Improving Recommendation Using Trust and Sentiment Inference from OSNs

Dimah Alahmadi and Xaio-Jun Zeng

***Abstract*—Recommender systems (RSs) provide personalised suggestions of information or products relevant to users' needs. Although RSs have made substantial progresses in theory and algorithm development and have achieved many commercial successes, how to utilise the widely available information in Online Social Networks (OSNs) has been largely overlooked. Noticing such a gap in the existing research in RSs and taking into account a user's selection being greatly influenced by his/her trusted friends and their opinions, this paper proposes a framework of Implicit Social Trust and Sentiment (ISTS) based RSs, which improves the existing recommendation approaches by exploring a new source of data from friends' short posts in microbloggings as micro-reviews. The impact degree of friends' sentiment and level being trusted to a user's selection are identified by using machine learning methods including Naive Bayes, Logistic Regression and Decision Trees. As the verification of the proposed framework, experiments using real social data from Twitter microblogger are presented and results show the effectiveness and promising of the proposed approach.**

***Index Terms*—Recommender systems, machine learning, trust, sentiment analysis, microblogging.**

I. Introduction

The ultimate feature of Web 2.0 is the free generated content by users. Users have the ability to expose their opinions in Online Social Networks OSNs such as Facebook, Wikis and Twitter. Apparently, an exponential growth of information has become available to users. Consequently, two cases are imposed: one is the difficulty for users to find contents that are relevant to their own interest among vast amount of alternatives. The other is the demand to a modern technique which can provide personalised recommendations by exploiting information in the current environment of world wide web.

In fact, traditional Recommender Systems (RSs) play an important role in providing recommendations and they are deployed in the industry, such as Amazon, Netflex and Ebay. Typically, there are two main techniques of RSs: Collaborative Filtering recommenders (CF) and Content-Based recommenders (CB) [1], [2]. In CF techniques, recommenders can predict relevant items for an active user by utilising his previous history of ratings from similar users as neighbours [3] or similar items [4]. On the other hand, CB recommenders enrich recommendation by building user-item profiles based on the most important features of item contents [5], [6]. Usually, these systems depends on items containing textual information, for example, content of web pages. Recently, in the attempt of improving traditional RSs, some studies enhance the recommendation by including trust relations. Trust increases the ability of RSs to approach more trusted users and as a consequence more reliable products to be suggested [7]–[9]. Another methods proposed to empower recommendation by extracting the sentiment information from long reviews that users have written about products [10]–[12].

However, there are several limitations which still appear in the aforementioned approaches. First of all, the user-item rating matrix in CF techniques suffers from the well-known problem of sparsity as people tend to rate few items, and this produces less dense of the available ratings in RSs as pointed out in [4]. Further, the challenge of the cold-start problem when new users or items do not have any rating history. Traditional CF and CB recommenders assume the existence of sufficient amount of ratings or content information in order to generate powerful recommendation but this is not true in many cases [1]. Second, most of trust-enhanced RSs are not realistic as they do not reflect the influence of real social connections in providing recommendations. In real world, we tend to trust our friends' opinions about books, movies and restaurants. Third, recommendation extracted from users' product reviews do not employ any advantages which can be harnessed to personalise recommendations from friends in OSNs such as Twitter, since vast amount of information and opinions are available in such networks.

In this paper, we aim to propose a solution to the above problems and model a recommender which can involve a user's OSNs to draw the user's preferences even in the case that he/she dose not have any rating history. We propose Implicit Social Trust and Sentiment (ISTS) based RSs. It is based on the assumption that users tend to be influenced by their friends' opinions even if they have different interests. We argue that OSNs, microbloggings in particular, can be a rich source of knowledge to personalise recommendation. Our research interest is, for an active user, to harness his/her friends' sentiments about products by using the trust degree between the user and his/her friends in social network. We explore whether, for a user, his/her trusted friends' posts may be considered as short reviews to empower recommendation. To the best of our knowledge, this is the first work employing sentiment and trust from microbloggings to generate personalised recommendations. The contributions of this paper are:

Employing OSNs as a new source of data in order to use the short posts messages as micro-reviews in the proposed ISTS recommendation framework.

Manuscript received November 1, 2014; revised January 5, 2015.

The authors are with School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK (e-mail: dimah.alahmadi@postgrad.manchester.ac.uk, x.zeng@manchester.ac.uk).

DOI: 10.7763/IJKE.2015.V1.2

1) Inferring multiple score ratings from friends' posts in microbloggings by using sentiment analysis technique, as these posts are short and include informal use of language.
2) Using intercommunication between friends as the trust indicator to the importance of friends' opinion to a user.
3) Improving the prediction performance using different machine learning classification algorithms, in particular, for new users.

The rest of this paper is organised as follows. In Section II, we provide an overview of some major studies and approaches for recommender systems. In Section III, we explain the environment of social network data that we utilise. Section IV details the problem we study in this paper. Our approaches in predicting ratings are presented in Section V. The results of the experimental analysis are presented in Section VI, followed by the conclusion and future work in Section VII.

