and friend f towards each other.

2. If $trust_{u,f_1} > trust_{u,f_2}$ then, sr_{f_1} contributes more to $R_{u,i}$ than sr_{f_2}.

3. $MIN_{f \in TF}(sr_{f,i}) \leq R_{u,i} \leq MAX_{f \in TF}(sr_{f,i})$ where $MIN_{f \in TF}(sr_{f,i})$ and $MAX_{f \in TF}(sr_{f,i})$ are the lowest and highest computed sr towards item i among F .

4.5.2 Classification-Based Inference Ratings Models

One of the aims of this work is to assess whether information coming from friends can contribute to estimating products ratings by using classification learning algorithms.

This section focuses on classification models used to estimate a final rating decision about an item for new users using $trust_{u,f}$ and $sr_{f,i}$. We conducted experiments with three selected algorithms: Naive Bayes classification, Logistic regression and Decision tree. The importance of these algorithms comes from yielding satisfactory results in different domain and the availability of related software tools [107]. The core engines behind these three classification algorithms are quite different. Next paragraphs provide brief details about each.

These three well-known machine learning models are used to represent or approximate the unknown function g as the output will be one of the class ratings $C = \{C_1, C_2 ... C_m\}$ and feature vectors $V = \{v_1, v_2 ... v_n\}$. The estimation results will be one of five nominal classes of ratings *ExtremeDislike Dislike, Like, Neutral, ExtremeLike*.

Naive Bayes (NB) classifier: it is a probabilistic approach for solving classification problems. In general, Bayesian classifiers are statistical classifiers. NB assumes that a feature affects a given class independently from other features. It assigns a class rating C_i to a given example x by calculating the probability to assign this class C_i to x as

$P(C|x)$. NB has certain constraints to allocate a class C_i if and only if:

$$P(C_i|X) > P(C_j|X) \quad for \quad i,j \in \{1,..m\} \tag{4.21}$$

Using the Bayesian rule to derive the posterior probability $P(C_i|X)$ can be done using the following formula:

$$P(C_i|X) = \frac{P(C_i)P(X|C_i)}{P(X)} \tag{4.22}$$

where P(X) plays no role in choosing C_i. Probability $P(X)$ will not be changed for all classes and so we need only to maximise $P(X|C_i)P(C_i)$. There are some premises to apply NB. It is a simple technique to use and tends to be optimal for particular domain classes with highly independent and irrelevant features. Moreover, the probabilistic nature of NB allows it to handle missing values [108]. We describe two more algorithms that might achieve better results.

Logistic Regression is another model we used. Generally speaking, regressions help in learning weights to be associated with each rating class in order to demonstrate the predictions obtained from relationships. Logistic regression is also considered as a linear model with more power to refine the parameters by minimising the error function. The output can be interpreted as a probability that gives confidence that a prediction belongs to a certain class. If we assume m is the number of classes $C_j = \{C_1, C_2...C_m\}$ and a number of examples with n number of features. The parameter matrix B is calculated as $n*(m-1)$. The probability that C_j being allocated with the exception of the last class, is:

$$P(C_j|X_i) = \frac{exp(x_i B_j)}{\sum_{j=1}^{m-1} exp(x_i B_j) + 1} \tag{4.23}$$

More details about ridge estimators in logistic regression can be found in [109].

A **Decision Tree** model is also applied. It is a nested set of rules used to split data. This recursive algorithm constructs a tree structure automatically starting from a root feature and ending with leaf nodes. When splitting data, a decision rule is applied for every feature, and then the feature that minimises the cost function is chosen to build the tree branches. The leaf node at the end of each branch is a class. There are many decision tree algorithms in the literature, in this work, we adopt C4.5 algorithms. The metric used to measure the best splitting of data by C4.5 algorithm is called information gain (IG) measurement, derived from the dataset itself to split the tree branches. Let

p_i be the probability that a subset of SD is labeled by c_i, then:

$$I(p) = -\Sigma_{c_i \in C} p_i \log_2 p_i \tag{4.24}$$

Then, we need to compute entropy $E(v_i)$ which gives the expectation information given when splitting by feature v_i when this v_i takes value a_i as one of possible values taken from a set $vals$ $a_i \in vals$:

$$E(v_i) = \Sigma_{a_i \in vals} \frac{|x \in SD|x = a_i|}{|SD|} I(p) \tag{4.25}$$

Building a classifier using a decision tree is very attractive due to several advantages. For instance, it is not expensive to construct and it has a fast computation time when doing classification. Another important advantage of decision trees is that we can obtain a set of rules that are easy to interpret when applying accuracy comparison to other well-known classification methods. More discussion about the area of decision tree can be found in [110]. We used Weka [1] library for building and model learning of all the aforementioned models.

4.5.3 Regression-Based Inference Ratings Models

One of the aims of this research is to investigate whether information coming from OSNs can be used to estimate product ratings by using regression learning algorithms. We conducted experiments using three selected regressions algorithms: Linear Regression, Random Forest and Support Vector Regression. Previously, these algorithms achieve good performance in many regression prediction applications with several software tools available to facilitate applying and comparing results. In addition, we have applied these three different regression algorithms to generate a clear picture of the behaviour and performance of the proposed model under these regressions. By doing this, evaluation of this model can be better explained when comparing it with different scope of algorithms, such as classification algorithms reported in [91]. It is also hoped that the findings can show better understanding of what type of regression is most accurate to guide the selection of the right method in the intended applications.

The three well-known machine learning regression models are used to represent the unknown function g as the output will be the estimation of $R_{u,i}$. In fact, the core engines behind these three algorithms are quite different. The next paragraphs briefly

[1] Weka3.6.9

describe each of these models.

1. *Linear Regression*: statistically, linear regression is an approach which describes the relationship between a scalar dependent variable and one or more independent variables. *Simple linear regression* uses one independent variable, while the case of *multi linear regression* considers more than one independent variable. Generally, all real-world regression models are based on multiple independent variables. Linear regression models data use a linear predictor function and estimate unknown model parameters. Linear regression is extensively used in applications to satisfy one of the following tasks:

- Prediction or forecasting: linear regression uses an observed dataset of dependent and independent variables and develops a predictive model. This predictive model can be used to make a prediction of the value of the dependent variable given only new values of the independents variables.

- Quantifying the strength of a relationship between a dependent variable and a number of independent variables; by doing so, we can assess which independent variable may have more or less effect on the dependent variable.

Formally, given a dataset with N examples and p independent variables as $dataset = \{x_{(i,1)}, ...x_{(i,p)}\}$ and $i = 1, 2, ..., N$, a linear regression model assumes that the relationship between the dependent variable y_i and the vector of x_i is linear. This is modelled through the error variable ε_i which adds noise to the linear relationship between the dependent variable y_i and independent variables x_i. Thus, the model can be presented in the form:

$$y_i = \beta_1 x_{i1} + ... + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, ..., N \tag{4.26}$$

or written in vector format as:

$$y = X\beta + \varepsilon \tag{4.27}$$

Further discussion about linear regression can be found in [111].

2. *Random Forest*: it is an ensemble learning method that operates by constructing a multitude of decision trees at the training stage. This regression is particularly useful for nonlinear multiple regressions. This approach combines the *bagging* idea and random selection of features. In a random forest, each node is split using the best among a subset of independents, also called predictors, randomly chosen at that node. First, let us describe *tree bagging*; if we have variable y_i responses to a set of features $X = x_i, ..., x_p$, bagging repeatedly selects a bootstrap sample n_j from the training set N

and fits regression trees to these sub sets. After training, prediction for new samples x' can be made by averaging the prediction from all x':

$$\hat{g} = \frac{1}{N}\Sigma_{n_j \in N}\hat{g}_{n_j}(x') \tag{4.28}$$

where $\hat{g}$ is the prediction model for the unseen sample x' using the average results from all the model sets $n_j \in N$. Unlike a single decision tree, random frosts use averaging to avoid common problems of high variance or high bias. This strategy performs very well compared to other algorithms in term of its robustness against overfitting. In addition, it has only two parameters to tune, the number of variables in the random subset at each node and the number of trees in the forest, which are usually not very sensitive to their settings. More details can be found in [112].

3. *Support Vector Regression (SVR)*: it is successfully employed in many applications (e.g., time series forecasting). Giving training data (x_i, y_i), SVR optimisation problem will be addressed as follows:

$$\begin{aligned} &minimise \quad \frac{1}{2}\omega^T\omega + C\Sigma_{i=1}^{N}(\xi_i + \xi_i^*) \\ &subject \quad to \begin{cases} y_i - \omega^T\phi(x_i) - b <= \varepsilon + \xi_i \\ \omega^T\phi(x_i) + b - y_i <= \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* >= 0 \end{cases} \end{aligned} \tag{4.29}$$

Where N is the number of samples. x_i is a vector of ith sample in the dataset and C is the constant that identifies the cost of error ε. The slack variables ξ_i and ξ_i^* deal with the infeasible constraints of the optimisation problem (i.e., upper training error and lower training error). For non-linear SVR, the kernel function ϕ maps the data into a higher dimensional feature space; further discussion of kernel functions is illustrated in [113]. Compared with linear regression, SVRs are nonlinear models and therefore they are able to represent more complicated behaviours. Compared to the Random Forest models, SVRs are smooth (differentiable) models and can predict behaviours variations smoothly and more precisely. Compared with other nonlinear models, such as neural networks, SVR models can be solved by quadratic programming, which provides global optimisation models in polynomial time. This is fundamentally different to obtaining neural network models by a back propagation optimisation algorithm, which can only obtain local optimisation models. These are the main motivations behind the selection of SVR in the prediction of ratings. Again, we used Weka [114] library for building and learning all the aforementioned models.

4.6 Experimental Setting

4.6.1 The Dataset

The properties of the dataset are summarised as follows:

- Since our framework incorporates friends' data into the recommendation foundation, we need a dataset that contains friends' posts and trust-related intercommunications information. In fact, there was no adequate dataset fits with the ISTS framework requirements since available datasets contained either ratings, reviews or a combination of ratings with reviews. For example the most popular datasets are Movielens which include users IDs, movies and ratings data. Ratings are structured and direct data from users to movies which is different from our approach requirements, we need to dig deep to collect data about intercommunication between users to represent trust such as re-tweet actions. In addition, we require users' friends short posts about items on OSNs.

- Due to the absence of the existing dataset that includes users intercommunication with friends' short posts, we built a software tool called Twitter Interaction Extractor (TIE) to search Twitter and prepare the dataset using the Twitter API for JAVA. TIE consists of two main parts (TIE-A and TIE-B) as demonstrated by algorithms 1 and 2 Page 100, which are implemented in JAVA to generate the dataset.

- In the first algorithm, we randomly choose 15 movies from the popular movielens[2] dataset; after that, we used the search query API to gather information about users who posted a tweet about each movie such as user's name and user's Twitter ID. We used the keywords *watching* and the *movie name* to retrieve the desired tweets about the mentioned movie; we also avoid celebrities and businesses who have a huge number of followers. Search keywords can be changed to fit with the target domain; for example, in book recommender systems, word

[2]http://grouplens.org/datasets/movielens/

Algorithm 1 TIE-A: Gathering Tweets Information

Input:Selected Movies items
Output: Relevant tweets and author ids
for every $movie_i$ **do**
  Search Query($movie_i$) in the public timeline
  if Relevant tweets content is found **then**
    Get id_i of content's author
    if id_i is not (celebrity or business) **then**
      $file \leftarrow content$
      $file \leftarrow id_i$
    end if
  end if
end for

Algorithm 2 TIE-B: Extracting Social Interactions

Input:Users ids of *tweets* publishers
Output: Re-tweets between every user id_i and every friend f_j
while id_i is not protected **do**
  $RT \leftarrow 0$ {RT: total number of re-tweet done by f_j}
  $rt \leftarrow 0$ {rt: number of re-tweets between f_j and id_i}
  Get FriendList of id_i
  $Fin \leftarrow$ Get followers number of id_i
  $Fout \leftarrow$ Get followings number of id_i
  while f_j in FriendList of id_i **do**
    if f_j is not protected **then**
      for every content in f_j's timeline **do**
        if content is available **then**
          if content is re-tweeted **then**
            $RT \leftarrow RT + 1$
          end if
          if content's author id $= id_i$ **then**
            $rt \leftarrow rt + 1$
          end if
        end if
      end for
      $file \leftarrow RT, rt, Fin, Fout$
    end if
    Get next f_j from FriendList
  end while
  Get next id_i
end while

such as *reading* will retrieve the desired tweets. For the second algorithm, after collecting the author/publisher information, we start to detect re-tweeting message activities and calculate the re-tweeting rate between this person and his friends. In addition, we obtain the numbers of followers and followings. It is important to highlight that when collecting this dataset, we start by capturing each tweet about a particular movie, then define the author of this tweet, *the trusted person*. After that, we analyse how friends communicated with this author (applying the trust metric we proposed). Conversely, in fact, the real proposed system starts searching given active users' accounts then tracks followings to find tweets about a movie. The reason behind starting the search in the opposite way when collecting the dataset is that it is difficult and time-consuming to allocate an arbitrary user on Twitter and then track his/her followings to find tweets about a certain item. Obtaining such subjective internal trust information between peers is not an easy task compared with objective trust computed over a community, such as reputation based systems [66]. In other words, reputation systems assign one fixed trust judgment to individuals, and therefore, all community members are considered to be affected by these values in their purchasing decisions.

- The collected dataset is called Social Data (SD1) has statistics that are summarised in Table 4.2. We collected these social data from Twitter in the period May-June 2013 as follows. Total number of users is 111 who are connected with a trusted relation with a friend (who tweeted about a movie selected from 15 movies described above). Example of the maximum trust values computed from SD1 is 1, and lowest trust is 0.05 using metric described in Section 4.3.2. Examples of sentiment rating *sr* computed based on the methodology in Section 4.4.2, came at the highest and smallest ratings as 5 and 1.7, respectively. SD1 is used to train and develop our models in this chapter, as well as in testing and evaluating the recommendation algorithms. An example of the dataset structure is shown in Table 4.3, every user such as user1 has trusted friends f in addition to the information of these friends such as their followings and followers to produce L values and the re-tweet RT rate between an user and the friend.

- In addition, to obtain real ratings about the gathered information we used three different annotators who we considered to be active Twitter users with no less than ten tweets per day. We needed a real user's decision associated with the collected social data. Figure 4.3 shows an example of a scenario based on real

Table 4.2: Statistics of the SD1 dataset

Users	Movies	Max trust	Min trust	Max sr	Min sr
111	15	1	0.05	5	1.7

Table 4.3: An Example of the dataset structure, where RT is computed based on Equation 4.6, and the inferred sentiment rating *sr* is computed based on method in Section 4.4.2

Users	Friend's tweets	sr	f's followers	f's followings	RT
user1	f's tweet: *Toy Story lol..*	4.2	127033	859	0.5000
user2	f's tweet: .. *Jumanji :D*	4.7	187	231	0.1000
user3	f's tweet: .. *high school musical was the best things to do today*	5	231	203	1
user4	f's tweet: *Father of the Bride!.. One of my favorite?*	5	435	251	0.3333
user5	f's tweet: *Watching Cinderella fun*	4	671	366	0.0588
user6	f's tweet: *Seven Pounds movie! :)*	3.5	668	527	0.3333

collected social data provided to annotators, in regression experiments. We have removed the movie titles from the scenarios provided to annotators because we want the annotators to focus on trust indicators and friends' opinions rather than being influenced by their personal opinions about certain movies and to avoid making movie annotation the goal of our task.

Consider the following scenario in the twitter space:

- Your have one friend x : you re-tweeted two of his/her tweets out of 14 re-tweets actions you have done at a particular time.

- Out of all his/her relations on twitter, this friend x has 99% followers and 1%following.

- This friend x posted a tweet about a (book, movie, mobile…etc.) and his/her opinion about it was rated as 3 star out of 5 stars.

How would you rate this item based on the previous scenario on a scale of [1-5]=[awful-excellent]; note that you can rate using fractions too; e.g. 3.5.

your answer is :

Figure 4.3: An example of a real scenario for an active user and a friend from Twitter provided to annotators

- The annotators have been asked to provide their opinions based on scalar rating from 1 to 5. For example, the three annotators gave ratings to the scenario 4.3 as

1.5, 3 and 1.5, then we chose the shared rating class between at least two of the annotators to be the ground truth, hence the chosen rating for this scenario is 1.5. In the case of no shared rating class, we consulted another annotator and because of this need we explored different ground truth methodology used in Chapter 5. The results of correlation coefficient show moderate positive relations as correlation between annotators 1 and 2 is 0.4, between annotators 2 and 3 is 0.4, and between annotators 1 and 3 is 0.5, while the average correlation results is 0.44. The results and evaluations beyond this correlation are logical in the domain of people's tastes and social preferences since such personal opinions are subject to change and strong positive relations may not reflect the real world situation. It is important to note that due to the continuous format of ratings that are provided by annotators we could not apply a measure such as kappa coefficient because it is based on the occurring of categories [115].

- Later in this thesis we collected a second dataset SD2 demonstrated in Chapter5. The proposed approach includes people tastes and their intercommunication which are subject to change over time. Therefore, in this thesis, to provide a suitable evaluation, two datasets are collected in two different time points rather than one big dataset.

4.6.2 Metrics

To compare the performance of the three regression algorithms used in this study, we applied different metrics. Generally speaking, *accuracy* metric is used to assess the prediction performance in test set. In our experiments, we randomly split the dataset into five non-overlapping folds. The experiments are repeated on the five folds. The reported results are based on the average of all trials on the five folds. Every fold is used as a test set using the model derived from the rest four folds. We applied the statistical accuracy metrics such as *Mean Absolute Error* (MAE), to evaluate the recommendation algorithm. This metric is the most accepted and widely used evaluation method in the recommendation community because it is easy to apply and comparisons can be interpreted directly. *MAE* is defined as:

$$MAE = \frac{\Sigma_{u,i \in N} |r_{u,i} - \hat{r}_{u,i}|}{N} \tag{4.30}$$

where N is the size the test set, and $r_{u,i}$ is the actual rating assigned by user u to movie i, and $\hat{r}_{u,i}$ indicates the rating estimated by the proposed recommendation algorithm. In more detail, $r_{u,i}$ is obtained when the ground truth value was collected based on scenarios provided to the annotators who evaluate our system. In classical recommenders, they hide some users' ratings about some items and then the proposed model is applied to predict ratings. After that, a comparison between actual and predicted ratings is carried out. However, in the case of new users, these systems do not have a history of ratings, and therefore, the ground truth is needed to make the required comparison. Obtaining smaller values of *MAE* demonstrate more accurate performance of the system.

In addition, another metric that is used the *Mean Absolute Percentage Error* (MAPE). It is a measure of accuracy in statistics and expresses accuracy in the form of a percentage. It is defined by the formula:

$$MAPE = \frac{100}{N} \quad \Sigma_{u \in N} |\frac{r_{u,i} - \hat{r}_{r,i}}{r_{u,i}}| \tag{4.31}$$

This metric is used to provide a summary of the accuracy of predictions and it helps to compare the performance of different algorithms. It is a representative measure of the overall prediction quality expressed in an easy and understandable way. We argue that rating prediction can have relative semantics between the main rating classes; hence, MAPE is an appropriate accuracy evaluation when it comes to variables (ratings) that depend on the proportional size of errors relative to the actual data.

To compare the performance of the three applied machine learning classification algorithms we applied different metrics. We used an *Accuracy* metric to indicate the percentage of the correctly classified instances in the test set. However, comparison using this metric is not sufficient because it is not sensitive to class distribution or the chance of being correct. Hence, we also used standards evaluation measurements that are widely used in information retrieval and classification such as *Precision* and *Recall* defined in Equations 4.32 and 4.33 respectively. These two metrics test the accuracy of the classification algorithm in predicting ratings [108].

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.32}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.33}$$

The F-measure is considered as the harmonic mean between the two metrics precision and recall to overcome any conflict between them. It is given as follow:

$$F - measure = 2.\frac{Precision.Recall}{Precision + Recall} \tag{4.34}$$

To illustrate the quality of our approach we compared the results with several baselines:

- **B1**: Item Average Score baseline(IAS), this method assigns for any unknown missing values the average of the corresponding item ratings.
- **B2**: Most Dominated Score baseline (MDS), this method assigns the majority of the rating score in the dataset to any missing values, which is equal to a score of 5 in our dataset SD.
- **B3**: Trust-based weighted mean approach proposed in [58]. This method does not require similarity utility and employs trust values.
- **B4**: Trust-based collaborative filtering approach proposed by Massa and Avesani [59], this method uses trust values instead of similarity values in the traditional collaborative algorithm.
- **B5**: This method is based on the probabilistic theory to infer multi-point scale of ratings from standard reviews presented in [16].
- **B6**: This approach is used in the literature and replaces an unknown rating by a random number generator function. In particular for new users, this approach is very simple computationally. It generates a random number in the range of ratings used within the system. However, if we have a large number of missing values, then it may cause inaccurate and less personalised recommendations .

For baseline methods B3 and B4, the path of trust is limited to one since the shortest propagated trust path yields better results, as illustrated by the authors of [58]. In fact, most literature studies assume the existence of user's ratings and friends' information; however, in our approach, we deal with the more challenging case where user rating profiles do not exist. We denote the method proposed in Section 4.5.3 as $ISTS_{regression}$.

In the classification experiments, all results from all methods rounded to the nearest class to allow unbiased comparisons since results from classification models and actual user ratings all come in discrete class values in the next evaluation section. For example if the output rating is 3.75 or 3.5, it will be rounded up to 4, whereas outputs of 2.2 and 2.33 will be assigned within the class rating 2. We denote the method proposed in section 4.5.1 as $ISTS_{heuristic}$ and the method proposed in Section 4.5.2 as $ISTS_{classification}$ (based on the best accuracy gained by the decision tree model).

4.7 Evaluation of Results and Discussion

4.7.1 Classification Results

Experiments were conducted to evaluate the accuracy of the recommendation classification algorithms given the computed trust and sentiment values derived from the collected data described in Section 4.6.1. Experiments were implemented using the trust in Section 4.3.1 and bag of words method described in Section 4.4.1.