II. Related Work

In this section we highlight related important approaches: 1) traditional collaborative recommender, 2) trust-enhanced recommender, and 3) reviews-based recommender.

First, the traditional collaborative filtering approaches can be either memory-based or model-based. These methods are based on the rating history from users. In the memory-based methods, similarity computation is a primary element. They use a heuristic utility of similarity between users' vectors such as Pearson Correlation Coefficient (PCC) or cosine similarity measure (VCC) [1], [5], [13], [14]. On the other hand, the model-based methods employ machine learning models to predict product ratings [15], [16]. For example, Sarwar *et al.* [4], [17] implemented clustering algorithms to identify groups of customers who rated similar products and these clusters can be seen as like minded neighbors. Since k clusters are created, recommendation prediction can be computed by averaging the ratings in that cluster. Miyahara and Pazzani [5] proposed a RS based on Naive Bayes classifier and they only considered items which co-rated between users. They manipulated two classes: like and don't like and features are selected in a preprocessing step. Recent proposals focused in the accuracy of predictions such as matrix factorization for collaborative filtering. The approach proposed in [8] involved social connections data in providing recommendation by assigning social regularization terms in order to constraint matrix factorization objective function. They assumed that friends rate products and hence they used PCC and VCC to measure similarity as intermediate step.

Second, more studies have focused on trust-enhanced recommenders. Some studies applied trust by building trust net- work based on the assumption that users can obtain more accurate recommendation from people they trust [18], [19]. These kinds of methods used direct evaluations of trust from users. Golbeck *et al.* [18] propagated trust from trust network so- called Web Of Trust WOT. Only friends whose trust evaluation exceeds a threshold will be involved in recommendation encounter. Recommendations are obtained by weighted average of ratings along with the trust value using Film Trust dataset. In another context, Massa *et al.* [19] used trust to filter the set of neighbours and only their rated items would be considered in predicting ratings to an active user. After filtering neighbours, they applied the traditional recommendation algorithm. The experiments were based on Epinion dataset which contains both users' ratings and the direct trust values from users towards each other.

Third, recent researches have been done to exploit the sentiment in the textual reviews to augment ratings in collaborative recommenders [11], [12], [20]. Authors in [11] tried to improve the RSs by leveraging topic and sentiment information at sentences level. They inferred ratings from text reviews written by users about restaurants in multi-point rating scale rather than only positive or negative polarities. They applied text regression model to estimate scaled sentiment ratings. They are the first who integrated the useful information in reviews into RSs. Lenug *et al.* [12] proposed a probabilistic sentiment inference framework. They applied natural language techniques to compute sentiment orientation in reviews. They built their rating inference model based on the Naive Bayes classifier. Then, they integrated between the inference ratings from reviews and a CF algorithm to increase users' preferences and achieved encouraging results. Esparza *et al.* in [21] investigated how to obtain recommendation from online microblogging services. They proposed a solution to exploit short posts written by users as product reviews. These posts are used to build user-item profile. Then a query search algorithm is applied to retrieve relevant item profiles based on a twitter-like review service called blipper.com. This study is similar to our work in using microblogging as a source of recommendation.

Some inherent drawbacks still have not been solved in the above mentioned methods. Most of these approaches require users to produce some structured data first such as trust evaluations and ratings to allow the corresponding systems to work properly. In fact, this is not practical and usually not available. Nevertheless, the weaknesses of sparsity and cold start problems appear in the case of trust network as it is in the user-item rating matrix. On the other hand, review-based recommenders require a user to write reviews and rate products to generate the suggestions. Unlike existing studies, our novel approach ISTS overcomes the need of ratings or written reviews by users and reflects the real hidden social trust relations. In our work we personalise recommendations from microbloggings using sentiment analysis and trust between friends.

III. Microblogging Services

In this section we introduce our target social network Twitter. Users in the microblogger Twitter can publish short posts in 140 character limit so-called tweets. Today, Twitter users can generate more than 300 Million tweets each day [22] about different topic and interest. For example, people can generate brief posts about their personal experience in reading books, watching movies, breaking news or even the release of new electronic gadgets. In addition, users have the choice to establish relationships among each others for social links, seeking information or identifying following/followers friends.

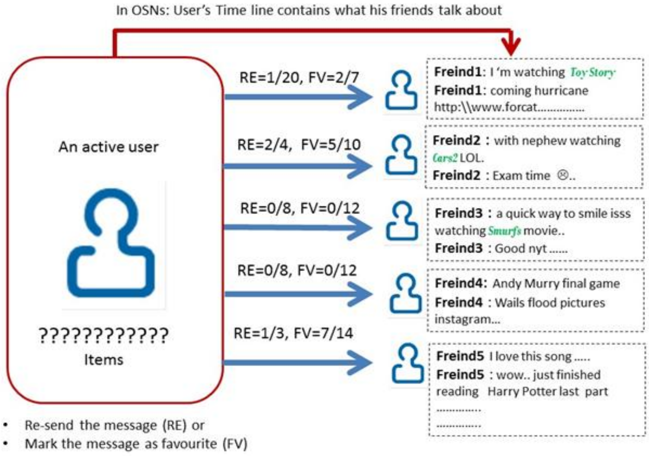

In OSNs: User's Time line contains what his friends talk about
An active user
????????????
Items
RE=1/20, FV=2/7
RE=2/4, FV=5/10
RE=0/8, FV=0/12
RE=0/8, FV=0/12
RE=1/3, FV=7/14
Freind1: I 'm watching Toy Story
Freind1: coming hurricane http:\\www.forcat..............
Freind2 : with nephew watching Cars2 LOL.
Freind2 : Exam time ☹..
Freind3 : a quick way to smile isss watching Smurfs movie..
Freind3 : Good nyt ......
Freind4: Andy Murry final game
Freind4 : Wails flood pictures instagram...
Freind5 I love this song .....
Freind5 : wow.. just finished reading Harry Potter last part
....
....
Re-send the message (RE) or
Mark the message as favourite (FV)

Fig. 1. Example of user's friends intercommunications in OSNs.

Measuring the different level of hidden and subjective trust relationships between friends in Twitter is crucial in our research. Therefore, we developed a tool to automatically collect social network data by using Twitter API. This tool extracts the required interactions between friends, and it is called Twitter Interaction Extractor (TIE), more details in Section VI. It is important to high light that we access only accounts that available to public and not any protected accounts for personal security. Nevertheless, some regulations that Twitter service applies increase the challenge of obtaining such social network data, for instance, the rate limit of accessing and requesting information from Twitter service and the dynamic change in the relations and contents. Moreover, some users choose to apply more privacy constraints on their accounts in order to avoid their information to be revealed to the public.

IV. ISTS Recommendation Framework

We believe that the social relations appearing among users and friends in OSNs influence users' purchasing behaviour. As the amount of information and opinions about products and services increase and divers more and more, harnessing friends' opinions and incorporating trust relations between them to improve recommendation are becoming a crucial need. Fig. 1 is an example for an active user who has not experienced lots of items or a completely new user in a retailer website. In this situation, drawing the user's taste and preferences is not available and most of the existing RSs algorithms cannot provide the personalised suggestions.

What normally people do when they are lack of information is to ask their friends as they trust them. People also tend to show interest and curiosity in items (movies, books, restaurants ...etc) that their friends like. Being influenced by friends' taste is a common feature in the real world. For the example in Fig. 1, the active user has five friends who broadcasted different messages, for example, Friend1 posted about movie, Friend4 posted some news about flood, or even about book such as Friend5. Let us assume the domain of movie recommenders, then the investigation of his friends' circles shows that Friend1, Friend2 and Friend3 have some knowledge about movies Toy Story, Cars2 and Smurfs. The challenge is that the active user definitely has different relationships with his peers. We need to analyse these social ties and know which friend that the active user may trust his opinion the most. To do so, it is important to take into account the communication behaviours between the active user and his friends as a trust indicator. Examples of these interactions are the action of resending messages from his friends (RE) and saving his friends' posts in favourite list (FV). It is also highly required to find out the level of sentiment orientation in friends' posts about movies and consider the informal use of language and icons.

Next section explains how to compute the implicit social trust, and then the sentiment analysis of friends' posts is introduced in Section IV-B.

A. Implicit Social Trust

In this section, we will detect some important communication activities between peers to indicate how people trust each other in OSNs. We believe that the interactions between friends in OSNs can indicate how much trust they can hold towards different friends. Some actions show how friends perceive each other such as re-tweeting, mentioning others, favouring others' posts and number of followers [23]. Communication activities in Twitter that we consider in this paper, are defined as: the action of re-tweeting which means that a user re-sends a tweet to all his/her friends to show the interest, and will be denoted as *RT*. We want to compute the trust relation between user *u* and one friend f among the group of friends *F* since $f \in F$.

Intuitionally, trust is identified as a normalised average:

$$trust_{u,f} = \frac{RT_{u,f}}{RT_{u,F}} \tag{1}$$

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

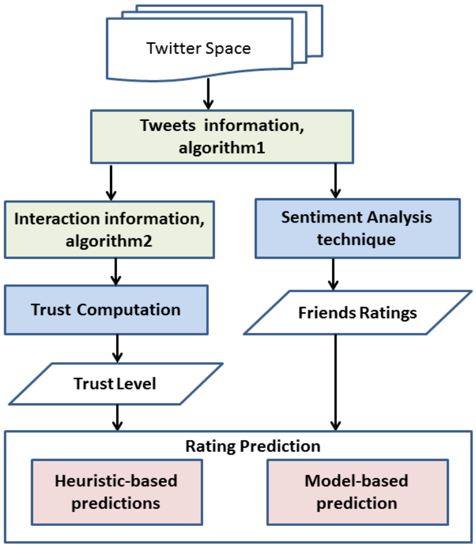

Twitter Space
Tweets information, algorithm1
Interaction information, algorithm2
Sentiment Analysis technique
Trust Computation
Friends Ratings
Trust Level
Rating Prediction
Heuristic-based predictions
Model-based prediction

Fig. 2. ISTS: The proposed framework.