Table 4.4: Evaluations of the Classification algorithms

Algorithms	Accuracy	Precision	Recall	F-Measure
Naive Bayes	57.65 %	0.58	0.58	0.57
Logistic Regression	67.60%	0.66	0.68	0.67
Decision Trees	72.10 %	0.73	0.72	0.72

Table 4.4 shows a summary of the results. We can see a comparison of the performance of the three algorithms. By looking at the F-measure values of the three algorithms (NB, logistic regression and decision trees), we can see that the decision tree has prediction of a higher power than the two others algorithms. It is also clear that the NB classifier has the worst accuracy percentage (58%) compared to the best performance achieved by decision trees algorithm (72%). More specifically, Figure 4.4 sheds more light on these metric results using the decision trees according to each class. The algorithm seems to perform better with negative classes, *Dislike* and *ExtermeDislike*, than positive classes. However, the *Neutral* class reflected the lowest accuracy, which may be due to the ambiguous nature of this class since it holds uncertainty about having a particular type of opinion. It is difficult to determine whether the *Neutral* class is closer to which polarity (negative or positive) unless further contextual information is included, such as demographic information, or more trust indicators are used.

We observed that trust is the feature used as criteria to split tree branches. The trust value which is used as the first threshold to divide the tree is 0.17 from resulted tree model using Weka. If the trust is below this threshold, the algorithm does not estimate

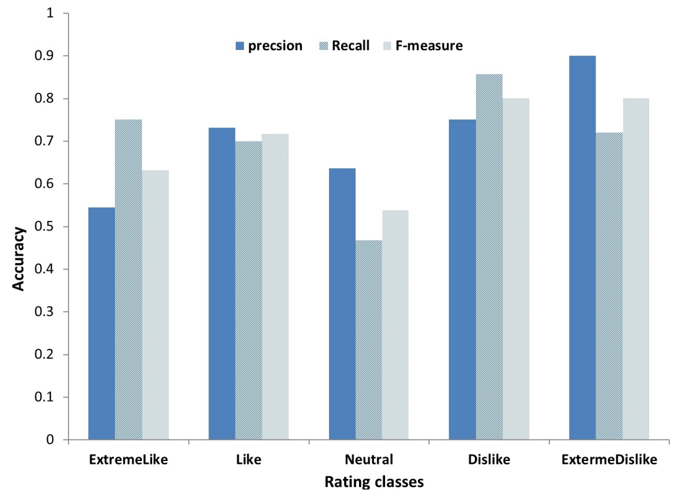


Figure 4.4: Precision, Recall and F-measure for each ratings class produced by the decision trees algorithm

any satisfaction classes such as *ExtremeLike* and *Like* or even *Neutral* class. In addition, the confusion matrix indicated most of the incorrect estimations were allocated to classes, which are closest to one another. For example, the decision tree algorithm allocates wrong ratings estimations associated with the *Like* class to neighbouring classes, *Neutral* or *ExtremeLike* which are semantically relevant rather than estimating further distance classes, such as *Dislike* or *ExtermeDislike*.

4.7.1.1 Recommendation Results with Different List Size

We evaluated the performance of our approach in relation to group of friends. In this experiment, we assumed the existence of 12 totally new users. Each user is assigned a group of friends of size 10 friends sampled from the dataset SD. Moreover, we applied a threshold of the lowest acceptable trust value of 0.17 based on the *information gain* in decision trees model. This threshold will be applied for all baselines. Only friends with trust values exceeding this threshold were involved in the recommendation process. After selecting trusted friends, this local community will be treated as a user's neighbourhood. Table 4.5 reports different values of the average of MAE from the groups of friends based on different algorithms. B4 achieved the highest error among all the methods. This is because it is based on the traditional collaborative filtering method which requires both common ratings between friends or items, and user average ratings. Therefore, since traditional collaborative filtering usually leads to poor

Table 4.5: MAE values using a size of friends' group of ten

Algorithms	MAE
B1	1.292
B2	1.735
B3	1.278
B4	3.236
B6	1.720
$ISTS_{heuristic}$	1.200
$ISTS_{classfication}$	**0.836**

recommendations for new user situations we excluded this algorithm from next experiment. It can be observed that both of our methods $ISTS_{heuristic}$ and $ISTS_{classification}$ returned the smallest error results of 1.2 and 0.836, respectively, indicating their superior performance.

We excluded B4 and B6 from next experiments for the following reasons:

- Due to the large error that obtained by B4 indicates low performance. This high error occurs because B4 includes trust based on collaborative filtering formula [59] which includes users average ratings and that explains why it fails to provide recommendation for new users with no history of ratings is available. However, B4 is evaluated again when testing regressions models because it includes trust component.

- There is no certain technique behind B6, it is based on choosing random ratings to fill the missing item ratings, hence error changes every time when it is applied. Therefore, B6 is excluded from the rest of experiments.

It is important to explain that B5 was not applied in the classification experiments. This is because B5 is based on the probabilistic sentiment analysis that is similar to the method used in regression approach, hence B5 is used later in the evaluation of regression algorithms.

4.7.1.2 Evaluation of Performance with a Sparse Group of Friends

In this experiment, we considered the smallest group of friends (e,g. containing only one friend) to show how these methods would perform in terms of accuracy on very sparse groups of friends.

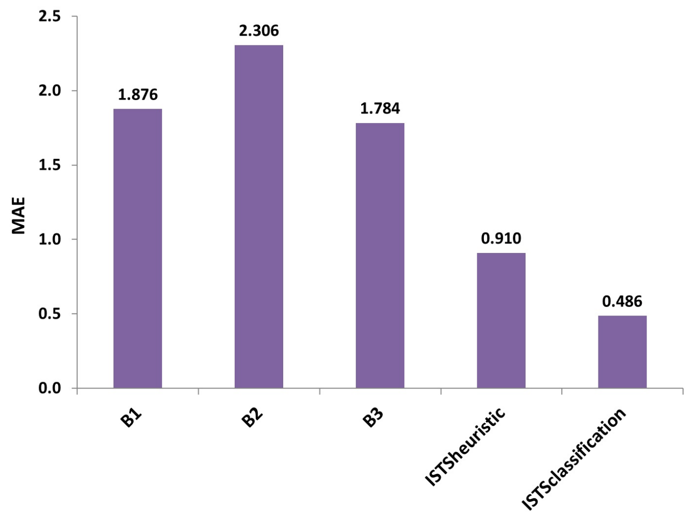


Figure 4.5: Performance Comparison using MAE

The results from this experiment were consistent with the previous one. From the results shown in Figure 4.5, we notice that our proposed methods outperformed and generated more accurate results than the different baselines of trust-based recommenders. Again, our second proposed approach $ISTS_{classification}$ outperformed $ISTS_{heuristic}$ returning the smallest MAE value, 0.486, which is consistent with the results of the previous experiment . An explanation of this trend is that $ISTS_{classification}$ is based on learning models which always achieve higher accuracy; whereas, $ISTS_{heuristic}$ is a heuristic method which can capture changes in tastes and in trust behaviour more easily than learning models despite being of lower accuracy. Broadly speaking, methods that utilise trust performed better than other baselines methods, which is consistent with findings previously reported in [2].

4.7.2 Regressions Algorithms Results

This experiment is used to evaluate the quality of the recommendation regression algorithms based on the dataset SD as described in Section 4.6.1. The experiment was implemented using the trust explained in Section 4.3.2 and probabilistic sentiment method described in Section 4.4.2.

The results are summarised in Table 4.6. We compared the performance of the three algorithms (random forest, linear regression and SVR) by looking at accuracy based on MAE and MAPE values. SVR produced the most accurate predictions compared to the other two algorithms, closely followed by linear regression. It is clear that the Random Forest algorithm has the worst accuracy results, represented by MAPE percentage of over 17%, while relatively better performance was shown by SVR which produces a percentage error of 16% of all test ratings. In the next sections, the rest of experiments are curried out based on the best performing algorithm the SVR model.

Table 4.6: Evaluations of Regression algorithms

Algorithms	MAE	MAPE
Random Forest	0.52	17.12 %
Linear Regression	0.47	16.56%
SVR	0.45	16.00 %

4.7.2.1 Recommendation Results with Different List Sizes

In this experiment, we need to simulate whether a recommender system has the ability to recommend a set of items for a given new user based on group of friends. We assumed the existence of 12 totally new users, with no history of ratings, and each assigned a group of 10 friends sampled from the dataset SD. After allocating the friends list of every user, we treated this local community to be the user's neighbourhood.

Table 4.7 shows different MAE results for a group size of 10 friends returned by different models. It can be observed that our proposed regression method $ISTS_{regression}$, based on the SVR algorithm, achieved the smallest error 0.349 in recommendations.

Table 4.7: MAE values using a list of 10 friends

Algorithms	MAE
B1	0.586
B2	0.791
B3	0.607
B4	1.628
B5	0.619
$ISTS_{regression}$	**0.349**

The baseline method, B4, returned the highest error among all the methods 1.628, as it is based on a traditional collaborative filtering algorithm, which is clearly affected by the absence of user's ratings history. Moreover, we tested the quality of performance using friends' list sizes of 3, 7 and 11 as shown in Figure 4.6.

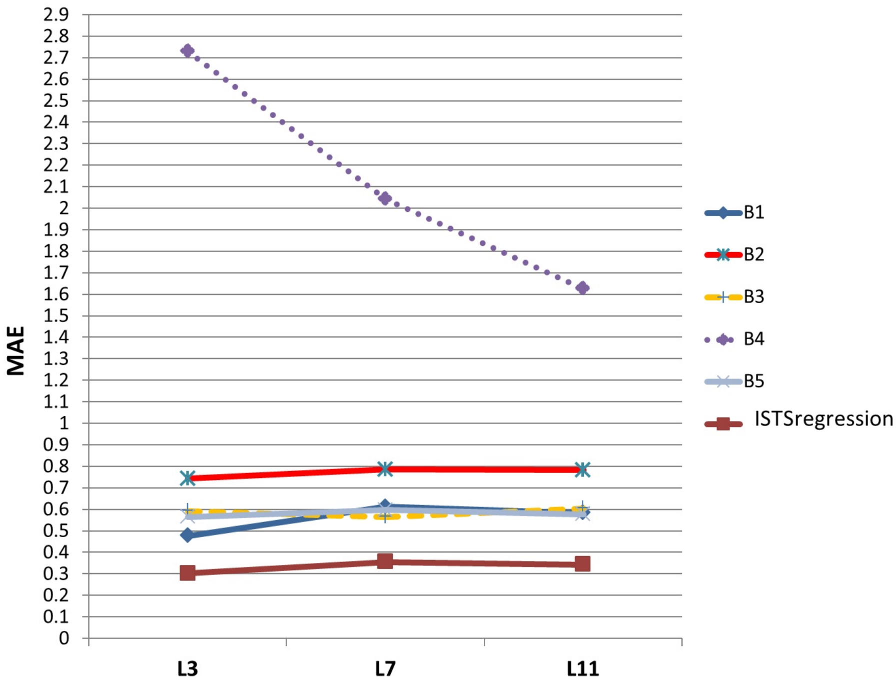


Figure 4.6: MAE values based on different sizes of friends' lists

$ISTS_{regression}$ showed consistent performance over all tested lists sizes with the lowest MAE values compared to the other methods. In addition, B1, B2 and B3 showed better overall performance than B4, which returned a high MAE with a list size of 3, and its performance improved as the list size increased. This high error occurs because B4 based on collaborative filtering formula [59] which includes users average ratings and that explains why it fails to provide recommendation for new users with no history of ratings is available. For this reasons B4 is excluded from next experiment.

4.7.2.2 Evaluation of Performance with Sparse Group of Friends

In this section, the experiment used the entire dataset SD. We also considered the smallest groups of friends contained only one friend to show how these methods perform in terms of accuracy in a very sparse friends' data as it is tested with classification algorithm before.

Again, the results from this experiment were consistent with the previous one. From the results shown in Figure 4.7, it is clear that our method outperformed the other methods and could generate more accurate results than baselines of trust-based recommenders. Another observation related to the impact of trust is that methods that adopted trust, such as *B3*, performed well which is consistent with published literature [2]. In addition, trust and sentiment inferred from OSNs' language features increased the effectiveness and the quality of prediction. This can be seen when $ISTS_{regression}$ is compared with *B5* which is based only on a probabilistic sentiment inferred from standard reviews.

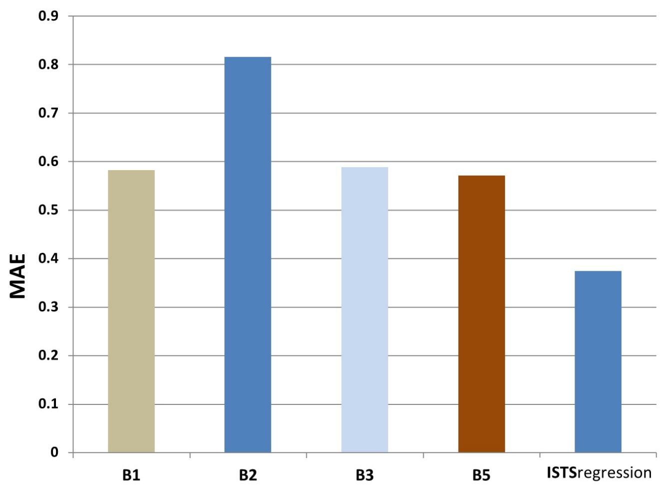


Figure 4.7: Performance Comparison in a very sparse friends' information

4.7.3 Regressions vs. Classification Modelling for New Users

$ISTS_{classification}$ can be used to solve problems associated with new users problem [91] by employing the bag of words methodology and the number of negative and positive words when measuring sentiment in tweets at the words level. In $ISTS_{classification}$, trust was computed only based on the re-tweet action and the rating prediction was produced using classification algorithms. In order to seek more accurate approaches to implement ISTS, the followings points can be discussed. On the one hand, $ISTS_{regression}$ [116], showed better results in terms of accuracy. For example, the MAE values obtained using $ISTS_{classification}$ based on decision trees in cases of a very spare group of friends and those of ten friends were 0.486 and 0.836, respectively, from Figure 4.5 and Table 4.5. While $ISTS_{regression}$ based particularly on regressions in particular SVR returned MAE values of 0.374 and 0.349 for those cases, respectively, from Table 4.7

and Figure 4.7. On the other hand, $ISTS_{regression}$ can capture the continuous relation between rating classes, for example predicted ratings of 4, 4.5, 4.75 and 5 are all under satisfactory category. This is the opposite of classification algorithms that manipulate any predicted value as a wrong answer if it is not retrieved as exactly as the supervised labels. Different from bag of words techniques, $ISTS_{regression}$ implemented probabilistic sentiment analysis which is the method used to calculate sentiment ratings *SR* from micro-reviews based on the weight of the sentiment word. In addition, it integrated *RT* and *L* into trust foundation. Based on these observations, $ISTS_{regression}$ performed better than $ISTS_{classification}$ in terms of accuracy of predictions.

4.7.4 Performance Evaluation Using Statistical Hypothesis Test

In this section, we are interested to study the significance of the results of research approach $ISTS_{regression}$ that shows the superior performance by applying a statistical hypothesis test. We want to test if the proposed algorithm produces error smaller than other baselines using SD1.

1. We investigate two types of hypothesis [117]. Null hypothesis (H0) is a simple hypothesis associated with a contradiction to a theory one would like to prove: ISTS does not provide improvements in reducing error compared to other baselines. Alternative hypothesis (H1) is a hypothesis that is associated with a theory that we would like to prove: ISTS improves the accuracy level by reducing error against the error produced by other baselines. This hypothesis will be used to test the significance of a research algorithm based on the resulted errors [118]. We test the significance of research approach $ISTS_{regression}$ against the baselines based on the dataset SD1.

2. Defining the alpha that shows the significance level which is commonly used as 0.05. This Alpha is used to be compared later with the P-value resulted from the statistical test. At this case, the confidence level is 95% which refers to the percentage of all samples that can be expected to provide lower error by ISTS.

3. Computing the P-value which is used to determine whether the results of the experiments are within the normal range of values for the error being observed. In other words, the minimum probability of error using test statistics to interpret the P-value from random sample [117]. For this testing, we used the t-test for performing a paired t-test [118] between two samples of error, the error in $ISTS_{regression}$ and baselines.

4. Comparing the resulted P-values with the significance level 0.05 to deicide the possibility of rejection the null hypothesis.

The results obtained using paired t-test which is implemented in MATLAB using the function *ttest* as follows: (1) Based on experiment in Section 4.7.2.1, the P-values with different baselines are B1: 0.00042805, B2: 0.0012, B3: 0.0058, B4: 0.000032899 and B5:0.0033. (2) Based on experiment in Section 4.7.2.2: hypothesis test against the baselines B1, B2, B3, B5, presents P-values: 0.0100, 0.0017147, 0.0167, 0.0032464 respectively. We can observe that the P-values are smaller than the significance level 0.05 and therefore we can reject the null hypothesis and the alternative hypothesises are accepted statistically that $ISTS_{regression}$ produces error smaller than other baselines.

4.8 Areas of Improvements

Although the ISTS method, discussed in this chapter, has supported important features, it has introduced some limitations, indicating room for improvement. These improvements include:

1. The trust indication that we have assumed in this chapter include only two factors: re-tweets RT and list of followers/followings L. We considered that both factors have the same level of effect on building $trust_{u,f}$ values. Therefore, one suggested improvement is to study the real power of each factor. In other words, we should know whether RT factor participates in trust determination more or at the same level as L. Consequently, different trust foundation will effect the prediction of items' ratings.

2. To verify Point 1, we need to implement a method to optimise contributions from RT and L.

3. Although, re-tweets RT and lists of followers/followings L are popular properties, we need to investigate the effect of other OSNs properties that may contribute to develop better measures of $trust_{u,f}$. Furthermore, we need to investigate the importance of every new factor, and hence, more reliable recommendations can be made.

4. To satisfy Point 3, we need to collect another dataset that includes the new trust factors and apply suitable data annotation techniques.

4.9 Chapter Summary

This chapter has started by illustrating the approach requirements. It has then described the methods used to build every component in the novel approach which are trust, sentiment and rating prediction. First, trust was modelled by considering the retweet (RT) actions between a target user u and friends or using a combination of RT and the list of followings/followers L. This chapter has demonstrated how to extract friends' opinions into rating scale; two sentiment analysis techniques were developed: a bag of words and probabilistic method. Methods used to predict ratings for new users were developed, including: a heuristic technique, classifications and regression models. The experiments conducted in this chapter demonstrated under different list of size. Although our proposed ISTS method has provided significantly accurate results, some areas of improvements still exist. This has led to design of an improved approach, which is an enhanced version of the ISTS method to overcome the identified limitations. The enhanced ISTS framework is discussed in the next chapter.

Chapter 5

Enhanced Approach: H-ISTS using Genetic algorithm for Trust Optimisation

5.1 Introduction

This chapter proposes an enhanced version of the ISTS method from Chapter 4 to support the optimisation of trust features using a genetic algorithm. We need to investigate the effect of every trust feature when measuring the total trust metric by optimising the parameters of every trust feature. Constantly, the new trust metrics are used to build personalised recommendations. Furthermore, in this chapter, a more fine-grained trust features are discussed and integrated in the new approach. A number of trust features on online social networks are explored and evaluated to indicate their influence in relation to trust. This approach is called **H**ybrid **I**mplicit **S**ocial **T**rust and **S**entiment (H-ISTS) based recommender framework and should overcome the limitations of the ISTS described in Section 4.8. H-ISTS has been published in [119].

This chapter is organised as follows. Section 5.2 gives an overview of the preliminary requirements of the H-ISTS. Section 5.3 introduces the needed background for genetic algorithms. Section 5.4 introduces the main ideas behind the H-ISTS. Section 5.5 describes the settings of the experiments carried out. Section 5.6 shows the analysis of the performance of H-ISTS. Section 5.7 points out some important issues should be further addressed. Section 5.8 summaries the chapter.

5.2 Additional Requirements of H-ISTS

Before proceeding to the design details of the H-ISTS approach, it is necessary to report the general requirements needed to be implemented in the H-ISTS approach. It is designed to fulfill the desired features described in Chapter 3. H-ISTS improves the ISTS and satisfies the requirements specified in Chapter 4. However, the use of followings is the additional requirement specified for the design of the H-ISTS.

1. **Optimising trust parameters**: H-ISTS should adapt a method to optimise trust metric. A Genetic Algorithm (GA) is used to tune the trust parameters and generate the right weights that describe the right power effect of each trust feature.

2. **Improving trust features**: This requirement should include additional online social trust features which may strengthen building trust. Consequently, more behaviours and activities between users and their friends on OSNs should be used to indicate their impact on the amount of trust.

3. **Collecting a new dataset**: This requirement necessitates crawling twitter for suitable dataset that satisfies the new requirements in Point 2.

Importantly, it is necessary to mention that the main ideas in this chapter are designed to improve trust representation in ISTS. Therefore, sentiment analysis that is adapted from here on is based on the probabilistic sentiment techniques that was used in Chapter 4.4.2 and $ISTS_{regression}$ approach described in 4.5.3 based on support vector regression.

5.3 Background: The Genetic Algorithm

Genetic algorithms is first introduced in 1975 by Holland based on Darwin's theory of evolution, [120]. In the field of artificial intelligence, GAs are used to simulate the natural evolutionary process to search for optimised solutions. GAs start search heuristically for solutions taken from random pool of a population and can be applied by implementing two important elements:

1. **GA operators**: A number of operations are used to run GAs:

   Selection: In every production of a generation, only a proportion of the current population is selected to generate the next, new generation. Only individuals that satisfy the problem objective function are used.

Crossover: In this operation, the GA chooses genes from the parents' chromosomes and creates a new generation by swapping the selected genes.

Mutation: After crossover generates the new combined offspring, mutation take place. This is to avoid making all solutions in a population into a local optimum of the proposed problem. Mutation randomly changes the position of the inherited genes in the offspring's chromosomes.