Due to the fact that people interactions vary over the time and relations are not static, we define the periods of times as

$T = \{t_1, t_2, ...t_W\}$ and then the same computation of trust in equation (1) is applied for each time period $t_j \in T$,

$$trust_{u,f}(t_j) = \frac{RT_{u,f}(t_j)}{RT_{u,F}(t_j)} \tag{2}$$

Based on the above equation, we can detect the trust between u and f over all periods of time T as,

$$TRUST_{u,f}(T) = \frac{1}{T}\sum_{j=1}^{T} trust_{u,f}(t_j) \tag{3}$$

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

B. Micro-Review Sentiment Analysis

It is more challenge to apply the sentiment analysis to short and informal tweets in OSNs comparing with long standard reviews of products. Moreover, these are user generated-contents with free text style that describes some part of other people's life such as news and hobbies not only is dedicated for reviewing products. In this paper we deal with these tweets as micro-reviews denoted as *mr* and any *mr* consists of sentiment words *sw* as $mr = \{sw_1, sw_2, ..., sw_m\}$. In addition to these sentiment words, people in OSNs widely use special language features and signs to reflect their emotions and interest about any topic such as emoticons, hashtags and capitalisation. Hence, every special Twitter's language feature is considered as negative or positive sw. We consider negative and positive opinion words listed in [10] that were applied successfully in standard movie reviews, however, we eliminate any word that has not appeared in our collected corpus. Table I explains the special language features we used.

TABLE I: LIST OF NEGATIVE AND POSITIVE TWITTER FEATURES

Positive Features	Negative Features
+Intensifier	-Intensifier
+Emoticon	-Emoticon
+Hashtag	-Hashtag
+abbreviation	-abbreviation
URL	Negation

In order to extract the described features from friends' tweets and build a bag of words we applied some necessary preprocessing steps: 1) Tokenization stage where we segment tweets into separate words by punctuation marks "," and spaces based on the feature extraction unigrams model because it has proven to perform better than bigrams model particularly in movie reviews. 2) Normalization such as removing stop words which is the articles such as "a", "the" and removing user name from tweets. Negation also will be tackled, it will be marked by the word "NOT". Any referral link will be indicated by "URL". We keep the name of the movie to help the sentiment polarity understanding of words. In addition, we use the online lexicon to identify the polarity in emoticons and abbreviation (*en.wikipedia.org/wiki/Emoticon*) such as the emoticon":)" and abbreviation "OMG". Positive and negative intensifiers are also counted such as "cooool". 3) Binary vector of features is implemented manually for every micro-review and any existence of the *sw* will be given value 1 even positive or negative, otherwise it will take the zero value.

In contrast to the binary sentiment methods that provide only the polarity of reviews as negative or positive, we need to obtain more precise sentiment analysis that describes more than simple like or dislike methods. Both the natural shortness of tweets and the requirement of finer-grain of sentiment analysis draw a challenge 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 $mrj = \{swj_1, swj_2, ..., swj_m\}$. We want to infer sentiment rating s^r to hold a class of ratings $\hat{sr} = \{sr_1, sr_2, ..., sr_S\}$, for example, $\hat{sr} = \{1, 2, 3, 4, 5\}$, since our goal is to allocate a rating to describe the strength of an opinion in micro-reviews. Now we can infer sentiment ratings by aggregating all the existence of positive sw's normalised by the total number of existence mentioned features previously either positive or negative, similar method is used in [11] but they worked only at sentence level without including the special language features of OSNs. The following equation illustrates how we compute the inference sentiment rating *sr* from *mr*:

$$sr_{f,i} = \frac{P(f,mr)_i}{P_{(f,mr)_i} + N_{(f,mr)_i}} \times S \tag{4}$$

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

$$P_{(f,mr)_i} = | P_{sw \in mr_j} | \tag{5}$$

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

$$N_{(f,mr)_i} = | N_{sw \in mr_j} | \tag{6}$$

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

sw in mr_j. From equation (4), *S* will be 5 in order to form an opinion strength in the popular five scale scores. Also, we need *sr* to be in this scale in our future work.

V. ITEMS INFERENCE RATINGS

The crucial step in RSs is to generate prediction of ratings. We develop two different techniques one based on heuristic foundation to predict ratings in Section V(A) and the second is to apply machine learning models in Section V(B).