2. **Fitness Function**: The goal of defining a fitness function is to evaluate whether the candidate solution, after performing the above operators, is considered as a potential good solution.

Many attempts have been made to apply GAs in RSs in different ways: clustering, hybrid user models and optimising similarity functions. In the clustering technique for example, GA-based K-means model as used in [121], users are clustered into groups that have similar tastes, then the collaborative filtering technique is applied over clusters, and the results are obtained with the only privilege of reducing the calculation time. Hybrid user models, such as that in [122], are based on combining collaborative filtering and content-based techniques to harness the advantages of each method. In these cases, GA can be designed to hold information, such as demographic characteristics by implementing demographic features into the chromosomal structure. In addition, in [34], GA is used to obtain the optimal similarity function instead of a correlation coefficient similarity metric based only on ratings information. In a recent research [123], it was illustrated that recommender systems based on context data over the web using computational intelligence algorithms, such as genetic algorithms, can show effective performance and substantially mitigate the cold-start and sparsity challenges. The three major operators used in genetic algorithms are selection, crossover, and mutation to find the offspring of the existing population. In our approach, a GA is used to optimise the extracted social trust features as follows: a tournament selection method is used to allocate the candidate chromosomes for the next generation. surviving chromosomes into the next generation go through the crossover step. The selected chromosomes are assigned to random positions as newly crossed chromosomes to generate the new population. The mutation operator starts after the crossover step to decide the potential of whether a chromosome can mutate in the next generation. In our approach, we use uniform mutation because of its property of fixed probability of changing gene structure in generations and the recurrent property of returning reversibly when any eventual convergence occurs.

A GA can be implemented effectively in the new approach as an optimisation method for three reasons: (1) RS problems have criteria to evaluate good solutions among the generated offspring; for example, the evaluation of error degree in resulted ratings. (2) GAs can help to find the best possible solution to improve the accuracy of the RSs and assist in finding personalised recommendation that are close to user's preferences. (3) Trust which is our target parameter for optimisation, in H-ISTS, can be divided into two elements; i.e, RT and L independently, to represent genes in GA. Using GA to identify what contributions of RT and L are to the trust component to achieve the most accurate prediction of the rating.

5.4 The H-ISTS Approach

As explained earlier, the H-ISTS framework is a novel method that is developed based on the ISTS and is designed to overcome its limitations. The next section is the trust optimisation method based on the GA.

5.4.1 Trust Optimisation

The fundamental objective of the H-ISTS method is to tune the trust metric in order to improve the prediction accuracy level. Consequently, we need to answer the question of which trust features can produce the largest effect on the final rating estimation which approximates new user's opinion.

Initially, we assume that new users in the H-ISTS framework have provided their Twitter accounts details to the system. Then, H-ISTS starts by analysing their followings' accounts. From Figure 5.1, it can be seen that the H-ISTS framework adopts three main components: first, a probabilistic sentiment analysis technique that is applied to extract the degree strength of opinions from micro-reviews as described in Chapter 4.4.2. Second, optimising social trust features, such as followings/followers and re-tweet actions using a genetic algorithm. Third, using both the optimised trust parameters and friends' opinion degree to predict ratings for new users using the support vector regression model (SVR). The overall process of our proposed approach is illustrated in Figure 5.1. The process of optimisation is repeated and fitness function is updated by new parameters of trust in every iteration.

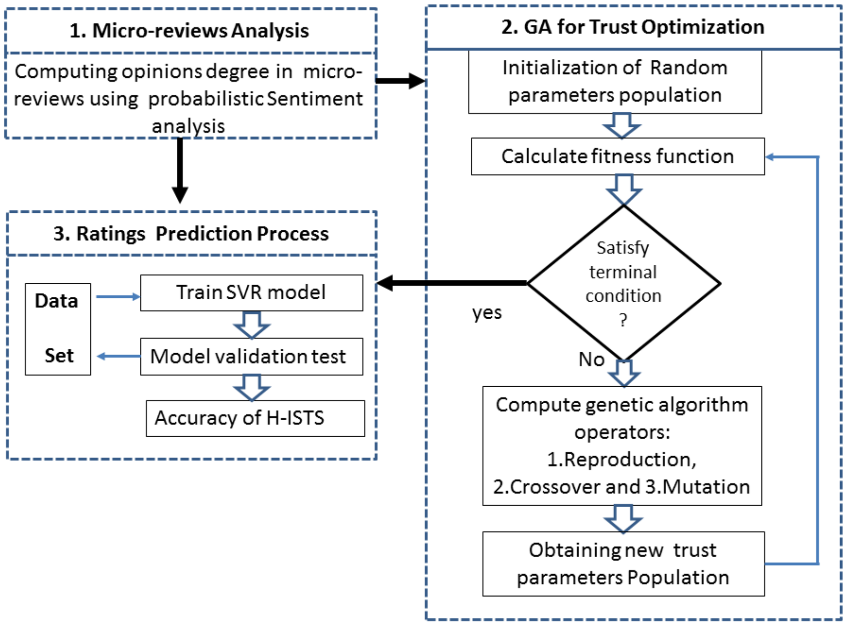


Figure 5.1: The Optimisation process of the H-ISTS framework

In the H-ISTS, the trust parameter values are dynamically optimised using an evolutionary process. Trust parameters are randomly defined to generate an initial population of chromosomes. All the values of the parameters are implemented into chromosomes with real number values. In the chromosome selection process, our proposed model implements a tournament method. Crossover and mutation operators are used to modify the chromosomes and ensure that only the best chromosomes in each generation survive and move to the next generation. The fitness function evaluates the candidate offspring. These processes are continuous and in every iteration, the termination criteria are tested to stop running the algorithm.

The H-ISTS model is coded using MATLAB. The SVR engine that is used to train and validate the model is LIBSVM3.19, as this tool is used widely and successfully in different studies. All source files and the required code are programmed in MATLAB.

More specifically, to enhance the trust representation, the H-ISTS should allocate the right parameters for the variables re-tweet: RT, followings/followers: L. Intuitively, trust is identified as a linear relation between the two features RT and L as follows:

$$trust_{u,f} = w * RT + (1-w) * L \tag{5.1}$$

where we denote trust between user u and friend f as $trust_{u,f}$, and w is the trust parameter that indicates the importance of variables RT and L . Obtaining the most suitable trust weight parameters w for a social trust recommender system depends on the nature of the social data. H-ISTS tries to allocate, among all the possible w values, the one that represents the trust function and reduces Mean Absolute Error (MAE) to the minimum. In order to do this optimisation goal, we use a Genetic Algorithm. We applies a supervised learning task where the fitness function is the accuracy metric of RSs. The population of our GA represents the set of different values of w. In this way, GA will determine the optimal value of w when it succeeds to improve the applied accuracy metric. This iterative algorithm terminates when a maximum number of generations has been produced.

From Figure 5.1, we need to design a fitness function which can assess the performance of each individual solution. MAE is used as a measurement to qualify every encounter of fitness function (we used only 80% of users as a training set from the dataset).

The smaller MAE value, the more accurate the performance of the model is. From algorithm 3, the main steps in the fitness function can be seen.

Algorithm 3 Fitness Function Algorithm

```
Input: trust values vectors: L (List of follwoings/followers) and RT (re-tweet actions)
Output: MAE
Y=0
for every u in N do
    Apply the optimized parameters from current generation to get trust value
    trust_u = w * RT + (1 - w) * L
end for
Train the data with new trust using SVR model
Predict rating using the trained model
Y = + MAE_u for every user in N
RETURN Y/N the average of MAE
```

The fitness function requires the trust dataset to include the L and RT information for every user. Then, for every user, trust is computed based on the current generation of parameters w. After new values of trust are prepared, ratings are estimated by building a trained SVR model. At this point, MAE is tested for all the estimated ratings.

The fitness function is executed again staring from line 3, after a new generation is produced and the termination criterion is valid.

We believe that our proposed H-ISTS approach is able to optimise several trust features from the OSNs and can be integrated with RS for developing a Hybrid Implicit Social Trust and Sentiment based approach for RSs. Next section discusses the integration of more trust features.

5.4.2 The New Trust Features

The main two features in trust that we explored are *RT* and *L* and that was for two reasons:

- Based on a number of studies [96, 99, 100, 106], *RT* and *L* are proposed to indicate hidden trust and the influence of users and tweets.

- *RT* and *L* are structured data and can be captured and included in experiments with less noise that is usually obtained from text information that need pre-processing steps. Capturing personal re-tweets between individuals is not an easy task, however re-tweet actions are structured data that attract researchers to utilise them directly in experiments and evaluations.

However, OSNs have other properties that allow people to communicate. In order to indicate which of these properties the H-ISTS will support, we conducted a survey that investigated the Twitter properties and user activities that may represent trust in an effective way.

The most Twitter properties that can be found in Twitter environment are:

- Re-tweet activities (RT): as described earlier, it is the action when one user re-send another friend's post. For this parameter, we asked participants about the effect of the frequency of re-tweets they carry out in relation to a friend's tweets.

- Mention activities (Mens): this is the activity of a user targeting another user by indicating the other user's name in their posted tweet.

- Followings: the number of twitter accounts that the user follows.

- Followers: the number of twitter accounts that follow the user.

- Tweets: the total number of tweets that have been posted by a certain account.

- Favourites (Fav): this indicates the activity when one user likes another user's posted tweets and saved it in a favourite list.
- Location (LOC): this is the location, which specifies the country and the city of the account holder.
- Social web pages (Web): this indicates the availability of web pages that a user links his/her twitter account to, such a Facebook, Instagram.
- Font colour (Fontcol): the font color used to edit the text.
- Name: this represents the name appears on the account front screen.
- Bio: this is brief information about the account holder.
- Image: the picture that represents the user and is located next to the user-name.
- Date: this is the date that indicates when the user joined twitter with a particular account.

As these features may contribute to increase trust; however, our goal is to integrate only the five most important features. We did not implement all features to maintain simplicity of the model and make it more applicable, less complex, easier to generalise to different environment. This raises questions about the influence of these features. To assess this influence, we asked 50 Twitter users about the above features using the online tool *www.surveymonkey.co.uk*. More than half of the participants were 22 to 37 aged people who use Twitter several times per day and about 70% of participants are female. Participants were asked to answer specific questions in order to indicate the importance of the different features they considered when assessing trust in friends. Based on all the previously mentioned features, two main tasks were included in the questionnaire:

(A) The participants were asked the question: Which kind of features in friends accounts may influence your trust and belief in their tweets which contain opinions about a product, such as movies, books, games, restaurants, etc.? In this task, the participants could evaluate every feature by choosing the importance on an scale of five points [Least important- Extremely important].

(B) They were asked to rank these features in a list from the most important to the least important feature they thought would determine their level of trust.

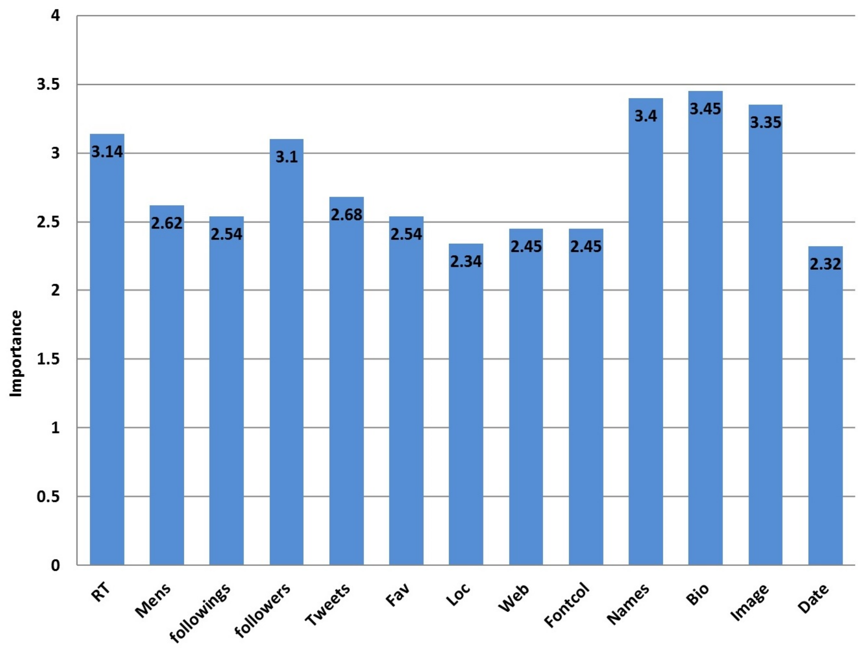


Figure 5.2: Findings of Part (A) of the online survey: Features importance

The survey results can be shown based on the two tasks above. The finding from the first part **(A)** are shown in Figure 5.2. The graph shows that some features were considered to be of the highest importance by the surveyed users based on a numerical weighted average: (RT, followers, Name, Bio, Image). In term of the relative importance scale, these features scored averaged of above 3 out of 5 on the custom scale we provided: 3.14, 3.10, 3.40, 3.45 and 3.35 respectively. From these findings, the most important feature that highly influenced trust in friends' accounts may be Biography information according the participants. By contrast, the feature that was the least influential according to the participants was the Date feature, which had an average score of 2.32.

Table 5.1 shows the findings of the second part **(B)** of the questionnaire. Because we intend to select only five features, we look only at the top five ranked positions in list of 13 features.

Table 5.1: Findings of Part (B) of the survey: Rank results

Features	Percentage %	Rank
Name, Bio	24.39	1
Followers , Name	19.51	2
Image, Tweets, Followers	14.63	3
Mens	20	4
RT	14.63	5

For the first position, 24.39% of participants chose the Name and the Bio features as the first rank in the list. Name and Followers were second place according to 19.51% of the participants. In the third position, the highest score went to Tweets, Image and Followers. The mention feature, Men, allocated to be in the fourth place by 20%. Finally, RT came in the fifth position to 14.63% of participants.

5.4.2.1 Importance of RT and L and other Features

We explained in the beginning of Section 5.4.2 the reasons of using *RT* and *L* in our approach. However, the results from the conducted survey have also indicated the importance of these two features *RT* and *L*. By looking at Figure 5.2, *RT* and *L* are among the highest important features above level 3. From Table 5.1, *RT* and *L* also obtained position in the top five positions when features were ranked by participants.

Features that were highly emphasised in both parts of the survey were selected to be included in the H-ISTS framework. Therefore, from the discussions in Section 5.4.2 we arrived to the following remarks:

- The participants indicated the importance of every feature, and the features that were determined to be of the highest importance were: RT, Name, Bio, Followers and Image. Date and location are given the smallest important average around 2.3.

- The participants answers in part **B** ranked features as follows:

  1. Name and Bio.
  2. Followers and Name.
  3. Image, tweets and followers.

4. Mens.
5. RT.

- However, we eliminated *Tweets* feature from the third place because of some tools became available that can be used to send automatic tweets [124]. Similarly, we did not include the Mens feature which came in fourth place, due to the nature of mentions, which implies direct posts to a particular user. In fact, they may be considered as personal or private conversations between two users, and hence, they include high level of noise and distraction from the main purpose of personalised item recommendations.

From the aforementioned remarks we included these features to build our new trust metric: RT, Followers, Name, Bio and Image. Next section describes the implementation of theses features.

5.4.3 Optimisation of the New Trust Features

In this section, we describe how every feature will affect the trust between a target user and a friend. More specifically, we can consider vector $T_{(u,f)}$ whose dimension is the number M of the possible trust features between user u towards friend f:

$$T_{(u,f)} = (t_{(u,f)_0}, \ldots, t_{(u,f)_M}) \tag{5.2}$$

As we have five trust features in our case,
$t_{(u,f)_0} = Re - tweetrate, t_{(u,f)_1} = Bio, t_{(u,f)_2} = Image, t_{(u,f)_3} = Name, t_{(u,f)_4} = List$. And we consider another vector **W** showing the importance of each trust feature in determining in T:

$$W = (w_0, w_1, \ldots, w_M) \tag{5.3}$$

We consider the following trust function $trust_{(u,f)}$, for each component in the vector W (i.e, w_i, represents the importance of the corresponding component in the trust vector T, t_i), which is determined as follows:

$$trust_{(u,f)} = \frac{1}{M} \Sigma_{i \in M} w_i t_i \tag{5.4}$$

Based on the fitness function Algorithm 4, w_i is optimised to minimise the MAE. The returned W vector represents the weight of each of trust feature and depends on the nature of the data in the given recommender system.

Algorithm 4 : The new Fitness Function Algorithm

```
Input: trust vectors: re-tweet(RT), Followers (L), Name, Bio and Image
Output: MAE
Y=0
for every u in N do
    Apply the optimized parameters from current generation to get trust value
    trust_(u,f) = 1/M Σ_{i∈M} w_i t_i
end for
Train the data with new trust values using SVR model
Predict rating using the trained model
Y = + MAE_u for every user in N
RETURN Y/N the average of MAE
```

5.5 Experimental Settings

5.5.1 The Dataset: SD2

As we pointed out earlier in Section 5.2, we need to collect a new dataset, namely Social Data 2 (SD2). It has the following properties:

- From Section 4.6, we reported the reasons behind collecting our dataset instead of using existing ones. By using the implemented source code of algorithms 1 and 2, the tool **TIE** can return the new trust features such as returning friends' names.

- We collected information about 120 users who have a trusted friend tweeted about a movie from 25 movies chosen randomly from Movielens.org, from July to August 2015. SD2 statistics are shown in Table 5.2. In addition, we explained earlier in Section 4.6 that our proposed approach based on people tastes and their intercommunication which subject to vary over time. Therefore, in this thesis, to achieve a suitable evaluation we have to rely on two datasets, SD1 and SD2, that are collected in two different times rather than one big dataset.

Table 5.2: Statistics of the SD2 dataset

Users	Movies	Max trust	Min trust	Max sr	Min sr
120	25	1	0.15	5	2.7

- Examples of three instances about the movie (*Little women*) are shown in Table 5.3 from the collected dataset SD2. The first row shows the information of a friend who posted a micro-reviews about the movie *Little women*. In the image feature, we put 1 to indicate the use of picture and not the default *egg* picture which is usually used in Twitter to represent profiles with no images. If the friend wrote a clear bio information, then this feature held the value 1, otherwise was set to zero. Also in the name feature, if the friend used a proper name, then this feature took the value 1, otherwise a zero was assigned if it is not a name, such as a sign.

Table 5.3: Examples from the dataset SD2

Users	user1	user2	user3
f's ID	35830xxxx	10962xxxx	46614xxxx
f's tweet	*Watching Little Women is interesting !!!!*	*with sister watching Little Women :)*	*super excited to be watching Little Women*
Followings	5146	165	222
Followers	5061	407	338
Image	1	1	1
Bio	1	1	1
Name	0	0	2
RT	10	4	10

5.5.2 Annotation and Data Agreement

This section gives a brief description about the annotation task to create our gold standard data. Gold standard data is the ground truth that is used to train, test, and evaluate algorithms that do empirical analysis. The annotators are asked to annotate a set of example data and perform analysis of the provided cases. In the machine learning sense, machines can learn from these example. A very important issue to be considered is the quality of annotation which is measured by computing inter-annotator agreements.

To study agreement between annotators in our dataset, SD2, we used the methodology *CrowdTruth*, illustrated in detailed in [125, 126, 127]. CrowdTruth is used in various fields to assess alignment of opinion; however, this approach is the first time to be applied in RSs. It is based on the assumption that there is a single, universally constant truth. Researchers try to overcome the fallacy of some common beliefs in building gold standards such as:

- **One truth**: Researchers usually, when collecting data, assume that there is only one correct interpretation for every single instance in the dataset. However, in Crowd Truth assessment, different opinions contribute to give the nearest interpretation to reality.

- **One is Enough**: Most annotated data are done by one person. However, in the used methodology, the higher number of annotators, the more acceptable the output.

- **Expert are better**: Expert people specialising in one filed can provide the ground truth. However, experts are expensive to hire and several points of view regarding the provided cases can offer better general perspective.

Since we seek a higher quality ground truth, we intended to apply the CrowdTruth methodology for the following reasons: Twitter is a noisy environment and users' judgement towards each other are variable and subjective. It is obvious that there is not one unique correct interpretation for every input scenario.

These are the steps we applied in building our gold standards:

- First, we prepared 120 scenarios similar to Figure 4.3 based on the gathered trust information for each user connected with a trusted friend explained in Section 5.4.2. These scenarios, (Appendix A), are provided to annotators to build the ground truth.

- We used ten annotators (previously we used 3 annotators to assess ISTS in Chapter 4). The annotators provided their ratings on how they may trust this user's opinion about any item that the user tweets given this user's information.

- Then, the answer of each annotator for each scenario could be represented as a vector. In these vectors, a value of 1 is given to each category answer choice the annotators thought was being expressed (annotators can indicate multiple relations).

- After that, we used these answers to form sentence disagreement vectors for each sentence by summing all of the annotators vectors for the scenario:

$$V_{scenario} = \Sigma_{annotator}(W_{scenario,annotator}) \tag{5.5}$$

Example:

Let us consider the first three scenarios in Appendix A, and the ten annotators answers are gathered and presented as scenario vectors, described in Table 5.4.

Table 5.4: Examples of scenarios vectors

	Ratings Types				
Scenarios	1	2	3	4	5
scenario1	0	0	3	2	5
scenario2	0	3	6	1	0
scenario3	3	5	2	0	0

For example, by looking at the vector of scenario1, three annotators chose rating 3 for scenario1, two chose rating 2, and five annotators chose the rating 5 for scenario1.

- The rating for the scenario is measured for each possible answer provided as the cosine of the similarity of the unit vector for one rating with the scenario vector.

$$scenario_{rating} = cos(V_{scenario}, rating) \tag{5.6}$$

Example:

The unit vector of each rating is: Rating1=[1,0,0,0,0], Rating2=[0,1,0,0,0], Rating3=[0,0,1,0,0], Rating4=[0,0,0,4,0], Rating5=[0,0,0,0,5], by applying the cosine similarity measure between every scenario vector and the five ratings vectors. Table 5.5 shows the weight results.

Table 5.5: Cosine weight results

	Ratings Types				
Scenarios	1	2	3	4	5
scenario1	0	0	0.5571	0.3714	**0.7428**
scenario2	0	0.5071	**0.8452**	0.1690	0
scenario3	0.5571	**0.7428**	0.3714	0	0

- Finally, the higher value of the weights, the more clearly that the rating expresses the scenario. From Table 5.5, the bolded weights are the highest weights, and the rating scores selected to be rating5 for scenario1, rating3 for scenario2, and rating2 for scenario3. These results are used for training and evaluation of the research approach; it is viewed as the probability that the scenario expresses the rating. The premise that we worked with is that there is not only one right answer, and diversity of opinion was to be preserved.

5.5.3 Parameters Design

This section presents the finding of a kernel that has high accuracy and good performance to be included in the SVR model. After undertaking a quick test on the training dataset, the result showed a polynomial Kernel coded as (k1) was the best performing candidate. This is based on the high accuracy of (k1) compared to the different other kernels we used: Linear (k0), RBF (k2) and Sigmoid (k3), as shown in Table 5.6.

Table 5.6: Performance of different kernels in the SVR model on the training dataset

	Kernels Types			
	k0	k1	k2	k3
MAE	0.4633	**0.4311**	0.4584	0.5677
MPAE	16.9460	**15.8890**	16.9350	20.0290

Therefore, we chose the polynomial kernel to be adopted in the GA process to optimise the trust parameters. We have to define some H-ISTS training parameters before starting the algorithm. These parameters are illustrated in Table 5.7. It shows that we started with 20 as a population size and we carried out two experiments, one with 20 generations and the second with 50 generations. Hence, the termination criteria were 20 and 50 in each experiment. The operator selection type was tournament and a uniform mutation type was set at rate of 0.5.

Table 5.7: The main H-ISTS model training parameters

Parameter	Value
Population size	20
Generations	20-50
Selection type	tournament
Mutation type	uniform
Mutation rate	0.5
Problem	minimisation

5.6 Results and Discussion

From the previous details, this section illustrates the results of H-ISTS framework and provides an assessment of the H-ISTS performance.

5.6.1 Optimisation Results

Figures 5.3 and 5.4 present the performance of the H-ISTS by focusing on minimising *MAE*. The H-ISTS framework is implemented for 20 generations and then 50 generations. When we increased the number of generations from 20, Figure 5.3, to 50, Figure 5.4, we did not notice any significant decrease in error. It is clear that the best achievable MAE through the whole optimisation process in both experiments was 0.416.

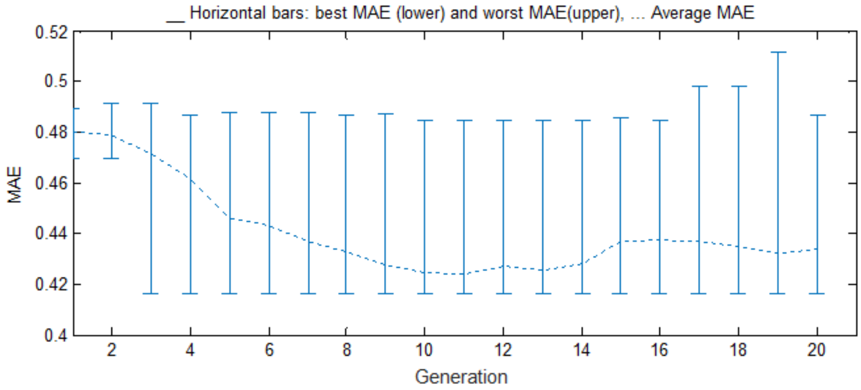


Figure 5.3: The Optimisation process of MAE in H-ISTS using 20 generations

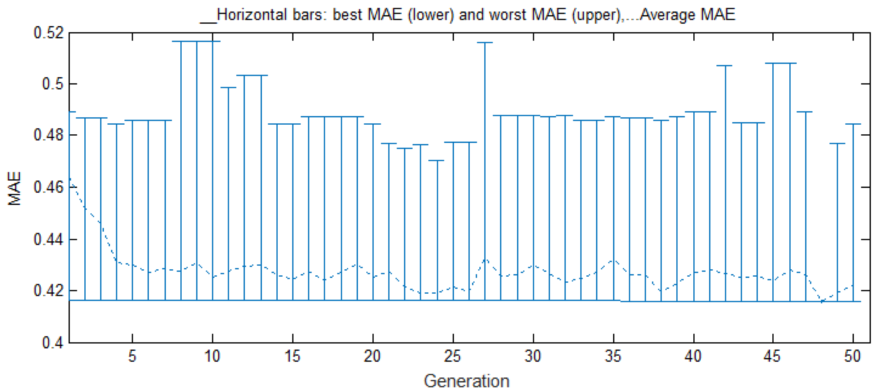


Figure 5.4: The Optimisation process of MAE in H-ISTS using 50 generations

5.6.2 H-ISTS Prediction Results using SD1

First, we divided SD1 into five random splits to apply cross validation tests. Then, every split set is used to test the trained model using the rest of the four splits. Errors were averaged over all trials. In addition, we tested the performance of H-ISTS in rating prediction for new users in two experiments:

(1) Ratings prediction for an active user assuming that there was one friend applied to every user in the SD1 dataset. Table 5.8 indicates that our model outperforms other baselines, since the MAE and MAPE were lowest at 0.4215 , and 15.7959% respectively.

(2) Ratings prediction assuming different random groups of friends 3, 7 and 11 sampled from the dataset for each of the new 12 users. From Table 5.9 we can see that the MAE results are always in the range [0.3808-0.4362] and the MAPE in the range [14.137%-16.296%], which is still the highest level of accuracy compared to the

baselines models. Therefore, the achieved performance of H-ISTS can represent how social trust optimised by GA and combined with friends' opinions can improve the quality of RS in terms of rating prediction accuracy.

Table 5.8: Accuracy performance when modelling only one friend per user

Algorithms	MAE	MAPE%
B1	0.5819	21.6318
B2	0.8165	31.5896
B3	0.5882	22.1179
B4	2.8411	94.6310
$ISTS_{regression}$	0.4500	16.0000
H-ISTS	**0.4215**	**15.7959**

Table 5.9: Accuracy performance when modelling 12 users with different friend list sizes

	MAE			MAPE%		
Algorithms	L3	L7	L11	L3	L7	L11
B1	0.4800	0.6152	0.5861	20.447	24.6486	23.1748
B2	0.7453	0.7877	0.7908	29.4194	31.1316	33.9059
B3	0.6149	0.5683	0.6094	24.9678	21.7700	22.2982
B4	2.7317	2.0453	1.6402	93.4479	74.3902	59.4767
$ISTS_{regression}$	0.4583	0.4633	0.4311	16.9400	16.9465	15.8896
H-ISTS	**0.3808**	**0.4362**	**0.3964**	**15.0503**	**16.2961**	**14.1370**

5.6.3 Performance Evaluation Using Statistical Hypothesis Test

Similar to hypothesis test setting in Section 4.7.4, we test the significance of performance improvement by looking at the P-value in the case of pair-t test using MATLAB. Compared with the baselines: $ISTS_{regression}$, B1, B2, B3 and B4, H-ISTS significantly improves the performance with P-values found less than the significance level 0.05 as described in Table 5.10, based on the experiment results of sparse group of friends, and the P-values which are based on the experiment results of different group of friends.

Table 5.10: The P-values

	Sparse group of friends	Different group of friends
$ISTS_{regression}$	0.04100000	0.03990000
B1	0.00006000	0.00110000
B2	0.00010286	0.00078000
B3	0.00350000	0.00090000
B4	0.00070000	0.00001100

5.6.4 Comparison between ISTS and H-ISTS

As we described before, H-ISTS is designed to improve the performance of ISTS in particular $ISTS_{regression}$. From Chapter 4, $ISTS_{regression}$ gives equal power to trust components RT and L, however, in H-ISTS these trust components are optimised by using GA. From Tables 5.8 and 5.9, the results of MAE and MAPE show better performance of H-ISTS against $ISTS_{regression}$ which indicate an important role of parameter optimisation step used to give different power to RT and L. Based on Equation 5.1, w is optimised by GA to be 0.79 and trust impact is optimised as follows:

$$trust_{u,f} = 0.79RT + 0.21 * L \tag{5.7}$$

It is important to mention that $ISTS_{regression}$ is based on two trust features, hence the next experiments have a different set up which is based on SD2 that includes rich trust features. For this reason $ISTS_{regression}$ is not involved in the comparison in the next experiments.

5.6.5 Results Based on SD2

Once the SD2 was prepared, we used it to carry out a range of experiments using Libsvm [128] and MATLAB with the same cross validation test settings. We are particulary interested in how the accuracy of recommendations is affected by the improved trust features information and the larger dataset.

Figures 5.5 and 5.6 illustrate the performance of our fitness function using the GA optimisation tool in MATLAB for 20 generations and 50 generations, respectively. In Figure 5.5, the data show the best fitness function value was 0.4413 for the 20 generation. Table 5.11 shows the wight parameters of $RT, L, Image, Name$ and Bio were 6.661, 9.853, 4.887, 4,09 and 2,611, respectively, and the simulation time was 21 minutes. For the 50 generation, Figure 5.6 shows that the best fitness function MAE value was 0.43919. From Table 5.11, the trust features weight parameters were 9.002, 15.125, 8.942, 8.738, 3.982, and the simulation time was 51 minutes. Running the algorithm several times it the delivered results were very similar results.

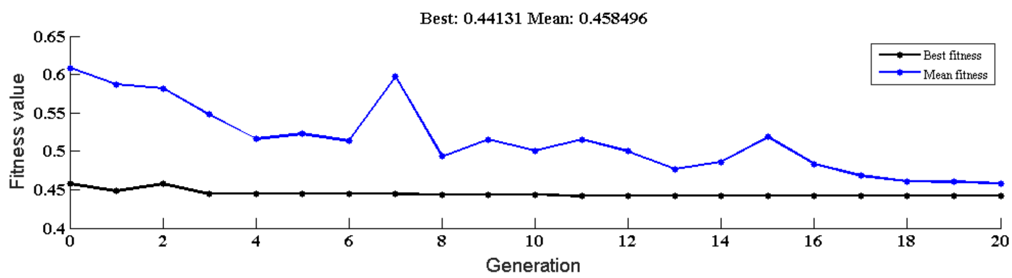


Figure 5.5: The Optimisation process of MAE in H-ISTS using 20 generations

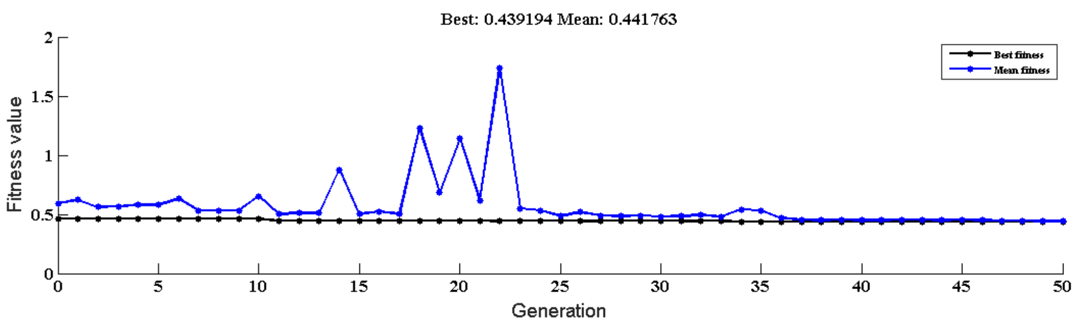


Figure 5.6: The Optimisation process of MAE in H-ISTS using 50 generations

Table 5.11: H-ISTS performance comparisons

Generations	Method	Var	RT	L	Image	Name	Bio	MAE
20	H-ISTS-SD1	2	0.790	0.210	-	-	-	0.4164
20	H-ISTS-SD2	2	0.241	0.759	-	-	-	0.4690
20	H-ISTS-SD2	3	4.231	8.189	-	4.586	-	0.4500
20	H-ISTS-SD2	5	6.661	9.853	4.887	4.090	2.611	0.4413
50	H-ISTS-SD1	2	0.800	0.200	-	-	-	0.4158
50	H-ISTS-SD2	2	0.375	0.625	-	-	-	0.4912
50	H-ISTS-SD2	3	4.491	7.450	-	2.525	-	0.4486
50	H-ISTS-SD2	5	9.002	15.125	8.942	8.738	3.982	0.4392

Table 5.11 demonstrates different H-ISTS performances in different experiments. H-ISTS is explored using the SD2 dataset. These simulations eliminated the rest of the identified trust features and kept only the two building blocks of trust in the $ISTS_{regerssion}$ which are *RT* and *L*. The results were: the w parameter is 0.241 and fitness function value is 0.469. Whereas when 50 generation, *w* for the RT feature was 0.375 and best fitness MAE was 0.4912. Another experiment was carried using three trust features RT, L and *Name*. The parameter weight values, over 20 generation, were 4.231, 8.189, 4.586, respectively, generating a fitness MAE value of 0.45. When simulating 50 generations, the results were 4.491, 7.45, and 2.525, respectively, with a fitness MAE value of 0.4486.

It is important to highlight some interesting differences about the importance of some features from Table 5.11. The highest weights were given to *RT* and *L* in the model results, however, *RT* was considered in the fifth place in the survey results,(Table 5.1). In the survey results, feature *Name* was at the top of the ranked list, however, in our model it is given a weight less than *RT* and *L*. Surprisingly, the *Bio* feature was considered to be in the first position of importance in the survey results, while based on our model results, the lowest weight that affects the trust was given to the *Bio* feature. Generally, this may happen because users, on OSNs, prejudge friends' accounts from the *Name* and the *Bio* information apparently, while *RT* and *L* may require more efforts to identify them.

5.6.6 Summary of Empirical Findings

In this section we highlight the following findings:

1. A GA has implemented with H-ISTS approach in several experiments that demonstrated that the feature *L* attained the highest weight among all the other selected trust parameters, *RT*, *Name*, *Image* and *Bio*. In contrast to SD2, *RT* was previously considered more important than *L* in SD1. The Bio feature always came as the least influential when building trust.

2. Weight values that obtained in the optimisation process for the parameters *Name* and *Image* indicated that they have close power effect on the trust metric.

3. The best overall optimisation results in SD1 privileged the feature *RT* and those in SD2 gave the highest support to the feature *L*.

4. However, using only *RT* and *L* while excluding the other features, *Name* and *Image* and *Bio* yielded less favourable accuracy results, Figure 5.11.

5. Based on the online survey and the empirical experiments results, we can say that the two features *RT* and *L* have an important effect when building trust. We can argue that *RT* and *L* can still be the most important trust factors that indicate trust between users on OSNs microblog services, and in particular Twitter.

6. Online recommenders have the choice of including the five trust features or use only the two features *RT* and *L*. This may still be dependent on the microblog service that they may adapt.

5.7 Areas of Improvement

Despite the progress of H-ISTS, it is important to ask how H-ISTS will perform with a collaborative filtering recommender. Arriving at this point, the ISTS and H-ISTS frameworks should be implemented to improve prediction of preferences of users who have a history of ratings. For this type of users, the systems usually know their preferences and therefore, it is important to investigate the fusing users' own tastes and their friends on OSNs to deliver more relevant and accurate recommendations. Hence, we need to reach the following goals:

- Combining the users' preferences of ratings and friends' tastes in rating predictions.

- Evaluating the coverage of user-item ratings matrix and accuracy level of predictions when H-ISTS framework and a CF recommender are integrated.

- Using H-ISTS to solve the inherent challenge of diversity problem.

The next chapter demonstrates the implementation of the above goals.

5.8 Chapter Summary

This chapter has described the H-ISTS approach, which is designed to address the limitations encountered with ISTS. New requirements have been satisfied with the H-ISTS approach by applying a GA for the optimisation process of trust parameters. Furthermore, a survey was conducted to identify the trust features that are most influential according to a sample of active Twitter users. The survey results illustrated that RT, the number of $Followers$, the $Name$, the $Image$ and the Bio of a user have the strongest influence on building trust between friends on Twitter. This chapter also introduced the use of CrowdTruth methodology in the annotation of SD2. Empirical experiments have been conducted to evaluate the H-ISTS framework performance, the results showed that H-ISTS has achieved the desired improvements. In relation to SD1, H-ISTS showed that RT has a higher power of influence on trust between users and friends. However, with the extended trust features as applied in SD2, the simulations showed that L has the highest influence on building trust. Hence, it can be concluded that both RT and L are the most influential factors required to indicate trust on the microblog services based on Twitter.

Chapter 6

Using the H-ISTS Approach to Enhance Collaborative Memory-Based Filtering Recommendation Methods

6.1 Chapter Introduction

This chapter describes the design and analysis of the integration of the novel H-ISTS approach with collaborative filtering (CF) recommenders to allow users who already have preferences to receive interesting personalised recommendations. It is important to study the influence of our approach on profiles of users who have already evaluated some items in recommenders. Users of CF recommenders can be provided with recommendations using others' similar profiles. Furthermore, users' friends on online social networks (OSNs) can assist their peers in such recommenders by providing valuable and interesting selections of items. Hence, these recommenders can exploit friends' preferences to augment and enrich users' profiles. In addition, having tastes different to friends' preferences can produce new and unexpected items that may increase the diversity in users' profiles in a very personalised manner.

In this chapter, Section 6.2 provides a brief review of memory-based methods; namely, user-based and item-based approaches, followed by an overview of the new research challenge of diversity in Section 6.3. Section 6.4 presents the design of the combination H-ISTS$_{\text{CF}}$ and illustrates the development of the required algorithms. Empirical experiments are detailed in Section 6.5. Section 6.6 presents the performance evaluation of H-ISTS$_{\text{CF}}$ in terms of findings related to H-ISTS$_{\text{user}}$ and H-ISTS$_{\text{item}}$, followed by further discussion and evaluation of the findings in Section 6.7.

Finally, Section 6.8 summarises the chapter.

6.2 Memory-Based Approaches: Background

Generally speaking, a recommender system is a predictor tool that exploits the historical recorded profiles of users in recommenders. RS analyses previous users' item ratings style to approach items with particular properties; for example the suggested items should match a user's previous choices. Moreover, users can sometimes require items that are novel and diverse, which present a particularly difficult challenge for recommender systems.

Collaborative filtering recommenders can be categorised into two types of techniques: memory-based and model-based approaches. As explained in Chapter 2, model-based approaches can use ratings information in users' profiles for learning in order to build a predictive model. Model parameters can be learned during the training step and later these models are used to predict new ratings.

On the other hand, solutions based on memory-based models can be applied using two known techniques; (1) user-based and (2) item-based methods.

6.2.1 User-Based Methods

User-based recommenders use other users' ratings, namely neighbours, as the basis of their prediction. Neighbours are selected based on having matching tastes and similar item rating behaviour to the target user. Items that have been highly liked by neighbours will be presented as suggestions to the target user. The more similar the rating style of one neighbour is to the style of a target user on the system, the more influential this neighbour's contribution becomes to the choice of items suggested for the target user.

In order to predict a rating R to a particular item i that an active user u requires, which will be denoted as $R_{u,i}$, weighted average of all ratings to that item is calculated based on the widely used formula proposed by Resnick et al. [7] and defined as follows:

$$R_{u,i} = \bar{r}_u + \frac{\Sigma_{a \in U}(r_{a,i} - \bar{r}_a).w_{u,a}}{\Sigma_{a \in U}|w_{u,a}|} \tag{6.1}$$

where $\bar{r}_u$ and $\bar{r}_a$ represent the averages of items ratings for user u and the neighbour a, respectively. The weight $w_{u,a}$ represents the extent of similarity between user u and neighbour a for common rated items in the past. These weights can be computed using the two most popular approaches, Pearson correlation coefficient (PCC) and cosine-based methods, as discussed in Sections 2.4.1 and 2.4.2.1, respectively.

6.2.2 Item-Based Methods

To predict the rating R for user u about missing item i, item-based methods are carried out in accordance to the ratings given to similar items to item i [8].

In contrast to user-based method, the similarity weight will be computed between items. For every user $u \in U$ who rate both items i and j they will be included in PCC similarity as follows:

$$w_{i,j} = \frac{\Sigma_{u \in U}(r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\Sigma_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\Sigma_{u \in U}(r_{u,j} - \bar{r}_j)^2}} \tag{6.2}$$

where $r_{u,i}$ and $r_{u,j}$ is the ratings of user u to items i and j. And $\bar{r}_i$ is the average ratings given to i while $\bar{r}_j$ is the average ratings of item j.

Now, to predict a rating $R_{u,i}$ for user u about item i according to the set I of rated items by u using the item-based method, can be defined using a weight average as follows:

$$R_{u,i} = \frac{\Sigma_{n \in I} r_{u,n} w_{i,n}}{\Sigma_{n \in I} |w_{i,n}|} \tag{6.3}$$

where $r_{u,n}$ represents the rating of all items $n \in I$ rated by user u, $w_{i,n}$ is the weight of similarity between i and n.

6.2.3 Motivations for Using Memory-Based Approaches

The proposed H-ISTS approach will be integrated and linked to memory-based collaborative recommenders, and therefore, this section describes the motivated properties of memory-based collaborative recommenders. There is an emerging belief that users' satisfaction and the quality of item recommendations cannot be guaranteed by only prediction accuracy [129]. Memory-based approaches excel in capturing local association in data, and therefore, systems which depend on such approaches can provide users with some items, which are very different from their usual preferences [129]. This occurs because memory-based techniques use the nearest neighbours ratings to be provided to the target user. In this way, users may explore new tastes and experiences. We can outline the strongest points of memory-based over model-based methods [129]:

- *Simplicity*: while model-based approaches require to set up thresholds before training and other parameters during the training stage, memory-based approaches are simpler to implement and only require to tune the number of ratings and the testset proportions.

- *Justifiability*: memory-based methods can provide an explanation to the target user to understand the recommendations. The highest neighbours ratings can be used to justify the list of the recommended items.

- *Efficiency*: one of the main advantages of memory-based systems is efficiency. The similarity calculation which is crucial in memory-based systems can be done offline which dose not cost as much as model training. The cost of training model steps need to be implemented frequently in online business systems. Moreover, the memory-based needed to store the closest neighbours is small and this is scalable to huge systems with millions of users and products.