A. Heuristic Prediction

To generate predictions heuristically, let us assume that the system needs to identify a rating R for an active user u about an item i, denoted as Ru,i. The trust values will act as weights to vote for the derived sentiment ratings sr. And let the set TF refers to set of friends who have trust relationship with u and Opinion about *i*, $TF = \{(f,i)_1, (f,i)_2, ..(f,i)_m\}$ and hence $TF \subset F$. The following formula is showing how the ISTS would predict ratings $R_{u,i}$ as weighted average:

$$R_{u,\,i} = \sum_{f,\, i \in TF} \frac{trust_{u,f}}{|TF|} \times sr_{f,\,i} \tag{7}$$

TF is only the group of friends who have opinions about item *i* among all Friends *F*. And *sr* is the extracted sentiment rating from friend f about item *i*. And trust refers to the trust value between user *u* and a friend f who posts information about *i*. We have also the following properties:

1) It is important to say that trust between one user u and his friend f may be not bidirectional, hence:
   if $u \rightarrow f$ and $f \rightarrow u$ then *trust* $u \rightarrow f \neq$ *trust* $f \rightarrow u$
2) If $(trust_{u,f1}) > (trust_{u,f2})$ then, *srf*1 contributes more to *Ru, i* than *srf*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 $MAX_{f \in TF}(sr_{f,i})$ is the lowest and highest computed *sr* towards item *i* among F.

B. Models-Based Prediction

One of the aims of this work is to validate whether information comes from friends can contribute to products ratings estimation by using classification learning algorithms. We conducted experiments with three selected algorithms:

Naive Bayes classification, Logistic regression and Decision tree. The importance of these algorithms come from yielding good result in different domain and the availability of the related software tools [24]. Applying these machine learning methods on our social data SD $=\{trust_{u,f}, sr_{f,i}\}$ needs to identify the relations between user-rating and features terms by IR which indicates the impact of each friend's opinion sr on user-rating about *i*, denoted as $IR_{(u,\,f,\,i)}$ and computed as followings:

$$IR_{(u,\,f,\,i)} = g(trust_{u,\,f}, sr_{f,\,i}) \tag{8}$$

where *g* is an unknown function we need to define. And then we aggregate the impact function I R to obtain the final user ratings as:

$$R_{u,\,i} = \frac{\sum_{f \in F} IR_{(u,\,f,\,i)}}{|TF|} \tag{9}$$

The above equation can be simplified as:

$$R_{u,i} = averageIR_{(u,\,f,\,i)} \tag{10}$$

Also, we can see that equation (7) is a special case under equation (8) when it is defined as follows:

$$g(trust_{u,\,f}, sr_{f,\,i}) = trust_{u,\,f} \times sr_{f,\,i} \tag{11}$$

The 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 $C = \{\, c_1, c_2, \dots c_m \,\}$ and feature vector $V = \{\, v_1, v_2 \dots v_n \,\}$. We examined the classification models on the social data dataset SD to predict one of five nominal classes of ratings *ExtremeLike*, *Like*, *Neutral*, *Dislike*, *ExtremeDislike*. In fact, the core engine behind these three classification algorithms are quite different. Next paragraphs describe brief details about each.

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 constraints to allocate a class C_i if and only if:

$$P(c_{i|} \,|\, X) > P(c_j \,|\, X) \quad \text{for } i, j \in \{1, ..\, m\} \tag{12}$$

Using the Bayes's rule to derive the posterior probability $P(C_i|X)$ as the following formula:

$$P(c_i \,|\, X) = \frac{P(c_i) P(X \,|\, c_i)}{P(X)} \tag{13}$$

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 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 [25]. We describe two more algorithms might achieve better results.

Another model we test is a Logistic Regression method. 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 consider as linear model with more power to refine the parameters by minimising the error function. The output can be interpreted as probability gives the confidence that a prediction belongs to a certain class. If we assume m number of classes $C_j = \{\, c_1, c_2, \dots c_m \,\}$ and number of examples with n number of features. The parameter matrix B is calculated as $n * (m-1)$. The probability that C_j with the exception of the last class is,

$$P(c_{j|} \,|\, X_i) = \frac{\exp(x_i B_j)}{\sum_{j=1}^{m-1} \exp(x_i B_j) + 1} \tag{14}$$

More details about ridge estimator in logistic regression can be found in [26].

A decision tree model is also applied. It is a nested set of rules that used to split the data. This recursive algorithm constructs a tree structure automatically starting from root features and ending with leaf nodes. When splitting the data a decision rule is applied for every feature then the feature that minimises the cost function is chosen to build 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 that used to measure the best splitting of data by C4.5 algorithm is called the information gain IG measurement derived from the dataset itself to split the tree branches. Let p_i is the probability that a subset of SD labeled by c_i, then:

$$I(p) = -\sum_{c_i \in C} p_i \log_2 p_i \tag{15}$$

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 values a_i as one of possible values taken from set *vals* $a_i \in$ *vals*:

$$E(v_i) = \sum_{a_i \in vals} \frac{|x \in SD \mid x = a_i|}{|SD|} I(p) \tag{16}$$

Building a classifier using a decision trees is very attractive due to several advantages. For instance, it is not expensive to construct and it shows fast computation time at classification. Another important advantage of decision tree that we can obtain a set of rules that are easy to interpret when applying an accuracy comparison to other well-known classification methods. More discussion about the area of decision tree can be found in [27]. We used Weka3.6.9 library for building and learning all the aforementioned models.