- *Stability*: memory-based methods are not highly affected by changes in users and items. For example, memory-based recommenders can capture changes in users' tastes without the requirement to repeat the training step, which is faster than model-based methods, and hence, new trends in users preferences can be adopted.

6.3 Additional Challenging Issue

General challenging issues in CF recommendation methods were described in Section 2.4.2.3. We designed the desired features in the proposed novel research approaches, which are presented in Chapters 4 and 5 of the thesis. In this chapter, since we are investigating the integration of H-ISTS framework and a memory-based CF recommender, we need to explore an additional challenge that is required to be satisfied. This challenge is described as the diversity challenge which is discussed below.

6.3.1 Diversity: Context and Motivation

When investigating this challenge, diversity is generally defined as the opposite of similarity [130]. In fact, when very similar items are recommended to a user every time, this is not useful for the user's needs or the systems needs. In other words, users need to explore different items that may acquire their attention. On the other hand, recommenders require exposing the range of items to the users in a reasonable way. Consider, for example, the case of a vacation package recommender. It might be more useful for a user to receive suggestions of the top-n recommendations for different locations varying in different subsets of detail determined and tuned by the user, rather than recommending top-n locations varying only based on hotels choices [131]. However, in this thesis, diversity is defined as unexpected and extended suggestions different from items that appear in the current user's profile.

In this section, we answer the question of why recommendation methods may add diversity as a desired feature to improve the quality of recommenders. Diversity is a recent concern which has attracted recommendation literature, and has become significant properties to be considered when designing a high quality recommender system. Despite the recent research emphasis on this area, diversity approaches still need to be improved and further developed in many aspects [130, 131, 132, 133]. Below are motivation points in relation to diversity challenge in RSs:

- **Users' interests are variable and include a high level of uncertainty**: Users' needs are heterogeneous and sometimes are contradictory. The analysis of recommenders of users' ratings and activities tend to explore very limited information about this complexity and uncertainty of interests. Therefore, diversity is used as a strategy to allocate suitable user preferences in an uncertain environment.

- **Natural human behaviour**: From a user's perspective, it is worth mentioning that diversity is desirably needed in order to acquire user satisfaction. As advocated by researchers in consumer behaviour studies, evidence shows that the variety-seeking tendency is a natural human behaviour [134]. People experience inherent satisfaction when they receive unexpected and unfamiliar ideas, while people satisfaction tends to drop with repeated consumption of the same type of products [135]. Diversity provides broad users' profiles and enrich their experiences by recommending new interests and tastes.

- **Expanding the business**: Diversification of items is a business strategy used to expand businesses and obtain more market benefits. Therefore, diversity is maintained in business to increase system activities, build customer loyalty and improve revenue [136].

The most popular method in evaluating diversity is to measure the distance between the recommended items in the list of recommendations, which is estimated based on similarity [133, 137]. Therefore, one way of evaluating this parameter is to assess the sum of the average of similarity measure between item pairs in the resulting profile. However, attempting to address diversity may come at the expense of providing less accurate recommendations [138].

In order to increase the added value of recommendations, we propose in this chapter the role of the H-ISTS framework to obtain the desired level of diversity while maintaining accuracy of recommendation. We therefore investigate how inferred friends' opinions can be a useful strategy to expand the set of recommended items, which have the potential to satisfy a user with varied items, rather than restricting their choice by providing repeated suggestions.

6.4 Problem Formulation and Algorithms

In this section, the integration of the H-ISTS approach to a collaborative filtering environment, denoted as H-ISTS$_{\mathrm{CF}}$, is discussed. This is to allow users in a collaborative filtering environment to receive valuable recommendations when linking our novel approach to a collaborative filtering recommender.

Figure 6.1 shows an example of original ratings matrix in the recommender system. Users rate a small portion of items and any missing items ratings are not-known to users. Our H-ISTS approach solves this issue by investigating trusted friends on OSNs about these missing items, and subsequently, rating predictions are made using friends' opinions. By doing so, users' selections will be augmented in accordance with the opinions of their trusted friends. In Figure 6.1, the resulting user-item ratings matrix shows the estimation of ratings for missing items.

In this chapter, we aim to satisfy the following goals:

- The design, implementation and evaluation of the integrated approach H-ISTS$_{\mathrm{CF}}$. The algorithm implementation include user-based and item-based methods. The prediction accuracy level for users who have a history of ratings is evaluated.

- The production of valuable recommendations to friends. Because of friends' contribution to the recommendations process, these friends can receive suggestions derived from the active users' profiles. The highest predicted ratings of items are provided to users' friends as a return.

- Analysis and evaluation of the coverage metric of user-item rating matrix by analysing the total number of items that H-ISTS$_{\mathrm{CF}}$ can predict.

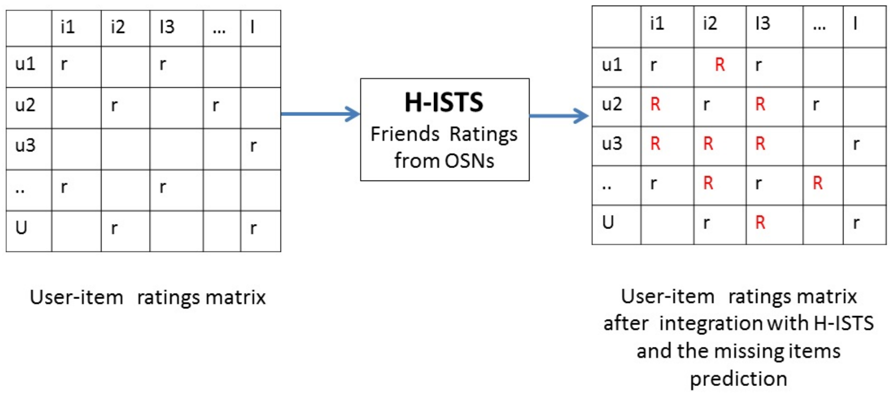


Figure 6.1: The integrated approach: H-ISTS$_{\text{CF}}$

- Investigation of the diversity challenge and assessing the improvements in diversity for users' profiles.

6.4.1 Notations

The notations used in this chapter are summarised below.

- H-ISTS$_{\text{user}}$: a user-based method when integrated with H-ISTS.
- H-ISTS$_{\text{item}}$: an item-based method when integrated with H-ISTS.
- H-ISTS$_{\text{user1}}$: a user-based method when integrated with H-ISTS using SD1.
- H-ISTS$_{\text{user2}}$: a user-based method when integrated with H-ISTS using SD2.
- H-ISTS$_{\text{item1}}$: an item-based method when integrated with H-ISTS using SD1.
- H-ISTS$_{\text{item2}}$: an item-based method when integrated with H-ISTS using SD2.
- FL_i: an item in the friend's list.
- FL_u: the group of friends of a particular user u.
- $FL_{f,i}$: item i recommend by a friend f in the list of friends FL.
- $ind(r_{u,i})$: the availability of a rating of user u about item i.
- $sim_{item}(i_n, i_m)$: similarity weight computed between two items i_n and i_m.

6.4.2 Algorithms

The proposed approach H-ISTS$_{\mathrm{CF}}$ is described in the following algorithms for both methods, user-based and item-based.

Algorithm 5 describes the model H-ISTS$_{\mathrm{user}}$. It includes two levels; the first level represents the implementation of H-ISTS carried out by acquiring users' twitter accounts and verifying the features needed to apply the H-ISTS model. It returns a friends' list FL for every user. Every user u has a defined followings list FL. Friends' posts on twitter are searched for any missing items i in the user-item ratings matrix. After that, every friend in FL is associated with inferred rating $SR_{f,i}$ towards item i.

Then, users in the CF recommender are checked for two cases. If the user is new to the system then their friends' opinion SR is used to replace the missing item. Otherwise, for users with existing profiles of ratings, items must be addressed as follows. If one item is not missing and friends provide an opinion about it, then $SR_{f,i}$ is ignored. In other words, friends' opinions are considered only when the corresponding rating in a user-item matrix is missing. The second level starts with computing user-user similarity weights according to Equation 2.11. Then, the user-based Equation 6.1 is applied for every user in order to build the predicted matrix R and estimate ratings for unrated items.

It is worth noting that these data are dependent on a retailer's tendency and frequency of storing this type of information in their logs for specific periods of time, for example every week or every three days.

In the second algorithm, Algorithm 6, the first level starts by searching users' profiles regarding missing items i. Then, friends' posts are analysed and explored for any opinions about the missing items. Subsequently, H-ISTS is applied to estimate friends' opinions as SR. In this way, new users adapt their choices based on their friends opinions on all inferred sentiment rating $SR_{f,i}$. However, users who have rated items before will accept their friends ratings SR only about the unexplored items. Next, the second level starts by computing item-item similarity weights to allocate the closest neighbourhood sets to every user using Equation 6.2. After that, item-based Equation 6.3 is used to predict the matrix R and provide ratings for unexplored items by users.

It is important to mention that the threshold that we used is 0.1 to filter neighbourhood group, based on the study by Ghazanfar [118]. This is because two reasons, first, the study tested different thresholds under the same popular dataset SML that we also used in our experiments. The author showed that the results of the lowest MAE was obtained when using similarity threshold=0.1. Second, our goal in this chapter is to

Algorithm 5 : H-ISTS$_{\text{user}}$

Input:user-item matrix M, users's twitter acct
Output: prediction of all missing items $r_{u,i}$ in the predicted matrix: R
determine the sets of the missing items for every user
First level: H-ISTS
connect users' twitter accounts
for every $u \in U$ **do**
    $FL_u \leftarrow H-ISTS$
end for
for every u_n **do**
    if $u_n isnewandi = 0$ **then**
        $u_n = FL_u$
    end if
    for every item in FL_i **do**
        if i=0 and FL_i not exist in user profile **then**
            $M(u_n, i) = SR_{f,i}$
        end if
    end for
end for
Second level: User-Based Algorithm
build neighbourhood users-user similarity, chosen THRESHOLD=0.1
$w_{u,a} = \frac{\Sigma_{i \in I_{a,u}} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\Sigma_{i \in I_{a,u}} (r_{a,i} - \bar{r}_a)^2 \quad \Sigma_{i \in I_{a,u}} (r_{u,i} - \bar{r}_u)^2}}$
Rating Prediction
$R_{u,i} = \bar{r}_u + \frac{\Sigma_{a \in U} (r_{a,i} - \bar{r}_a).w_{u,a}}{\Sigma_{a \in U} |w_{u,a}|}$
RETURN R

Algorithm 6 : H-ISTS$_{\text{item}}$

Input:user-item matrix M, users's twitter acct
Output: prediction of missing items $r_{u,i}$ in the predicted matrix R
determine the sets of the missing items for every user
First level: H-ISTS
connect users' twitter accounts
for every $u \in U$ **do**
  $FL_u \leftarrow H - ISTS$
end for
for every u_n **do**
  if $u_n isnewandi = 0$ **then**
    $u_n = FL_u$
  end if
  for every item in FL_i **do**
    if i=0 and FL_i dose not exist in user profile **then**
      $M(u_n, i) = SR_{f,i}$
    end if
  end for
end for
Second level: Item-Based Algorithm
build neighbourhood based on item-item similarity, chosen THRESHOLD=0.1
$w_{i,j} = \frac{\Sigma_{u\in U}(r_{u,i}-\bar{r}_i)(r_{u,j}-\hat{r}_j)}{\sqrt{\Sigma_{u\in U}(r_{u,i}-\bar{r}_i)^2}\sqrt{\Sigma_{u\in U}(r_{u,j}-\bar{r}_j)^2}}$
Rating Prediction
$R_{u,i} = \frac{\Sigma_{n\in N} r_{u,n} w_{i,n}}{\Sigma_{n\in N}|w_{i,n}|}$
RETURN R

study the effect of the integration of research approach ISTS to enhance memory-based algorithms, and not to investigate the similarity in pure memory-based algorithms.

6.4.3 Non-accuracy Metrics

We have used the accuracy metrics, such as MAE and MAPE in this thesis, in Chapter 4 and 5. Although these metrics have been successfully and widely used throughout the literature, a user's satisfaction needs to be measured not only by the accuracy of prediction. Recently, non-accuracy metrics have attracted more interest in the area of recommendation research, an attempt to capture and better represent user satisfaction [132]. Coverage and diversity metric are non-accuracy evaluation methods that have been applied in this work, and these methods are described next.

6.4.3.1 The Coverage Metric

Coverage is an important element when testing the quality of recommender systems. Coverage measures the ability of the online business service to recommend range of items in their catalogue. This metric shows if the recommenders' engines are successful in recommending more items than the existing ones.

We consider the coverage as the percentage of items that can be predicted and provided to customers compared to all the available ones and so-called **catalogue coverage** [132]. Alternatively, coverage metric can be measured based on the user profile. In other words, this type of coverage is achieved by computing the number of rated items for every user that are required in order to make recommendations. In this work, we considered the catalogue coverage metric in order to satisfy the research requirement of alleviating the new users cold-start problem, it is defined as:

$$coverage = \frac{\Sigma_{u \in U} \quad \Sigma_{i \in I_u} ind(r_{u,i})}{|R|} \tag{6.4}$$

$$ind(r_{u,i}) = \begin{cases} 1, & \text{if } r_{u,i} \in [1-5] \\ 0, & \text{otherwise} \end{cases} \tag{6.5}$$

for every user u in the set of users U, and the index $ind(r_{u,i})$ indicates the existence of a rating in an index in the user-item rating matrix, R, for every item i in the set of the rated item of user u, I_u, divided by the available ratings in the matrix R .

6.4.3.2 Diversity Metric

We used a diversity metric which we defined, in Section 6.3.1, as unexpected and extended suggestions different from items that appear in the current user's profile. This metric is used to measure the distances between every item in the resulted profile for every user. In [133], the authors introduced the Intra-List Similarity (ILS) metric to capture diversity.

For every user $u_n \in U$, we need to measure the inter item similarity among items in user's profile pr_u, hence, $sim_{item} : I \times I \rightarrow [-1, +1]$. $ILS(pr_u)$ is defined using sim_{item} as follows:

$$ILS(pr_u) = \frac{\sum_{i_n \in pr_u} \sum_{i_m \in pr_u, i_n \neq i_m} \quad sim_{item}(i_n, i_m)}{2} \tag{6.6}$$

where every item similarity is compared to the rest of items in user's profile, where similarity $sim_{item}(i_n, i_m)$ is between items i_n and i_m in user's profile pr_u. As this metric shows how dissimilar the recommended items are from each other, therefore, smaller values are better indicators of diversity.

6.5 Experimental Settings

In this section, we start by introducing the baselines methods that used to benchmark H-ISTS$_{\text{CF}}$. We use the baseline B1 from Section 4.6.2 and four additional baselines:

- **B1**: Item Average Score baseline (IAS), this method assigns for any unknown missing values the average of the corresponding item ratings.
- **B7**: The user-based method proposed by Resnick et al. [7], described in Section 6.2.1.
- **B8**: The item-based method proposed by Sarwar et al. [8] described in Section 6.2.2. The authors claimed that item-based can be more accurate than the user-based method.
- **B9**: The method proposed by Breese et al. [31]; this user-based method are presented where the main idea is to boost ratings by default ratings to decrease sparsity in the user-item rating matrix. Authors claimed that their method outperformed the user-based method.
- **B10**: This approach is used in the literature by combining the user-based and item-based methods and computing the average of their results.

6.5.1 Dataset Description

In the domain of recommender systems, it is widely common to test approaches and recommender algorithms from movie recommendation websites. These websites provide different datasets and make them available for researchers. In the literature, researchers used to benchmark their recommendation algorithms and analyse the claimed quality of the systems. The MovieLens group make different datasets available online with different sizes and properties, such as MovieLens 100K ratings, which was released in April-1998 and is based on an integer scale from 1 (bad) to 5 (excellent). In this work, we have used MovieLens dataset to validate our proposed approach, which is based on MovieLens ML10 ratings (denoted as **SML**). This dataset is released in 2009 includes 71567 users selected at random, 10681 movies, and 10000054 ratings on a floating point scale from 1:0 to 5:0 with an interval of difference 0,5. Characteristics of this datasets fit with our research requirements because of three main points. First, we are testing trust based on the opinions of users' friends in Twitter. We need a dataset considered quite recent to have the chance to match some existing movies in tweets with the SML10 dataset. In fact, people on twitter post opinions about a variety of movies. Hence, the chance of our algorithm allocating tweets contains movies become higher. Second, the inferred friends' ratings that we computed are also based on a floating point scale, which perfectly match the rating data provided by SML. Third, the SML dataset focuses on every user's profile rather than the combination of ratings movies, which provides the needed information for our model. We used a subset of 50 users' profiles of ratings taking from this dataset.

In order to simulate the research approach describe above, a real world scenario can be obtained using two datasets:

1. Users' ratings towards movies are represented in the dataset SML.
2. Social dataset that we have collected and applied in the previous Chapters 4 and 5, SD1 and SD2. These datasets represent the friends' connections to users. Subsets of these datasets are assigned to every user in SML.

6.6 Performance Evaluation

This section describes the performance of H-ISTS$_{\mathrm{CF}}$ after implementing the above settings. Results are shown based on different metric, MAE, Coverage enhancement, and diversity.

All evaluations in the following experiments are implemented and coded in MATLAB. The results are reported for all users, where we hide some of the users' item ratings and then apply the coded algorithms. Subsequently, we carried out comparisons between the actual item ratings in users' profiles and the estimated ones after the running the implemented algorithms.

6.6.1 Analysis of the Performance of H-ISTS$_{\mathrm{CF}}$

Performance analysis for the integrated was carried out by including different scenarios to investigate any improvements obtained. The SML dataset has the following features: the sparsity level in the user-item matrix is at 0.9983, and hence the rating density is: 0.0017, which reflects a low level of ratings.

The following sections show the results of performance evaluation for H-ISTS$_{\mathrm{CF}}$ using both user-based and item-based methods.

6.6.1.1 H-ISTS$_{\mathrm{user}}$ Performance

This section presents the results obtained from the implementation of the H-ISTS$_{\mathrm{user}}$ algorithm. This algorithm solves the problem based on the perspective of the users and their friends. Table 6.1 shows the MAE and coverage of H-ISTS$_{\mathrm{user1}}$ using SD1 and H-ISTS$_{\mathrm{user2}}$ using SD2. Comparing the MAE of the traditional user-based denoted baseline model, as B7, 0.7434 to that of H-ISTS$_{\mathrm{user1}}$ and H-ISTS$_{\mathrm{user2}}$: 0.7516 and 0.7810, errors are slightly increased. However, it is noticeable that there was the improvement in the coverage metric from the original user-item ratings matrix (0.0017) after implementation of both proposed approaches H-ISTS$_{\mathrm{user1}}$ and H-ISTS$_{\mathrm{user2}}$. This table suggests a trade off between MAE and coverage results. That means increasing the matrix coverage using the predicted items comes over a decrease in the level accuracy in predictions.

Table 6.1: H-ISTS$_{\mathrm{user}}$ performance

	H-ISTS$_{\mathrm{user1}}$	H-ISTS$_{\mathrm{user2}}$	B7
MAE	0.7516	0.7477	0.7434
Coverage	0.0269	0.0277	0.0206

The plot in Figure 6.2 shows the performance of algorithm H-ISTS$_{\mathrm{user}}$ compared to the original matrix coverage. It is observed that H-ISTS$_{\mathrm{user1}}$ and H-ISTS$_{\mathrm{user2}}$ show better coverage performance than the coverage obtained from the traditional user-based model B7.

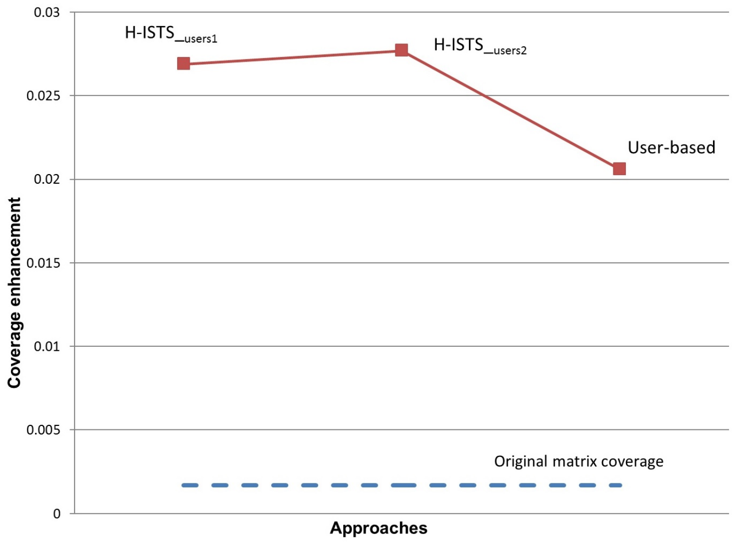


Figure 6.2: Coverage metric of H-ISTS$_{\text{user1}}$, H-ISTS$_{\text{user2}}$ and the user-based baseline (B7)

6.6.1.2 H-ISTS$_{\text{item}}$ Performance

In this experiment, recommendations prediction is based on H-ISTS$_{\text{item}}$. Table 6.2 shows that the proposed algorithms outperform other models in terms of accuracy. It is also noticeable that H-ISTS$_{\text{item1}}$ and H-ISTS$_{\text{item2}}$ are very comparable in achieving the lowest levels of MAE which are 0.7117 and 0.7091, respectively. The baseline item-based algorithm, denoted B8, shows a higher degree of error compared to the proposed approaches.

Table 6.2: H-ISTS$_{\text{item}}$ performance

	H-ISTS$_{\text{item1}}$	H-ISTS$_{\text{item2}}$	B8
MAE	0.7117	0.7091	0.7537
Coverage	0.0265	0.0271	0.0221

Table 6.2 also shows that our proposed algorithm improve the coverage metric significantly. In particular, H-ISTS$_{\text{item2}}$ provides an increase in the coverage metric by 0.0271 which is the best overall coverage level in this experiment. The H-ISTS$_{\text{item2}}$

algorithm is therefore successful in improving the original matrix coverage to a satisfactory coverage result. On the other hand, from Figure 6.3 the lowest coverage is given by the traditional baseline user-based method, B8, which enhances the original matrix coverage metric to 0.0221.