VI. Experimental Analysis

A. Dataset

The dataset that we need must contain friends published posts (micro-reviews) from OSNs and trust relations besides the classic users and items ratings information. In fact, there is no adequate dataset with the predefined requirements since the available datasets contain only ratings, only reviews or ratings with reviews. Given the absence of friends information, therefore, we built a software tool TIE to scrawl Twitter and prepare the friends information using the Twitter API for JAVA (contact the authors for the collected dataset).

We could collect social data about 111 users' information as follows: firstly, we randomly choose movies form the popular movielens dataset (http://grouplens.org/datasets/movielens/), after that we used the search tool TIE to gather information about a person who posted a tweet about the chosen movie such as name and twitter ID. Secondly, since we were able to allocate the publisher information then we start to detect the re-tweeting messages activities rate between this person and his friends. On the other hand, we used three annotators, who we considered as active Twitter users with no less than ten tweets per day, to annotate the collected scenarios given re-tweets information and sentiment rating

B. Metrics

To compare the performance between the three applied machine learning classification algorithms we applied different metrics. We used Accuracy metric to indicate the percentage of the correctly classified instances in the test set. However, this metric is not enough because it is not sensitive to class distribution or the chance of being correct. Hence, we also used the standards evaluation measurements that are widely used in information retrieval and classification such as Precision and Recall defined in equations (17), (18) respectively. These two metrics test the accuracy of classification algorithm in predicting ratings [25].

$$precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{17}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{18}$$

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:

$$F - measure = 2.\frac{Precision.Recall}{Precision + recall} \tag{19}$$

For testing, we randomly split the dataset into ten non-overlapped folds to apply 10-fold cross-validation. The experiments are repeated on the ten folds. Every fold is used as test set and the rest nine folds used as training set. Our results are computed based on the average of all the ten folds models.

We also applied the statistical accuracy metrics such as Mean Absolute Error (MAE) to evaluate the recommendation algorithm. It is the most widely used and acceptable in the recommendation community because it is easy to apply and we can interpret comparisons directly. *MAE* is defined,

$$MAE = \frac{\sum_{u,i \in N} |r_{u,i} - r_{u,i}|}{N} \tag{20}$$

where N is the size of the test set, and $r_{u,i}$ is the rating assigned by user u to movie i, and $\hat{r}_{u,i}$ indicates the rating estimated by the proposed recommendation algorithm. Obtaining small results of MAE shows more accurate performance of the system. To show the quality of our approach we need to compare the results with the following baselines:

B1: Item average score baseline (IAS), this method gives any unknown missing values the mean value of the corresponding item ratings.

B2: Most dominated score baseline (MDS), this method assigns the majority of the class ratings in the dataset to any unknown ratings, and it is the 5 score in our dataset SD.

B3: Trust-based weighted mean approach proposed in [18]. It is a trust-aware recommender and uses explicit trust values. It does not involve any similarity measurement which makes it more appropriate to compare with.

B4: Trust-based collaborative filtering approach proposed by Massa *et al*. [19] which based on the fact that similarity and trust are correlated when applying traditional collaborative filtering technique and we applied it as item-based baseline.

For B3 and B4 we adjusted the chain of trust path to one as authors in [18] found that shorter propagated path yield better results. All results from all methods rounded to the nearest class to allow unbiased comparisons since results from classification models and actual user ratings are all come in discrete class values. For example if the output rating is 3.75 or 3.5 it will be rounded to be 4, and if the output 2.2 and 2.33 will be the class rating of 2. We denote our method proposed in Section V-A as **I**ST S_1 and the method proposed in Section V-B as **I** ST S_2 (based on best accuracy gained by tree decision model).

C. *Classification Results*

Experiment is applied to evaluate the recommendation classification algorithms accuracy given the computed trust and sentiment values derived from the collected data described in Section VI-A.

From Table II, we can compare the performance of the three algorithms by looking at F-measure of NB, logistics regression and decision tree were 0.57, 0.67 and 0.72 respectively. This means that decision tree has power of prediction higher than the two others algorithms. It is clear that NB classifier has the worst accuracy percentage while the best performance is given by decision tree it can correctly predict 72% of test ratings. More specifically, Fig. 3 sheds the light on these metrics results using decision trees according to each class category. There is more determination about negative classes Dislike and *ExtermeDislike* than positive classes. Meanwhile the Neutral class gains the lowest accuracy and this may due to the ambiguous nature with this class since it holds uncertainty about opinion. It is difficult to know whether Neutral class is closer to which polarities (negative or positive) unless further contextual information is included such as demographic information or more trust indicators are used.