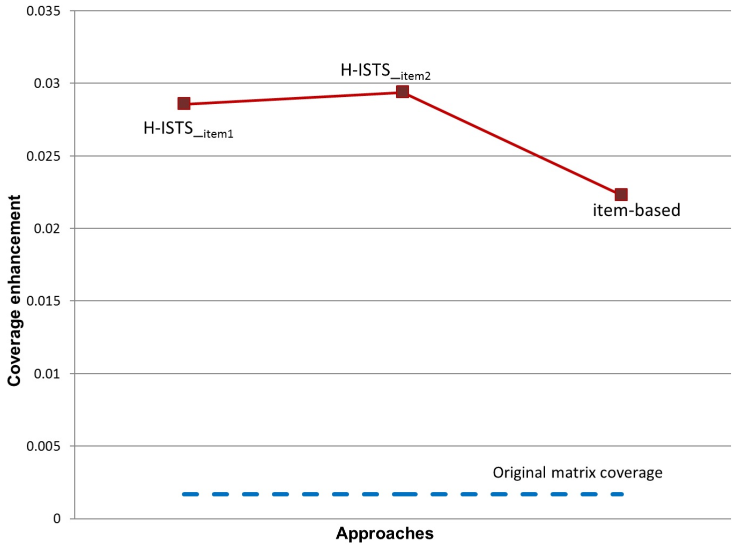


Figure 6.3: Coverage metric of H-ISTS$_{\text{item1}}$, H-ISTS$_{\text{item2}}$ and the item-based baseline (B8)

Therefore, the proposed approach H-ISTS$_{\text{item}}$ performs better than item-based algorithm, B8, in terms of the accuracy and coverage. However, H-ISTS$_{\text{item}}$ outperforms the H-ISTS$_{\text{user}}$ only in terms of accuracy.

6.6.2 Comparison of Accuracy with Baselines Algorithms

Table 6.3 gives an overall comparison of the performance of different baseline algorithms with our proposed approaches in terms of the accuracy metric, MAE, using the subset from the SML dataset. In addition, the results of our approaches are reported using SD1 and SD2. The baseline B1 gives the worst accuracy level, with MAE of 0.97. The traditional memory-based baselines B7 and B8 show superior performance to B9. Our proposed algorithms H-ISTS$_{\text{item1}}$ and H-ISTS$_{\text{item2}}$ outperform all other algorithms, with MAE values of 0.7117 and 0.7091, respectively, whereas the proposed approaches H-ISTS$_{\text{user1}}$ and H-ISTS$_{\text{user2}}$ perform at a comparable level to baseline algorithms: B7 (user-based algorithm), B8 (item-based algorithm), B9(method proposed by Breese et al.) and B10 (the average of combined user-based and item-based models).

Table 6.3: H-ISTS$_{\mathrm{CF}}$ performance compared to baseline approaches

Algorithms	MAE
B1	0.9700
B7	0.7434
B8	0.7537
B9	0.7460
B10	0.7520
H-ISTS$_{\mathrm{user1}}$	0.7516
H-ISTS$_{\mathrm{user2}}$	0.7477
H-ISTS$_{\mathrm{item1}}$	0.7117
H-ISTS$_{\mathrm{item2}}$	0.7091

6.6.2.1 Performance Evaluation Using Statistical Hypothesis Test

From previous evaluations, we can see that H-ISTS$_{\mathrm{item}}$ improves the performance with the smallest MAE, this is confirmed significantly by testing the P-value in the case of paired t-test compared with the baseline recommenders. P-values are found to be less than 0.044 in both datasets, which is smaller than the significance level 0.05.

However, H-ISTS$_{\mathrm{user}}$ was not significant as the P-value was about 0.74 compared with significance level 0.05, this is also can be inferred from Table 6.3 as it shows that baselines such as B7, B8 and B9 perform better in term of accuracy. However, H-ISTS$_{\mathrm{user}}$ obtains better performance in diversity as it will be explained later.

6.6.3 Top-N Recommendations for Friends

This section illustrates how the friends FL_u of every user can receive useful recommendations using the proposed approach. The active user is provided a recommendation list that shows the highest interests, with predicted item ratings greater than 4 considered as very interesting items. From Figure 6.4, we can see how many every active user can possibly suggest Top-N interesting items to all friends in FL_u for every user. The average number of recommended items is 540 movies shown as the horizontal line. The recommended list includes the original active user item ratings along with the collaborative algorithm results.

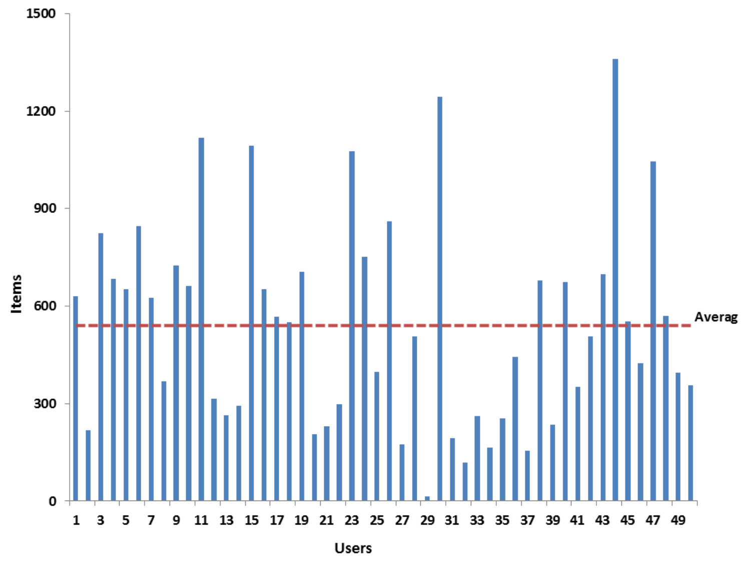


Figure 6.4: Items that can be recommended to friends based on H-ISTS$_{\text{CF}}$ and SD_2

6.6.4 New Users' Profiles

A set of new users in the dataset, eight users, were provided by rich recommendations with an average of 359 recommended movies per user as shown in Figure 6.5. Traditional collaborative filtering recommenders failed to allocate any items to these users profiles, with simulation outputs returning zero recommended items for these new users. The newly obtained recommendations using the proposed approaches can be used as a starting point to augment new users' profiles and those that show a high level of sparsity.

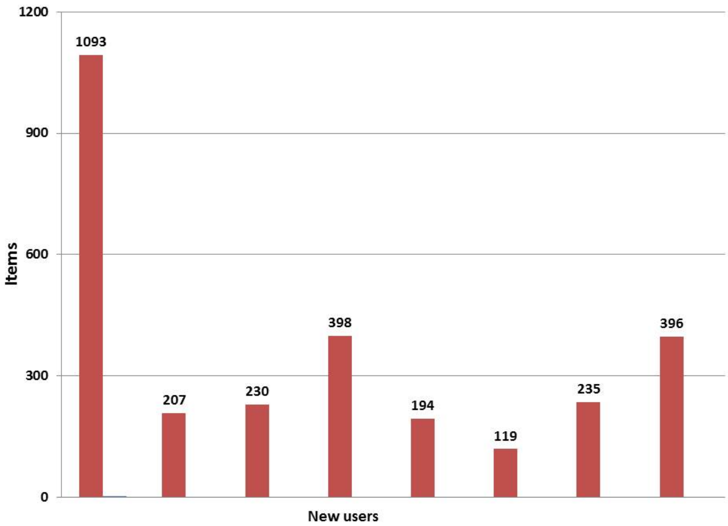


Figure 6.5: The number of items in new users' profiles recommended using H-ISTS$_{CF}$ and SD_2

6.6.5 Measuring Diversity

In this section, we describe how the proposed H-ISTS$_{CF}$ can improve the quality of the resulting suggestions by offering more variety of interesting recommendations that users have not explored before. Therefore, the increase in diversity is measured and analysed and the results are shown in Figures 6.6 and 6.7 using ILS metric specified in section 6.4.3.2. These figures show that the original profile diversity is 0.1275. There is an increase in similarity between items in users' profiles when using item-based models leading to diversity metric values of 102.21 in traditional item-based method, and for H-ISTS$_{item1}$ is 135 in Figure 6.6, and for H-ISTS$_{item2}$ is 141.17 in Figure 6.7. On the other hand, user-based methods produced less similar items in users' profiles, compared to the initial profile diversity. Based on user-based methods, the profile diversity level is -2.074 while using H-ISTS$_{user1}$ is -2.53, Figure 6.6, compared to -3.35 H-ISTS$_{user2}$ in Figure 6.7.

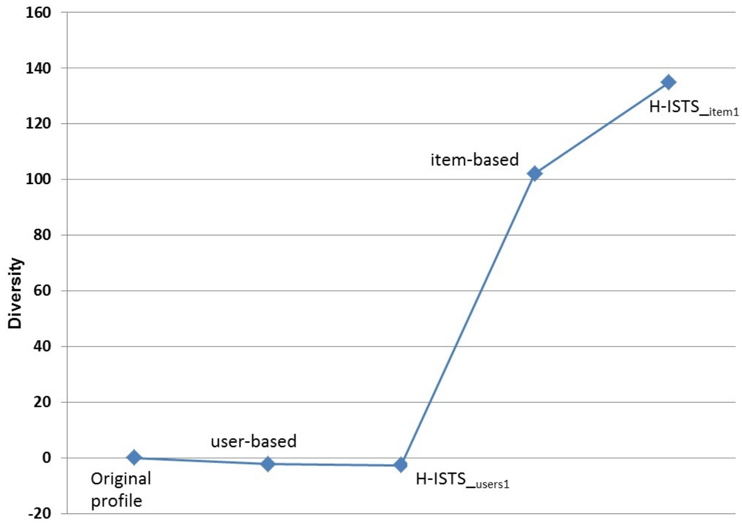


Figure 6.6: Average diversity in user-based, item-based, H-ISTS$_{\text{item1}}$ and H-ISTS$_{\text{user1}}$

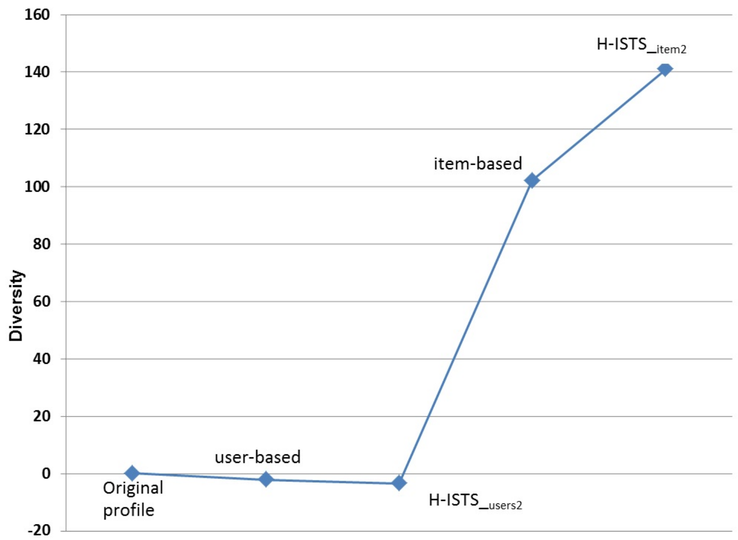


Figure 6.7: Average diversity in user-based, item-based, H-ISTS$_{\text{item2}}$ and H-ISTS$_{\text{user2}}$

Therefore, these results combined show that the dissimilarity, and therefore, higher diversity, when employing user-based approaches, especially H-ISTS$_{\text{user2}}$.

6.7 Further Discussion

It is evident from the empirical results that collaborative filtering systems can be highly affected by information coming from OSNs. Results from our proposed approach are very promising compared to state-of-the-art baseline models. This implies that OSNs can be a valuable source to boost a user's profile on recommender systems. Specifically, integrating the developed H-ISTS with a recommender system can allow friends from online social communities to contribute to an important role in allocating highly novel personalised quality suggestions to users of different online businesses.

We can outline the following interesting findings:

1. The proposed approaches H-ISTS$_{\text{item1}}$ and H-ISTS$_{\text{item2}}$: In term of accuracy, these methods scored the best overall error rates which means that the predicted items were very close to the actual ones. This is consistent with findings reported by Sarwar et.al [8]. However, the report showed a decrease in performance in coverage and the diversity metric. The implementation of H-ISTS$_{\text{item1}}$ and H-ISTS$_{\text{item2}}$ algorithms yielded lower diversity results, and therefore, high accuracy predictions are expected to be generated by these models for very similar recommended items.

2. The proposed approaches H-ISTS$_{\text{user1}}$ and H-ISTS$_{\text{user2}}$: These algorithms produced very comparable accuracy results to baselines. Results in terms of diversity showed that H-ISTS$_{\text{user1}}$ and H-ISTS$_{\text{user2}}$ algorithms offer the best diversity in users' profiles. It is worth noting that this diversity occurs in users' profiles created based on friends' opinions. Because users tend to be influenced by their friends' tastes, these recommended items are considered as interesting suggestions, even in cases when users and their friends have different tastes.

3. Because of the fact that our approaches consider friends' points of view, therefore, recommendations can also be provided to friends based on the users tastes. In other words, an active user can participate effectively in finding potential interesting items to be recommended to a friend. Hence, as this friend's opinion is inferred as a strong positive rating towards an item this inferred rating will implicitly attract more similar items in the user-item matrix when predicting the active user's profile. In this way, this friend can acquire some more item recommendations when their respective profile is provided with highly rated items from the predicted active user's profile. As the proposed algorithms seeking the

best recommendations through out the neighbourhood of users or items within the system, friends' preferences guide the algorithms significantly towards similar items, thereby, novel and diverse items can be recommended to the active user.

4. New users to the recommender can obtain item recommendations using the proposed approach H-ISTS$_{\mathrm{CF}}$. The profiles of new users will be expanded with rich personalised recommendations, whereas traditional collaborative filtering algorithms failed to provide new users with any predicted items. This is because collaborative filtering recommenders need users to rate items first, using similarity measurements, the system can be applied across the user-item rating matrix. Thus, new users and users who rated only few items have traditionally received poor recommendations in these conventional collaborative filtering recommenders.

5. The different proposed algorithms can be selected and adapted to fit into various online business. They are based on the domain where these algorithms would be applied. The question is: *What are the most important features required in online retailer systems?* Systems may need to increase the coverage of the business items catalogue at the expense of accuracy prediction at certain stages of business development, then demand may be more focused on accurate predictions at later stages. Another example is that some recommenders aim to allocate similar users because the number of users is remarkably smaller than the number of items. In this case, the approach H-ISTS$_{\mathrm{user}}$ may be the best to be applied as more coverage of items is achieved and novel items will therefore be added to users' profiles. However, for systems with a larger number of users compared to items, then H-ISTS$_{\mathrm{item}}$ may fit better as high accuracy and very similar items may be required in this case.

6.8 Chapter Summary

In this chapter, we modelled and evaluated the performance of an integrated of of H-ISTS approach with a collaborative filtering recommender, H-ISTS$_{\mathrm{CF}}$. The chapter has started by giving a background about memory-based collaborative filtering recommenders, including a description of user-based and item-based methods . These methods were then used to build the approaches H-ISTS$_{\mathrm{user}}$ and H-ISTS$_{\mathrm{item}}$. The goals to be satisfied by the study were then highlighted. The main goal was to provide personalised recommendations to existing users with profiles of ratings using the novel approach H-ISTS. Two algorithms were proposed to implement the user-based and the item-based models. The performance evaluation include accuracy and the non-accuracy metrics, including the diversity, which represent challenging issues in the recommendation environment. Coverage of the user-item matrix was evaluated to compare the effect of predicted items on the user-item ratings matrix. Experiments were conducted to evaluate the performance of the H-ISTS$_{\mathrm{CF}}$ models. The results were discussed under different metrics: MAE for accuracy, a coverage and the ILS metric for diversity. The chapter was concluded by highlighting the most interesting findings from the experiments results.

Chapter 7

Conclusions and Future Work

The aim of this thesis is to use online social networks (OSNs), and in particular microblog services as another source of trust and sentiment data in order to improve personalised recommendation approaches. In this chapter, we discuss the contributions and the findings of the thesis by answering the question: *what did we learn from this thesis?*

7.1 Thesis Contributions

The thesis has led to the following novel contributions which are organized as the answers to the research questions in Section 1.2.

1. Based on a thorough investigation of the literature in the domain of recommendations, in Chapter 2, we have identified a gap in the current recommendation literature and discussed the overlooked use of data from online friends in social networks. It was obvious that information coming from users' online connections were hugely ignored in supporting recommender systems. A number of drawbacks were highlighted in the current trust-based and reviews-based recommenders. Therefore, a number of desired features in recommender systems were identified and discussed. These features are highly demanded when developing RSs:

    - F1: Preserving the accuracy level; this requirement is important because it is traditionally considered as the main metric to validate the performance of RSs.

- F2: Supporting user/item cold-start solution; because new users have not yet built their profiles, and therefore, it is challenging for most RSs to understand and find new users' preferences.
- F3: Addressing sparsity and increasing matrix coverage, this feature is included to satisfy online business needs to allow most of the catalogue items to be seen and accessed by users.
- F4: Using rating information as the main data source; in spite of the need to adopt new data sources, rating information still remain a concrete element consistently needed when designing RSs.
- F5: Using sentiment analysis (SA) techniques to infer implicit opinions from product reviews to provide relevant recommendations to a given user. Recently, this feature has attracted interest in recommendation research to look deeper into written reviews for recommendation personalisation purposes.
- F6: Implementing trust-based approaches using web of trust (WOT) or implicit trust based on user systems.
- F7: Using OSNs as a new source of data, such as supporting the idea of using microbloggers on OSNs to augment recommendations.
- F8: Integrating people's intercommunications from OSNs for more accurate and diverse recommendations .

This thesis has provided a critical analysis of the current RS literature against the designed and these desired features.

2. The development of the tool *TIE* to make the required information available. It provided evidence that such opinions and sentiment information are available and extractable. This provides an answer to the first research question: *(1) How can we derive recommendations from OSNs about particular items?*

3. The design, implementation and the evaluation of Implicit Social Trust and Sentiment Analysis (ISTS) approach. The novelty of ISTS is reflected in the following:

   - This approach does not require users to rate items and or write product reviews. This privilege gives the ISTS method the capability to work with the minimum information provided by users.

- It links between trust and sentiment analysis and uses them in one integrated framework, although these two elements are managed separately in the literature.

The ISTS has a number of interesting features. First, it supports social trust in a different way from the current trust-based recommenders as explained in Section 4.3.2. Secondly, the approach integrates the principles of sentiment analysis techniques to reflect friends' opinions in a very representative way, as detailed in Section 4.4.2. Finally, ISTS has been built upon two types of machine learning approaches. First, classification techniques which are used to estimate the ratings for unknown items in ratings categories. Second, because of the aim to represent unknown ratings in a very precise way, regression algorithms were introduced for the rating estimation task. Support vector regression (SVR) performed the best in this task producing the most accurate rating prediction. Despite the success of ISTS in achieving significant performance results, some limitations were identified, which required to be resolved and addressed. These limitations are described in Section 4.8, which were the basis for building the H-ISTS approach.

4. The design, implementation and evaluation of the Hybrid Implicit Social Trust and Sentiment Analysis (H-ISTS) approach is detailed in Chapter 5. This approach provided solutions to overcome the limitations identified in the ISTS. It mainly enhanced the social trust representation, as it is important to explore which trust components have the highest impact on trust values. To do so, genetic algorithm (GA) was used to optimise trust parameters. For achieving the goal of more sophisticated social the trust representations, we increased the number of trust features to include elements, such as Bio information, the Name and the Image, in addition to the features already used in the ISTS system, re-tweets (RT) and followers/followings (L). In order to verify this, we have prepared the a new dataset for the new social trust requirements. By doing this, we have been able to understand and compare the different importance features among the more fine-grained new social trust components.

   RT and *L* were proven to be among the important trust features to indicate the strength of trust between an active user and a friend. In the survey results, feature *Name* was at the top of the ranked list, however, in our model it is given a weight less than *RT* and *L*. The *Bio* feature was considered to be in the first position of importance in survey results, and based on our model results, the lowest weight that affects the trust was given to the *Bio* feature. Generally, this may happen because users, on OSNs, prejudge friends' accounts from the *Name* and the *Bio* information apparently, while *RT* and *L* may require more efforts to identify them.

5. From contributions 3 and 4, we have answered the second research Question: (2) *How can the trust model be built to select friends?* the third research Question: (3) *What is the best method to derive the correct sentiment that describes friends' opinions from their posts?* And the fourth research Question: (4) *How to design an effective method to predict ratings for new users?*

6. Investigation of the performance by the integration of H-ISTS framework with a collaborative filtering (CF) recommender, H-ISTS$_{\mathrm{CF}}$. After integration, we have analysed the effect of users' OSN behaviour in a CF framework. In Chapter 6, we have designed several scenarios to validate by using the two datasets SD1 and SD2, with variable social trust parameters. Our approach was analysed in related to user-based and item-based techniques of CF and have investigated improvements achieved in coverage and diversity as well as accuracy in RSs outputs. In RSs, diversity problems present a challenge when there is a requirement to enrich users with new and interesting recommendations which are different from their usual pattern of ratings. People tend to be influenced significantly by opinions and tastes of friends, families and acquaintances. Consequently, it is important to use this link to provide users with interesting and unexpected items observed from their online connections. Different from recommendations from strangers -other users in RSs- users' online friends can expand and enrich users's profiles with new valuable selections. Interestingly, the thesis found out that the proposed approach H-ISTS$_{\mathrm{user}}$ performed well in terms of improving diversity and coverage, whereas H-ISTS$_{\mathrm{item}}$ offered the best performance in terms of accuracy and recommending a defined range of items that were close to the ones which already existing in a user's profile. Arriving to this conclusion, the thesis answered the fifth research Question: (5) *What will the collaborative filtering (CF) performance will be when we fuse ordinary users' preferences with friends' preferences derived from OSNs?* and the sixth research Question: (6) *How can the new approach contribute to the diversity challenge in CF users' profiles?*.