TABLE II: EVALUATIONS OF THE CLASSIFICATION RESULTS

Algorithms	Accuracy	Precision	Recall	F-measure
Na ïve Bayes	57.65%	0.58	0.58	0.57
Logistic-Regression	67.6%	0.66	0.68	0.67
Decision Trees	72.1%	0.73	0.72	0.72

We observed that trust is the feature used as criteria to split tree branches. Trust value which is used as first threshold to divide the tree is 0.17. If the trust is below this threshold, the algorithm does not estimate any satisfaction classes such as ExtremeLike and Like or even Neutral. Also, by looking at confusion matrix, we can notice that most of the incorrect estimations were allocated in nearest classes. For example, the decision tree algorithm allocates the wrong ratings estimation that associated with Like label in neighboring classes: Neutral or ExtremeLike which are semantically relevant rather than going further distance classes such as Dislike or ExtermeDislike.

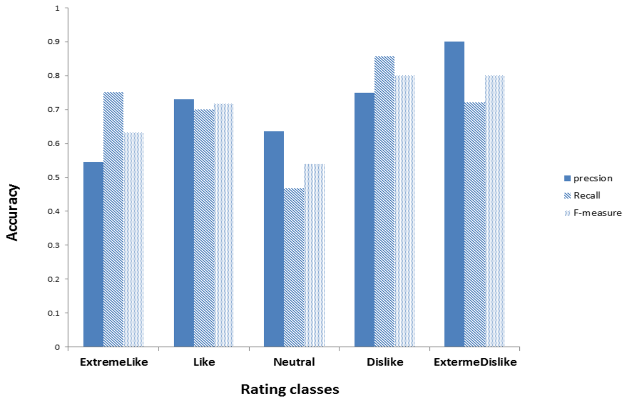


Fig. 3. Precision, recall and F-measure for each ratings class produced by decision trees.

D. *Recommendation Results*

We will evaluate the performance of our approach having group of friends. In this experiment, we assume the existence of totally 12 new users. Each user is assigned a group of friends of size 10 sampled from the dataset SD. Moreover, we applied a threshold of the lowest acceptable trust value. Only friends who exceeding this threshold will be involved in recommendation process. We chose the value of the threshold to be 0.17 as this value of trust is the information gain in decision tree model. This threshold will be applied for all baselines. After selecting the trusted friends, this local community will be treated as user's neighbourhood. Table III reports different values based on the average of MAE from friends groups of the 12 users. 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. Thus, traditional collaborative filtering lead to poor recommendation in new user situations. Due to the large error obtained by B4 we excluded it in the next experiment. It can be observed that both of our methods **I** ST S_1 and **I** ST S_2 gained the smallest error results 1.2 and 0.836 respectively.

TABLE III: MAE VALUES USING SIZE OF FRIENDS GROUP OF TEN

Algorithms	MAE
B1	1.292
B2	1.735
B3	1.278
B4	3.236
ISTS$_1$	1.20
ISTS$_2$	0.836

E. *Comparisons Results*

In this experiment we used the entire collected data SD for testing. We also considered the smallest group of friends (e,g. contains only one friend) to show how these methods would perform in term of accuracy in a very sparse group of friend.

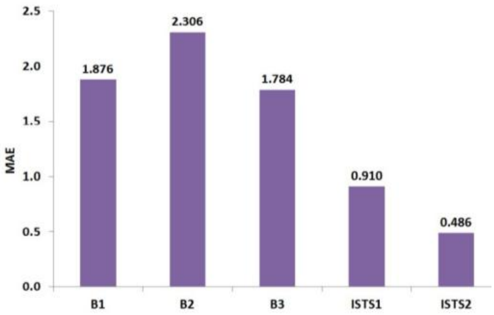


Fig. 4. Performance comparison using MAE.

In fact, the results from this experiment came consistent with the previous one. From results are shown in Fig. 4, we can noticed that our methods outperform other methods and can generate better result than the baselines of trust-aware recommenders. We also observed that the second approach ISTS_2 outperforms the first one since it acquires 0.486 the smallest *MAE* result. An explanation of this is the fact that, ISTS_2 is based on learning models which is always achieve higher accuracy, however, ISTS_1 is heuristic method and easier to capture changes in tastes and detect changes in trust behaviours more than learnt models but with lower accuracy. Broadly, methods that utilise trust performed better than other baselines and it is also observed in [28].

VII. Conclusion

In this paper, we explore the potential of social information derived from microbloggings as a source of user relevant recommendations. In contrast to traditional RSs which are based mainly on structured data, we investigate the data comes from the current web environment. We propose the approach ISTS that can exploit two factors from OSNs: the sentiment orientation in friends posts about certain items and the trust relations between friends. Our evaluations are applied on real social data from Twitter. The results show that the these short and inconsistent posts can empower the users preferences data in particular when no preferences of history were available. Several machine learning classification algorithms were used to classify a score rating, and the tree decision model performs the best accuracy metrics results. In the future challenge, we believe that user's taste is absolutely very important in personalisation. Hence we plan to fuse user's own preferences - when exists - with social information comes from OSNs to enrich and augment collaborative filtering recommenders.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," *Recommender Systems Handbook*, pp. 1–35, 2011.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," in *Proc. the 1994 ACM Conference on Computer Supported Cooperative Work*, ACM, 1994, pp. 175–186.