7.2 Future Work

We propose the following interesting areas for further future work.

- The novel approaches ISTS, H-ISTS and H-ISTS$_{\text{CF}}$ have provided solutions to predict items ratings for different users and therefore, improvement in users' profiles have been investigated. However, this thesis did not explore the augmentation of items profiles. Items can be linked with their OSNs connections. and RSs can establish accounts on OSNs and track followers of item and study the resultant behaviour. In this way, RSs can suggest preliminary rating estimations for new items based on followers' posts. In spite of the fact that this method is not tailored to fit a particular user's needs, it still can boots new items by rating prediction from an item reputation point of view.

- The design of ISTS and H-ISTS approaches has integrated with memory-based techniques. We intend to further extend our approach by designing social matrix factorisation (MF) models to be built upon the concept of the proposed approach used in this thesis. It is known in the literature that these models enhance the scalability of recommenders, and hence, we can by, extension, increase the recommender quality. For example, friends' opinions can be used as the regularisation parameters in MF models. However, such improvements may achieve potential success if the models are designed to avoid complexity and expensive computational cost associated with building MF models.

- The approach in this work can be used in a new RS directions, including the explanation recommender systems. Few works have been carried out in this direction. This work can be developed to link to explanation RSs. Users can be provided with justifications about items selections. For example, for a recommended item that has given a high positive rating, the system should have details to justify the rating; such as high rating is explained by including friends' opinions. Future studies can implement RSs which have the feature of interacting with users and the facility to tune and refine options coming from ONSs. For example, these systems may give users the ability to exclude a certain friend from the recommendation process.

- Profile privacy attacks are a threat faced by RSs. The recommended items in our approach might be less prone to profiles attacks, as they are drawn in a very personalised manner using information from trusted friends. Because the resulting recommendations reflect the trusted friends' opinions about items, the possibility of profile attacks in these recommender systems can be low. More evaluations can be applied to measure the degree of protection that the proposed approaches will present under several scenarios. Future studies can investigate the privacy and protection of profiles from attacks when considering the proposed approaches in this thesis.

- In ISTS and H-ISTS approaches, we considered the direct connection of friends to users. In other words, only friends appearing in followings and followers lists are considered. A set of experiments can be conducted to investigate a longer propagated path to include friends of friends. An improved version of these approaches can assess the friends' opinions, including to one further friend in the social connection path. This may be useful as studies based on WOT showed that shorter paths yield better recommendations. We should conduct experiments to demonstrate whether such approaches provide better results in longer or shorter paths. These experiments can shed more light on how trust can be affected by the path length in microblogs.

- Another area that can be suggested for future research is extending this study to include other microblog services. As mentioned before, our targeted OSN is Twitter. It may be of interest to know whether other microblogger platforms such as Facebook, can produce similar or better results. Most of the novel components in our approaches are available on other microbloggers which makes the ISTS and H-ISTS models applicable and easily generalisable to other services. However, experiments should be maintained under these microbloggers' conditions.

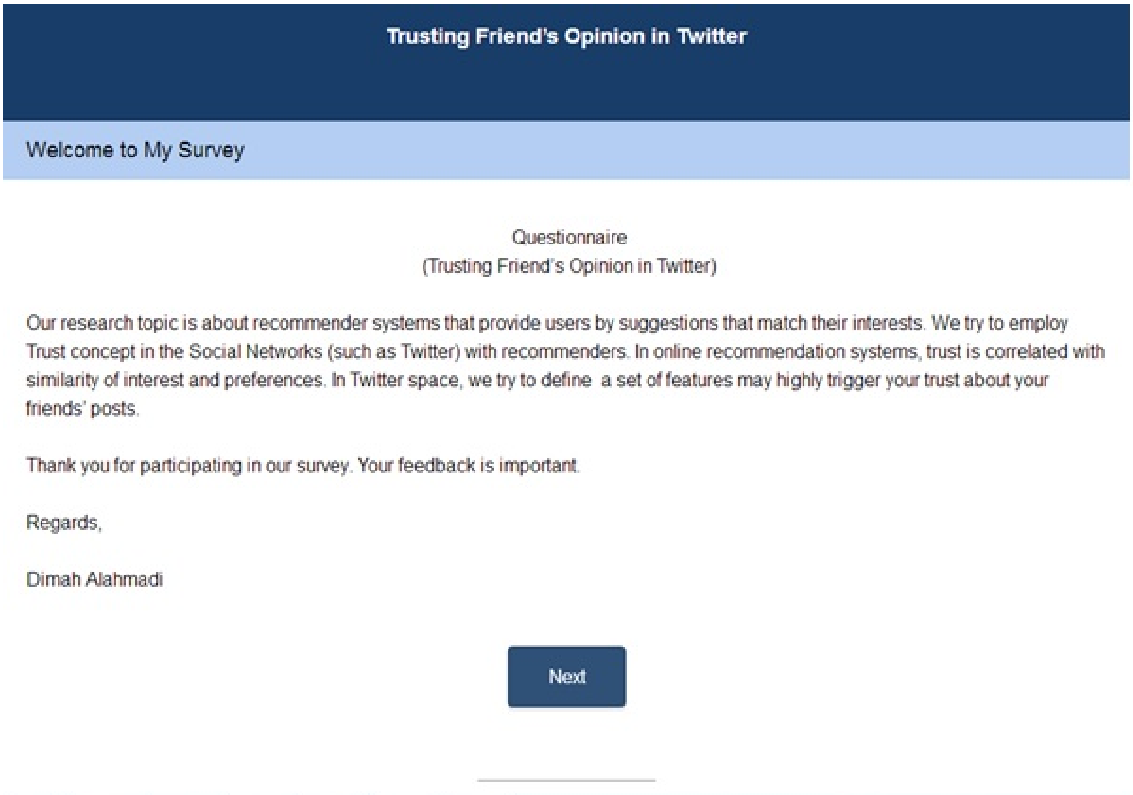
- It is clear that the findings in this thesis demonstrated the utility of the modern web environment. Therefore, the approach in this thesis can be extended to the Semantic Web domain. Future studies can build ontologies of preferences, which can be seen as a conceptualisation process that presents a logical description for shared data [139]. For example, users may visit different websites for different purposes, such as purchasing items and sharing news and opinions, and ontolgies can intelligently understand and represent users' tastes across domains and applications and can capture their general preferences in different platforms. Therefore, these ontologies can guide users based on the ISTS and

H-ISTS frameworks as a building block to the adopted approach.

Appendix A

Trust Features Survey

We asked 50 Twitter users about the possible trust features using the online survey tool *www.surveymonkey.co.uk*. Emails were sent to users with the link to the survey. The users were varied from students in Manchester University and their friends on Twitter. Responses arrived to researcher via the online tool. More than half of the participants were 22 to 37 aged people who use Twitter several times per day and about 70% of participants are female. Participants were asked to answer specific questions in order to indicate the importance of the different features they considered when assessing trust in friends.



Trusting Friend's Opinion in Twitter

Welcome to My Survey

Questionnaire
(Trusting Friend's Opinion in Twitter)

Our research topic is about recommender systems that provide users by suggestions that match their interests. We try to employ Trust concept in the Social Networks (such as Twitter) with recommenders. In online recommendation systems, trust is correlated with similarity of interest and preferences. In Twitter space, we try to define a set of features may highly trigger your trust about your friends' posts.

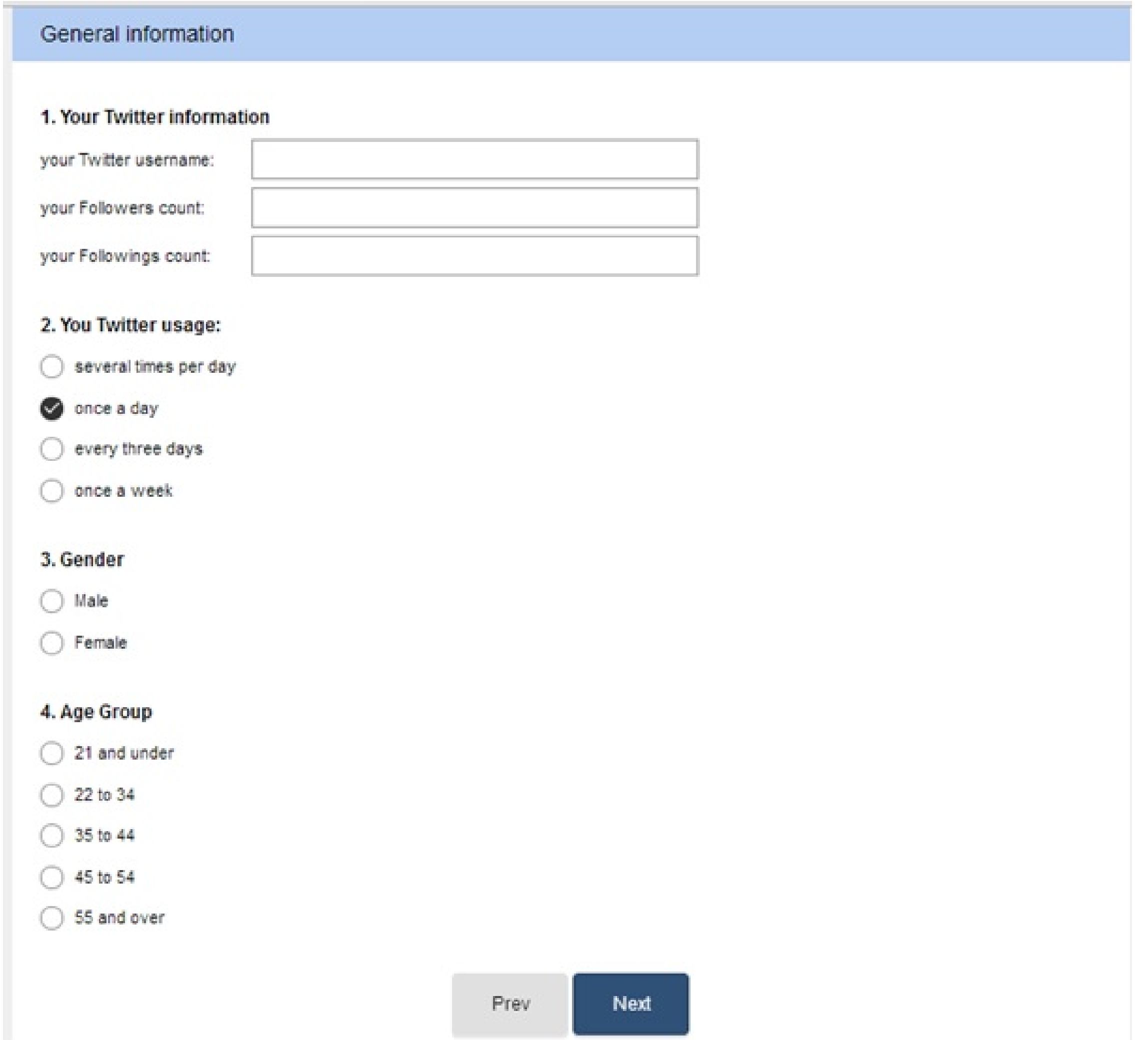
Thank you for participating in our survey. Your feedback is important.

Regards,

Dimah Alahmadi

Next

Figure A.1: Page1



General information

1. Your Twitter information

your Twitter username:

your Followers count:

your Followings count:

2. You Twitter usage:

- ○ several times per day
- ● once a day
- ○ every three days
- ○ once a week

3. Gender

- ○ Male
- ○ Female

4. Age Group

- ○ 21 and under
- ○ 22 to 34
- ○ 35 to 44
- ○ 45 to 54
- ○ 55 and over

Prev Next

Figure A.2: Page2

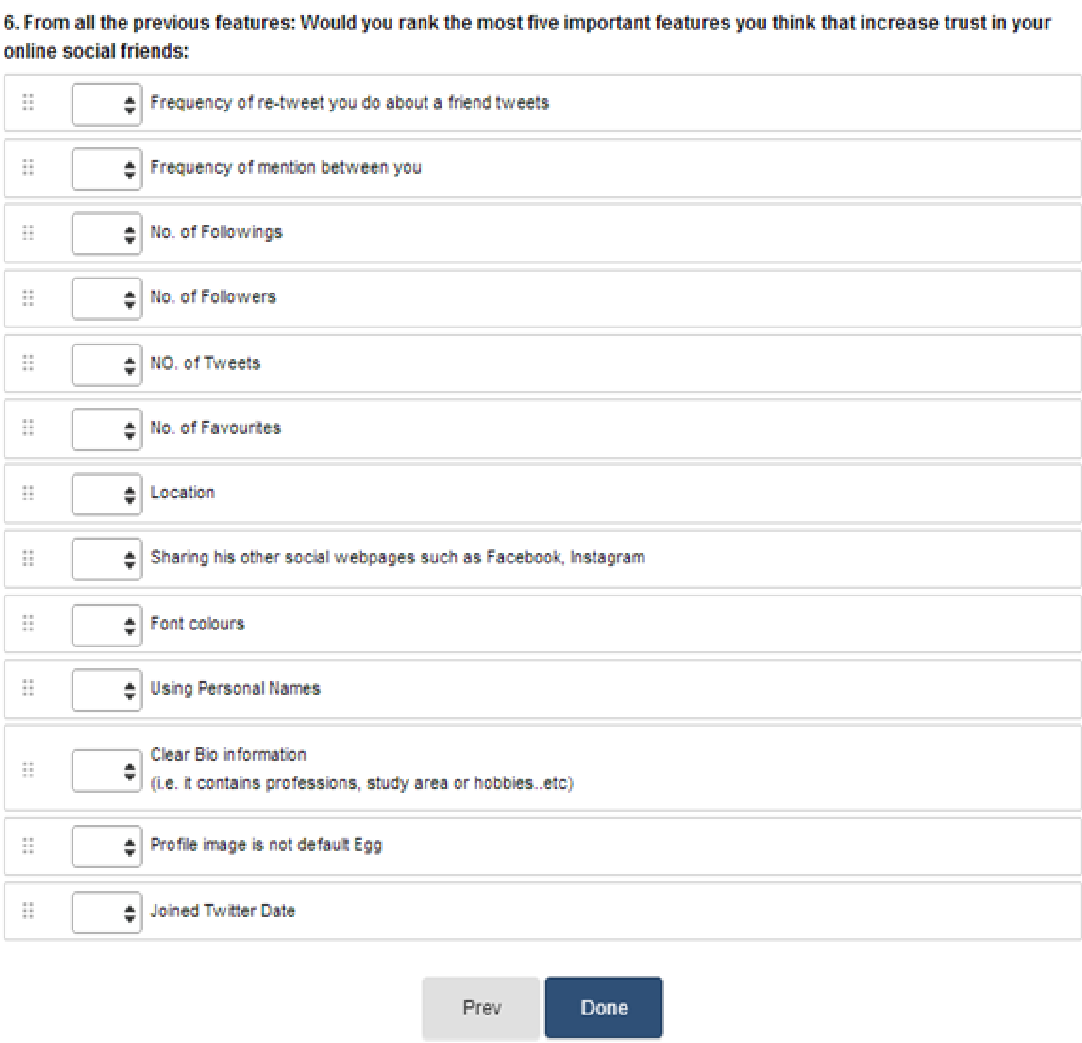
Assessing the trust in friends in Twitter

In this section, participants will be asked to answer questions to indicate the importance of different features on assessing the trust in friends.

5. Which kind of features in Friends' accounts may influence your trust and believing in their tweets that contain opinion about product such as movie, book, game, restaurants...etc?

	Least Important(1)	Not Important(2)	Neutral(3)	Important(4)	Extremely Important(5)
1.Frequency of re-tweet you do about a friend tweets	○	○	○	○	○
2.Frequency of mention between you	○	○	○	○	○
3.No. of Followings	○	○	○	○	○
4.No. of Followers	○	○	○	○	○
5.NO. of Tweets	○	○	○	○	○
6.No. of Favourites	○	○	○	○	○
7.Location	○	○	○	○	○
8.Sharing his other social webpages such as Facebook, Instagram	○	○	○	○	○
9.Font colours	○	○	○	○	○
10.Using Personal Names	○	○	○	○	○
11.Clear Bio information (i.e. it contains professions, study area or hobbies..etc)	○	○	○	○	○
12.Profile image is not default Egg	○	○	○	○	○
13.Joined Twitter Date	○	○	○	○	○

Figure A.3: Page3


6. From all the previous features: Would you rank the most five important features you think that increase trust in your online social friends:

- Frequency of re-tweet you do about a friend tweets
- Frequency of mention between you
- No. of Followings
- No. of Followers
- NO. of Tweets
- No. of Favourites
- Location
- Sharing his other social webpages such as Facebook, Instagram
- Font colours
- Using Personal Names
- Clear Bio information (i.e. it contains professions, study area or hobbies..etc)
- Profile image is not default Egg
- Joined Twitter Date

Prev | Done

Figure A.4: Page4

Appendix B

Scenarios

Getting Recommendation from Friends in Twitter

Information and introductions:

- Read the information about re-tweet actions and followers/followings in percentage rather than numbers to make it easy for annotators.
- Picture profile: it is the image appears next to the user in Twitter not the default egg.
- Bio Information: is the text that the users write on their twitter page. It is considered clear bio when the user describes hobbies, professions, interests, etc., otherwise it is unclear bio.
- Personal name: indicates whether the user uses a real personal name, for example *Peter*, and it is not such as *Coool:)*.
- Your answer is considered as a rating between [1-5].

Scenarios are:

1-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets three times out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 40% followers and 60%following. This friend x, posted a tweet about a movie and an opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

2-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his 1 re-tweet actions out of 6 re-tweets youve done recently. This friend x has percentages of total relations as 39% followers and 61%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

3-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 35% followers and 75%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

4-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 54% followers and 46%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

5-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

6-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

7-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of two re-tweets actions youve done recently. This friend x has percentages of total relations as 56% followers and 44%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

8-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two times out of six re-tweets actions youve done recently. This friend x has percentages of total relations as 51% followers and 49%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

9-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 51% followers and 49%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

10-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

11-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 40% followers and 60%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is : 12-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 re-tweets actions youve done recently. This friend x has percentages of total relations as 59% followers and 41%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

13-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

14-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of 13 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is: 15-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 12 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

16-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two times out of 8 re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

17-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

18-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one time out of ten re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

19-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

20-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

21-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

22-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets two out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

23-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets two out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

24-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 70% followers and 30%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

25-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one time out of only one re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

26-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 46% followers and 54%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

27-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 3 out of 12 re-tweets actions youve done recently. This friend x has percentages of total relations as 46% followers and 54%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

28-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 46% followers and 54%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information How are you going to rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

29-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 78% followers and 22%following. This friend x,

posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

30-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 78% followers and 22%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

31-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 3 out of 17 re-tweets actions youve done recently. This friend x has percentages of total relations as 78% followers and 22%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

32-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 78% followers and 22%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:
33-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets zero re-tweets actions youve done recently. This friend x has percentages of total relations as 85 This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

34-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 56% followers and 44%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is: 35-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 56 This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

36-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his zero re-tweets actions youve done recently. This friend x has percentages of total relations as 22% followers and 78%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

37-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 15 re-tweets actions youve done recently. This friend x has percentages of total relations as 52% followers and 48%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

38-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 19 re-tweets actions youve done recently. This friend x has percentages of total relations as 52% followers and 48%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information How are you going to rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

39-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 16 re-tweets actions youve done recently. This friend x has percentages of total relations as 54% followers and 45%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4.1 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information How are you going to rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

40-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets one out of 11 re-tweets actions youve done recently. This friend x has percentages of total relations as 54% followers and 45%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

41-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 17 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

42-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 4 out of 14 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

43-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x,

posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

44-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 3 out of 17 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

45-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets two out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

46-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 75% followers and 25%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

47-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets one out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 75% followers and 25%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

48-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

49-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets two out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

50-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 8 re-tweets actions youve done recently. This friend x has percentages of total relations as 28% followers and 72%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

51-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets one out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 53%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

52-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 7 re-tweets actions youve done recently. This friend x has percentages of total relations as 53% followers and 47%following. This friend x,

posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

53-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 57 This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

54-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 57 This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information How are you going to rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

55-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 11 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

56-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets one out of 11 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

57-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of only one re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

58-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

59-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

60-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 17 re-tweets actions youve done recently. This friend x has percentages of total relations as 39% followers and 61%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

61-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 17 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

62-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of only one re-tweets actions youve done recently. This friend x has percentages of total relations as 30% followers and 70%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

63-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets zero re-tweets actions youve done recently. This friend x has percentages of total relations as 53% followers and 47%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

64-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 36% followers and 64%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information How are you going to rate this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

65-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets one out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

66-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets two out of 19 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

67-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 20 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

68-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 14 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

69-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

70-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 4 out of 9 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is: 71-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets 3 out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

72-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets one out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 47% followers and 53%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

73-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 2 out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 90% followers and 10%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

74-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 90% followers and 10%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

75-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 90% followers and 10%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

76-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 out of 12 re-tweets actions youve done recently. This friend x has percentages of total relations as 90 This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

77-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 2 out of 16 re-tweets actions youve done recently. This friend x has percentages of total relations as 90 This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

78-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 out of 14 re-tweets actions youve done recently. This friend x has percentages of total relations as 85% followers and 15%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

79-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 out of 20 re-tweets actions youve done recently. This friend x has percentages of total relations as 54 This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

80-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets 1 out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 54% followers and 46%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

81-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 out of 1 re-tweets actions youve done recently. This friend x has percentages of total relations as 65% followers and 35%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

82-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 65% followers and 35%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

83-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 out of 20 re-tweets actions youve done recently. This friend x has percentages of total relations as 65% followers and 35%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

84-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 time out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 47 This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

85-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 time out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 52% followers and 48%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 4.45 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