[4] B. Sarwar *et al*., "Item-based collaborative filtering recommendation algorithms," in *Proc. the 10th International Conference on World Wide Web*, ACM, 2001, pp. 285–295.

[5] 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.

[6] L. Si and R. Jin, "Flexible mixture model for collaborative filtering," in *Proc. Machine Learning-International Workshop then Conference*, vol. 20, no. 2, 2003, p. 704.

[7] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2009, pp. 203–210.

[8] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. the Fourth ACM International Conference on Web Search and Data Mining*, ACM, 2011, pp. 287–296.

[9] Y. Kim and R. Phalak, "A trust prediction framework in rating-based experience sharing social networks without a web of trust," *Information Sciences*, 2012.

[10] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. the ACL-02 Conference on Empirical Methods in Natural Language Processing*, vol. 10, 2002, pp. 79–86.

[11] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," presented at 12th International Workshop on the Web and Databases, 2009.

[12] 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

[13] 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.

[14] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, p. 4, 2009.

[15] 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.

[16] X. Su and T. M. Khoshgoftaar, "Collaborative filtering for multi-class data using belief nets algorithms," in *Proc. International Conference on Tools with Artificial Intelligence*, IEEE, 2006, pp. 497–504.

[17] B. M. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Recommender systems for large-scale e-commerce: Scalable neighborhood formation using clustering," in *Proc. the Fifth International Conference on Computer and Information Technology*, vol. 1, 2002.

[18] J. A. Golbeck, "Computing and applying trust in web-based social networks," 2005.

[19] P. Massa and P. Avesani, "Trust-aware recommender systems," in *Proc. the 2007 ACM Conference on Recommender Systems*, ACM, 2007, pp. 17–24.

[20] F. Peleja, P. Dias, and J. Magalha˜es, "A regularized recommendation algorithm with probabilistic sentiment-ratings," in *Proc. 12th International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2012, pp. 701–708.

[21] S. G. Esparza, M. P. O'Mahony, and B. Smyth, "Mining the real time web: A novel approach to product recommendation," *Knowledge-Based Systems*, vol. 29, pp. 3–11, 2012.

[22] S. M. Kywe, E.-P. Lim, and F. Zhu, "A survey of recommender systems in twitter," *Social Informatics*, Springer, 2012, pp. 420–433.

[23] B. Hajian and T. White, "Modelling influence in a social network: Metrics and evaluation," in *Proc. Third International Conference on Privacy, Security, Risk and Trust (Passat)*, IEEE, 2011, pp. 497–500.

[24] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann, 2005.

[25] X. Amatriain, A. Jaimes, N. Oliver, and J. M. Pujol, "Data mining methods for recommender systems," *Recommender Systems Handbook*, Springer, 2011, pp. 39–71.

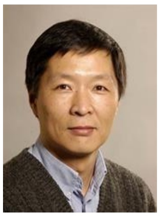
[26] S. L. Cessie and J. V. Houwelingen, "Ridge estimators in logistic regression," *Applied Statistics*, pp. 191–201, 1992.

[27] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers-a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 4, pp. 476–487, 2005.

[28] P. Victor, M. D. Cock, and C. Cornelis, "Trust and recommendations," *Recommender Systems Handbook*, Springer, 2011, pp. 645–675.

Dimah Alahmadi has a bachelor degree (1999) in computer science from King AbdulAziz University (KAAU), Jeddah, Saudi Arabia, and the master degree (2003) in information system from (KAAU). She is a lecturer in Information System Department in the Faculty of Computing and Information Technology at KAAU. She is currently a PhD student in

Manchester University in Machine Learning and Optimization Group (MLO), Manchester, UK. Her current interest is in recommender systems (RSs).



Xiao-Jun Zeng received the B.Sc. degree in mathematics and the M.Sc. degree in control theory and operation research from Xiamen University, Xiamen, China, and the Ph.D. degree in computation from the University of Manchester, Manchester, U.K. From 1985 to 1992, he was with the Department of Computer and Systems Sciences, Xiamen University, where he was a lecturer and an associate professor. From 1996 to 2002, he was with Knowledge Support Systems (KSS) Ltd., Manchester, where he was a scientific developer, a senior scientific developer, and the head of research, developing intelligent decision support systems. He has been with the University of Manchester since 2002, where he is currently a senior lecturer with the School of Computer Science. His current research interests include fuzzy systems and control, neural networks, machine learning and data mining, decision support systems and game theory, and health informatics. Dr. Zeng is an associate editor of the IEEE Transactions on Fuzzy Systems and a member of the peer review college of the U.K. Engineering and Physical Sciences Research Council. He was a recipient of the European Information Society Technologies Award in 1999 and the Microsoft European Retail Application Developer Awards in 2001 and 2003 with KSS Ltd, Manchester, UK.