86-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 2 times out of nine re-tweets actions youve done recently. This friend x has percentages of total relations as 95% followers and 5%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

87-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 95% followers and 5%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

88-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 2 times out of 8 re-tweets actions youve done recently. This friend x has percentages of total relations as 95% followers and 5%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

89-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 19 re-tweets actions youve done recently. This friend x has percentages of total relations as 95% followers and 5%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3.1 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

90-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

91-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 8 re-tweets actions youve done recently. This friend x has percentages of total relations as 57% followers and 43%following. This friend x, posted a tweet about a (book, movie, mobileetc) and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

92-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets 1 times out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 57 This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

93-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 times out of 2 re-tweets actions youve done recently. This friend x has percentages of total relations as 68% followers and 32%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

94-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 times out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 63% followers and 37%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:
95-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets 1 times out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 55% followers and 45%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

96-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 55% followers and 45%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

97-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 2 times out of 8 re-tweets actions youve done recently. This friend x has percentages of total relations as 55% followers and 45%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear

Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

98-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 13 re-tweets actions youve done recently. This friend x has percentages of total relations as 29% followers and 71%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

99-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 53% followers and 47%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

100-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 16 re-tweets actions youve done recently. This friend x has percentages of total relations as 66% followers and 34%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

101-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 66% followers and 34%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

102-Consider the following scenario in Twitter: Your have one friend x : you re-tweeted his tweets 0 times out of 11 re-tweets actions youve done recently. This friend x has percentages of total relations as 60% followers and 40%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a NOT clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

103-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 13 re-tweets actions youve done recently. This friend x has percentages of total relations as 96% followers and 4%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

104-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 20 re-tweets actions youve done recently. This friend x has percentages of total relations as 96% followers and 4%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

105-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 96% followers and 4%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is :

106-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 19 re-tweets actions youve done recently. This friend x has percentages of total relations as 96% followers and 4%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

107-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 1 times out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 11 This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

108-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 11% followers and 89%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

109-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 4 re-tweets actions youve done recently. This friend x has percentages of total relations as 14% followers and 86%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

110-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 14 re-tweets actions youve done recently. This friend x has percentages of total relations as 47% followers and 53%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:
111-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 10 re-tweets actions youve done recently. This friend x has percentages of total relations as 25% followers and 75%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 4 star out of five. He/she uses a picture profile He/she uses personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

112-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 times out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 52% followers and 48%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

113-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 0 times out of 6 re-tweets actions youve done recently. This friend x has percentages of total relations as 44% followers and 56%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

114-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 2 times out of 13 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

115-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

116-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 3 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is :

117-Consider the following scenario in Twitter: You have one friend x : you re-tweeted his tweets 1 times out of 5 re-tweets actions youve done recently. This friend x has percentages of total relations as 50% followers and 50%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses NOT personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

118-Consider the following scenario in Twitter: Your have one friend x: you re-tweeted his tweets 0 times out of 14re-tweets actions youve done recently. This friend x has percentages of total relations as 54% followers and 46%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 3 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

119-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 2 times out of 19 re-tweets actions youve done recently. This friend x has percentages of total relations as 1% followers and 99%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

120-Consider the following scenario in Twitter: You have one friend x: you re-tweeted his tweets 0 times out of 20 re-tweets actions youve done recently. This friend x has percentages of total relations as 1% followers and 99%following. This friend x, posted a tweet about a movie and his opinion about it was extracted as 5 star out of five. He/she uses a picture profile He/she uses personal name He/she uses NOT a clear Bio information What is your opinion about this item based on the previous scenario between [1-5]=[awful-excellent] you can choose number such as 4.5. your answer is:

Bibliography

[1] S. Alag, *Collective intelligence in action.* Manning New York, 2009.

[2] P. Victor, M. De Cock, and C. Cornelis, "Trust and recommendations," in *Recommender Systems Handbook*, pp. 645–675, Springer, 2011.

[3] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pp. 203–210, ACM, 2009.

[4] J. Lin, K. Sugiyama, M.-Y. Kan, and T.-S. Chua, "Addressing cold-start in app recommendation: Latent user models constructed from twitter followers," in *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pp. 283–292, ACM, 2013.

[5] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[6] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," *Recommender Systems Handbook*, pp. 1–35, 2011.

[7] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pp. 175–186, ACM, 1994.

[8] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001.

[9] M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine learning*, vol. 27, no. 3, pp. 313–331, 1997.

[10] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, vol. 20, p. 704, 2003.

[11] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 287–296, ACM, 2011.

[12] Y. Kim and R. Phalak, "A trust prediction framework in rating-based experience sharing social networks without a web of trust," *Information Sciences*, 2012.

[13] C. Birtolo and D. Ronca, "Advances in clustering collaborative filtering by means of fuzzy c-means and trust," *Expert Systems with Applications*, vol. 40, no. 17, pp. 6997–7009, 2013.

[14] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pp. 79–86, Association for Computational Linguistics, 2002.

[15] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," in *12th International Workshop on the Web and Databases*, 2009.

[16] C. W.-K. Leung, S. C.-F. Chan, F.-L. Chung, and G. Ngai, "A probabilistic rating inference framework for mining user preferences from reviews," *World Wide Web*, vol. 14, no. 2, pp. 187–215, 2011.

[17] X. Zhou, Y. Xu, Y. Li, A. Josang, and C. Cox, "The state-of-the-art in personalized recommender systems for social networking," *Artificial Intelligence Review*, vol. 37, no. 2, pp. 119–132, 2012.

[18] S. Rosenthal, P. Nakov, S. Kiritchenko, S. M. Mohammad, A. Ritter, and V. Stoyanov, "Semeval-2015 task 10: Sentiment analysis in twitter," in *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval*, 2015.

[19] U. Shardanand and P. Maes, "Social information filtering: algorithms for automating word of mouth," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 210–217, ACM Press/Addison-Wesley Publishing Co., 1995.

[20] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *Proceedings of the 16th international conference on World Wide Web*, pp. 271–280, ACM, 2007.

[21] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[22] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[23] R. Baeza-Yates, B. Ribeiro-Neto, *et al.*, *Modern information retrieval*, vol. 463. ACM press New York, 1999.

[24] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the fifth ACM conference on Digital libraries*, pp. 195–204, ACM, 2000.

[25] M. Steinbach, G. Karypis, V. Kumar, *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, vol. 400, pp. 525–526, Boston, 2000.

[26] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*, pp. 119–159, Springer, 2015.

[27] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[28] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, 1997.

[29] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 194–201, ACM Press/Addison-Wesley Publishing Co., 1995.

[30] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in *Proceedings of the fifth international conference on computer and information technology*, vol. 1, 2002.

[31] J. S. Breese, D. Heckerman, and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 43–52, Morgan Kaufmann Publishers Inc., 1998.

[32] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, p. 4, 2009.

[33] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, vol. 178, no. 1, pp. 37–51, 2008.

[34] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-based systems*, vol. 24, no. 8, pp. 1310–1316, 2011.

[35] K. Miyahara and M. J. Pazzani, "Improvement of collaborative filtering with the simple bayesian classifier," *IPSJ Journal*, vol. 43, no. 11, pp. 3429–3437, 2002.

[36] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms," in *Tools with Artificial Intelligence, 2006. ICTAI'06. 18th IEEE International Conference on*, pp. 497–504, IEEE, 2006.

[37] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, and Z. Chen, "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 114–121, ACM, 2005.

[38] S. Vucetic and Z. Obradovic, "Collaborative filtering using a regression-based approach," *Knowledge and Information Systems*, vol. 7, no. 1, pp. 1–22, 2005.

[39] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in neural information processing systems*, pp. 1329–1336, 2004.

[40] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*, pp. 713–719, ACM, 2005.

[41] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Investigation of various matrix factorization methods for large recommender systems," in *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pp. 553–562, IEEE, 2008.

[42] R. M. Bell and Y. Koren, "Lessons from the netflix prize challenge," *ACM SIGKDD Explorations Newsletter*, vol. 9, no. 2, pp. 75–79, 2007.

[43] J. Bennett and S. Lanning, "The netflix prize," in *Proceedings of KDD cup and workshop*, vol. 2007, p. 35, 2007.

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

[45] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[46] T. George and S. Merugu, "A scalable collaborative filtering framework based on co-clustering," in *Data Mining, Fifth IEEE International Conference on*, pp. 4–pp, IEEE, 2005.

[47] M. C. Puerta Melguizo, L. Boves, A. Deshpande, and O. M. Ramos, "A proactive recommendation system for writing: Helping without disrupting," in *Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!*, pp. 89–95, ACM, 2007.

[48] P. Han, B. Xie, F. Yang, and R. Shen, "A scalable p2p recommender system based on distributed collaborative filtering," *Expert systems with applications*, vol. 27, no. 2, pp. 203–210, 2004.

[49] H.-W. Tung and V.-W. Soo, "A personalized restaurant recommender agent for mobile e-service," in *e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on*, pp. 259–262, IEEE, 2004.

[50] O. Averjanova, F. Ricci, and Q. N. Nguyen, "Map-based interaction with a conversational mobile recommender system," in *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on*, pp. 212–218, IEEE, 2008.

[51] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.

[52] P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Proceedings of the National Conference on Artificial Intelligence*, pp. 187–192, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002.

[53] L. B. Marinho, A. Nanopoulos, L. Schmidt-Thieme, R. Jäschke, A. Hotho, G. Stumme, and P. Symeonidis, "Social tagging recommender systems," in *Recommender systems handbook*, pp. 615–644, Springer, 2011.

[54] C.-L. Huang, P.-H. Yeh, C.-W. Lin, and D.-C. Wu, "Utilizing user tag-based interests in recommender systems for social resource sharing websites," *Knowledge-Based Systems*, vol. 56, pp. 86–96, 2014.

[55] G. Guo, "Integrating trust and similarity to ameliorate the data sparsity and cold start for recommender systems," in *Proceedings of the 7th ACM conference on Recommender systems*, pp. 451–454, ACM, 2013.

[56] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*, pp. 640–651, ACM, 2003.

[57] D. Artz and Y. Gil, "A survey of trust in computer science and the semantic web," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5, no. 2, pp. 58–71, 2007.

[58] J. A. Golbeck, "Computing and applying trust in web-based social networks," 2005.

[59] P. Massa and P. Avesani, “Trust-aware recommender systems,” in *Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17–24, ACM, 2007.

[60] M. Richardson, R. Agrawal, and P. Domingos, “Trust management for the semantic web,” in *The Semantic Web-ISWC 2003*, pp. 351–368, Springer, 2003.

[61] C.-N. Ziegler and G. Lausen, “Propagation models for trust and distrust in social networks,” *Information Systems Frontiers*, vol. 7, no. 4-5, pp. 337–358, 2005.

[62] P. Cofta, “Distrust,” in *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet*, pp. 250–258, ACM, 2006.

[63] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins, “Propagation of trust and distrust,” in *Proceedings of the 13th international conference on World Wide Web*, pp. 403–412, ACM, 2004.

[64] P. Victor, C. Cornelis, M. De Cock, and P. Pinheiro da Silva, “Gradual trust and distrust in recommender systems,” *Fuzzy Sets and Systems*, vol. 160, no. 10, pp. 1367–1382, 2009.

[65] D. Zejda, “From subjective trust to objective trustworthiness in on-line social networks: Overview and challenges,” *Journal of Systems Integration*, vol. 1, no. 1-2, pp. 16–22, 2010.

[66] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, “Reputation systems,” *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[67] L. Page, S. Brin, R. Motwani, and T. Winograd, “The pagerank citation ranking: bringing order to the web.,” 1999.

[68] S. Y. Rieh and D. R. Danielson, “Credibility: A multidisciplinary framework,” *Annual review of information science and technology*, vol. 41, no. 1, pp. 307–364, 2007.

[69] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “Providing justifications in recommender systems,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 6, pp. 1262–1272, 2008.

[70] P. Avesani, P. Massa, and R. Tiella, "A trust-enhanced recommender system application: Moleskiing," in *Proceedings of the 2005 ACM symposium on Applied computing*, pp. 1589–1593, ACM, 2005.

[71] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174, ACM, 2005.

[72] J. Golbeck and J. Hendler, "Filmtrust: Movie recommendations using trust in web-based social networks," in *Proceedings of the IEEE Consumer communications and networking conference*, vol. 96, Citeseer, 2006.

[73] J. Golbeck, "Generating predictive movie recommendations from trust in social networks," *Trust Management*, pp. 93–104, 2006.

[74] P. Massa and P. Avesani, "Trust-aware collaborative filtering for recommender systems," *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pp. 492–508, 2004.

[75] K. K. Bharadwaj and M. Y. H. Al-Shamri, "Fuzzy computational models for trust and reputation systems," *Electronic Commerce Research and Applications*, vol. 8, no. 1, pp. 37–47, 2009.

[76] V. Kant and K. K. Bharadwaj, "Incorporating fuzzy trust in collaborative filtering based recommender systems," in *Swarm, Evolutionary, and Memetic Computing*, pp. 433–440, Springer, 2011.

[77] T. Bhuiyan, "A survey on the relationship between trust and interest similarity in online social networks," *Journal of Emerging Technologies in Web Intelligence*, vol. 2, no. 4, pp. 291–299, 2010.

[78] J. O'DONOVAN and B. Smyth, "Mining trust values from recommendation errors," *International Journal on Artificial Intelligence Tools*, vol. 15, no. 06, pp. 945–962, 2006.

[79] P. Victor, *Trust networks for recommender systems*. PhD thesis, Ghent University, 2010.

[80] P. Victor, C. Cornelis, M. De Cock, and A. M. Teredesai, "Key figure impact in trust-enhanced recommender systems," *AI Communications*, vol. 21, no. 2, pp. 127–143, 2008.

[81] O. Arazy, I. Elsane, B. Shapira, and N. Kumar, "Social relationships in recommender systems," in *Proc. of the 17th Workshop on Information Technologies & Systems*, 2007.

[82] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and trends in information retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[83] F. Peleja, P. Dias, and J. Magalhães, "A regularized recommendation algorithm with probabilistic sentiment-ratings," in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pp. 701–708, IEEE, 2012.

[84] S. R. Das and M. Y. Chen, "Yahoo! for amazon: Sentiment extraction from small talk on the web," *Management Science*, vol. 53, no. 9, pp. 1375–1388, 2007.

[85] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in *Proceedings of the 12th international conference on World Wide Web*, pp. 519–528, ACM, 2003.

[86] P. A. Pavlou and A. Dimoka, "The nature and role of feedback text comments in online marketplaces: Implications for trust building, price premiums, and seller differentiation," *Information Systems Research*, vol. 17, no. 4, pp. 392–414, 2006.

[87] A. Ghose and P. G. Ipeirotis, "Designing novel review ranking systems: predicting the usefulness and impact of reviews," in *Proceedings of the ninth international conference on Electronic commerce*, pp. 303–310, ACM, 2007.

[88] Š. Pero and T. Horváth, "Opinion-driven matrix factorization for rating prediction," in *User Modeling, Adaptation, and Personalization*, pp. 1–13, Springer, 2013.

[89] E. Najmi, K. Hashmi, Z. Malik, A. Rezgui, and H. U. Khan, "Capra: a comprehensive approach to product ranking using customer reviews," *Computing*, vol. 97, no. 8, pp. 843–867, 2015.

[90] F. Liu and H. J. Lee, "Use of social network information to enhance collaborative filtering performance," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4772–4778, 2010.

[91] D. Alahmadi and X.-J. Zeng, "Improving recommendation using trust and sentiment inference from osns," *International Journal of Knowledge Engineering*, vol. 1, no. 1, pp. 9–17, 2015.

[92] S. Garcia Esparza, M. P. O'Mahony, and B. Smyth, "On the real-time web as a source of recommendation knowledge," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 305–308, ACM, 2010.

[93] S. Garcia Esparza, M. P. OMahony, and B. Smyth, "Mining the real-time web: a novel approach to product recommendation," *Knowledge-Based Systems*, vol. 29, pp. 3–11, 2012.

[94] D. Maynard, K. Bontcheva, and D. Rout, "Challenges in developing opinion mining tools for social media," *Proceedings of@ NLP can u tag# usergeneratedcontent*, 2012.

[95] S. M. Kywe, E.-P. Lim, and F. Zhu, "A survey of recommender systems in twitter," in *Social Informatics*, pp. 420–433, Springer, 2012.

[96] B. Hajian and T. White, "Modelling influence in a social network: Metrics and evaluation," in *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pp. 497–500, IEEE, 2011.

[97] R. Sinha and K. Swearingen, "Comparing recommendations made by online systems and friends," in *Proceedings of the Delos-NSF workshop on personalization and recommender systems in digital libraries*, vol. 106, Dublin, Ireland, 2001.

[98] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 65–74, ACM, 2011.

[99] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, pp. 591–600, ACM, 2010.

[100] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring user influence in twitter: The million follower fallacy.," *ICWSM*, vol. 10, no. 10-17, p. 30, 2010.

[101] S.-M. Kim and E. Hovy, "Automatic identification of pro and con reasons in online reviews," in *Proceedings of the COLING/ACL on Main conference poster sessions*, pp. 483–490, Association for Computational Linguistics, 2006.

[102] V. Hatzivassiloglou and J. M. Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity," in *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pp. 299–305, Association for Computational Linguistics, 2000.

[103] F. Benamara, C. Cesarano, A. Picariello, D. R. Recupero, and V. S. Subrahmanian, "Sentiment analysis: Adjectives and adverbs are better than adjectives alone.," in *ICWSM*, 2007.

[104] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354, Association for Computational Linguistics, 2005.

[105] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *Proceedings of LREC*, vol. 6, pp. 417–422, 2006.

[106] M. Tavakolifard, K. C. Almeroth, and J. A. Gulla, "Does social contact matter?: modelling the hidden web of trust underlying twitter," in *Proceedings of the 22nd international conference on World Wide Web companion*, pp. 981–988, International World Wide Web Conferences Steering Committee, 2013.

[107] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[108] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, "Data mining methods for recommender systems," in *Recommender Systems Handbook*, pp. 39–71, Springer, 2011.

[109] S. Le Cessie and J. Van Houwelingen, "Ridge estimators in logistic regression," *Applied statistics*, pp. 191–201, 1992.

[110] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers-a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 4, pp. 476–487, 2005.

[111] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[112] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[113] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[114] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[115] A. J. Viera, J. M. Garrett, *et al.*, "Understanding interobserver agreement: the kappa statistic," *Fam Med*, vol. 37, no. 5, pp. 360–363, 2005.

[116] D. H. Alahmadi and X.-J. Zeng, "Ists: Implicit social trust and sentiment based approach to recommender systems," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8840–8849, 2015.

[117] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[118] M. A. Ghazanfar, *Robust, scalable, and practical algorithms for recommender systems*. PhD thesis, University of Southampton, 2012.

[119] D. H. Alahmadi and X.-J. Zeng, "Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis," in *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*, pp. 1045–1052, IEEE, 2015.

[120] P. Reed, B. Minsker, and D. E. Goldberg, "Designing a competent simple genetic algorithm for search and optimization," *Water Resources Research*, vol. 36, no. 12, pp. 3757–3761, 2000.

[121] K.-j. Kim and H. Ahn, "A recommender system using ga k-means clustering in an online shopping market," *Expert systems with applications*, vol. 34, no. 2, pp. 1200–1209, 2008.

[122] M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert systems with applications*, vol. 35, no. 3, pp. 1386–1399, 2008.

[123] A. Abbas, L. Zhang, and S. U. Khan, "A survey on context-aware recommender systems based on computational intelligence techniques," *Computing*, pp. 1–24, 2015.

[124] E. Lloret and M. Palomar, "Towards automatic tweet generation: A comparative study from the text summarization perspective in the journalism genre," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6624–6630, 2013.

[125] L. Aroyo and C. Welty, "Measuring crowd truth for medical relation extraction," in *2013 AAAI Fall Symposium Series*, 2013.

[126] L. Aroyo, C. Welty, *et al.*, "Truth is a lie: Crowd truth and the seven myths of human annotation," *AI Magazine*, vol. 36, no. 1, pp. 15–24, 2015.

[127] L. Aroyo and C. Welty, "Crowd truth: Harnessing disagreement in crowdsourcing a relation extraction gold standard," *WebSci2013. ACM*, vol. 2013, 2013.

[128] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[129] X. Ning, C. Desrosiers, and G. Karypis, "A comprehensive survey of neighborhood-based recommendation methods," in *Recommender Systems Handbook*, pp. 37–76, Springer, 2015.

[130] P. Castells, N. J. Hurley, and S. Vargas, "Novelty and diversity in recommender systems," in *Recommender Systems Handbook*, pp. 881–918, Springer, 2015.

[131] B. Smyth and P. McClave, "Similarity vs. diversity," in *Case-Based Reasoning Research and Development*, pp. 347–361, Springer, 2001.

[132] A. Gunawardana and G. Shani, "Evaluating recommender systems," in *Recommender Systems Handbook*, pp. 265–308, Springer, 2015.

[133] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, pp. 22–32, ACM, 2005.

[134] L. McAlister and E. Pessemier, “Variety seeking behavior: An interdisciplinary review,” *Journal of Consumer research*, pp. 311–322, 1982.

[135] C. H. Coombs and G. S. Avrunin, “Single-peaked functions and the theory of preference.,” *Psychological review*, vol. 84, no. 2, p. 216, 1977.

[136] M. Lubatkin and S. Chatterjee, “Extending modern portfolio theory into the domain of corporate diversification: does it apply?,” *Academy of Management Journal*, vol. 37, no. 1, pp. 109–136, 1994.

[137] K. Bradley and B. Smyth, “Improving recommendation diversity,” in *Proceedings of the Twelfth National Conference in Artificial Intelligence and Cognitive Science (AICS-01)*, pp. 75–84, Citeseer, 2001.

[138] M. Zhang and N. Hurley, “Avoiding monotony: improving the diversity of recommendation lists,” in *Proceedings of the 2008 ACM conference on Recommender systems*, pp. 123–130, ACM, 2008.

[139] T. R. Gruber *et al.*, “A translation approach to portable ontology specifications,” *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.