Trust-Based Infinitesimals for Enhanced Collaborative Filtering

Maria Chowdhury, Alex Thomo, Bill Wadge

University of Victoria
BC, Canada
{mwchow,thomo,wwadge}@cs.uvic.ca

Abstract

In this paper we propose a novel recommender system which enhances user-based collaborative filtering by using a trust-based social network. Our main idea is to use infinitesimal numbers and polynomials for capturing natural preferences in aggregating opinions of trusted users. We use these opinions to "help" users who are similar to an active user to come up with recommendations for items for which they might not have an opinion themselves. We argue that the method we propose reflects better the real life behaviour of the people. Our method is justified by the experimental results; we are the first to break a stated "barrier" of 0.73 for the mean absolute error (MAE) of the predicted ratings. Our results are based on a large, real life dataset from Epinions.com, for which, we also achieve a prediction coverage that is significantly better than that of the state-of-the-art methods.

1 Introduction

One of the key innovations in on-line marketing is the creation of recommender systems for suggesting new interesting items to users (or buyers). In essence a recommender system (RS) tries to predict the ratings that users would give to different items. The recommendations should be of good quality, otherwise the users would soon loose the confidence on the RS and consider it just another spamming annoyance. On the other hand, the system should be able to recommend a good range of items to the users, not just few ones. These two desiderata, the quality of the predictions, and the coverage of recommendations are often two conflicting goals. Typically, the better the quality of predictions is, the worse the coverage gets, and vice versa. In this paper we propose a novel recommender system which has at the same time both high quality of predictions as well as great item coverage.

15^{th} International Conference on Management of Data COMAD 2009, Mysore, India, December 9–12, 2009

The quality of predictions is usually measured by the Mean Absolute Error (MAE), which is computed by trying to predict the existing, real user ratings (after they are hidden) and then compute the differences of the predictions from these real ratings. On the other hand, the coverage is estimated as the percentage of the existing ratings that the RS is able to approximate.

Regarding MAE, there exists a belief that there is some inherent "magic barrier" below which MAE cannot go. As the seminal work by Herlocker et. al. ([3]) puts it, the recommender systems working with ratings in a scale from one to five hit a MAE barrier of 0.73, and achieving a MAE below that is very difficult due to "the natural variability" of humans when rating items.

In this paper, we are the first to break the above barrier. We achieve this by using the power of the underlying Epinions social network expressed by trust statements between the users. Furthermore, we do not improve the MAE by compromising the coverage. In fact our coverage is significantly better than the coverage of the state-of-the-art methods.

Motivation and Main Idea. On-line social networks have become a very important part of decision making and other activities in our daily lives. We use these systems to communicate with each other, to make new friends, to buy or sell products on-line, to collect reviews of products, to play games etc. Online social networks such as *Orkut, Facebook, LinkedIn, MySpace*, etc have a huge number of users and there is a tremendous increase in the number of users every day.

We believe that looking for recommendations is one of the most important uses of social networks. In this paper we introduce a new recommender system which leverages the power of a social network created by users who issue trust statements with respect to other users. The trust statements, over time, create a precious web of trust, which, as we show, can be used to significantly enhance the quality of recommendations.

Notably, our system blends together collaborative filtering and trust-based reasoning. Collaborative filtering (CF) identifies similar users based on the product ratings that the users have issued over time. The similarity between two users is typically determined by calculating the Pearson correlation between the users' rating vectors. Then, the recommendation to a user u for an item i is generated by averaging the ratings of the similar users for item i.

To illustrate, given a user, say Bob, and an item, say HP laptop, in order to generate a recommendation for HP laptop to Bob, CF will find the users who are similar to Bob, say Alice and Jon, and then average their ratings for HP laptop.

The problem is: "What if Alice or Jon, or both, do not have a rating for the HP laptop?" Clearly in such cases the system would suffer from data sparsity and the quality of recommendations will degrade.

The intuition behind our solution is to make "Alice" and/or "Jon" (in this example) "come up" with an opinion about HP laptop by using the available trust-based social network. Based on how friendship connections are evaluated in real life, we propose that a user aggregates first the opinions of his/her (immediate) friends, and considers the opinions of the friends-of-friends only if the friends are unable to provide some opinion. If the latter happens, then the opinion of a "second degree" friend who is trusted by many "first degree" friends should be more important than the opinion of some other "second degree" friend who is trusted by fewer "first degree" friends. This idea can be naturally generalized to more than one or two levels of friendship connections.

We believe that this reasoning reflects better the people's real life behaviour; we trust our friends "infinitely" more than the friends-of-friends, and we trust them in turn "infinitely" more that the friends-of-friends-of-friends, whom, after all, in real life, we might not even know at all.

Contributions. Specifically, our contributions in this paper are as follows.

1. We present a method to inject the power of a trust-based social network into Collaborative Filtering.

2. We propose the idea of having ratings which are "infinitely" more important than other ratings. We capture this by using infinitesimal numbers and polynomials. In this way we obtain a framework in which one can elegantly set qualitative preferences or semantics for a recommender system.

3. We present a recursive formula for aggregating user opinions based on our framework of rating polynomials. The aggregated opinion given to a user u for an item i is concisely expressed as a Hadamard division of two infinitesimal polynomials.

4. We present a detailed experimental evaluation of our system and show that it significantly outperforms state-of-the-art methods both with respect to the quality of recommendations as well as their coverage.

Organization. The rest of the paper is organized as follows. In Section 2 we give an overview of related works. In Section 3 we present an outline of our method. In Section 4 we present the hyperreal numbers which include both the real and infinitesimal numbers. In Section 5 we give the main data structures used by our method. In Section 6 we present our hybrid CF-and-trust-based recommendation method. In Section 7 we show the results of our evaluation. Finally, Section 8 concludes the paper.

2 Related Works

Using trust networks for recommender systems has been identified as a promising direction for improving the quality of recommendations. Some important works on trust-based recommender systems are [6, 9, 2, 1, 8, 12]. They study different aspects of trust-based recommenders, such as computing or inferring the trustworthiness of the users, devising effective trust metrics, applying trust-based recommendation techniques in specific domains etc.

In [7], Massa and Avesani present a deep comparative study of trust-based recommender systems vs. a classical recommender system based on Collaborative Filtering. We believe that [7] is seminal in that it shows that recommendations based on local trust neighborhoods are significantly better than recommendations based on global reputation systems such as Google's PageRank [10]. Notably, Massa and Avesani derive these results on a large, real-life dataset, which they collected from the Epinions.com site. We also use this dataset[1] for evaluating our system. Furthermore, they propose studying the effectiveness of recommender systems not only on all users and items, but also on several well-chosen, critical categories of users and items.

[4] is a recent work which proposes a random-walk method for combining trust with item-based recommendations. On the other hand, our method combines trust with user-based recommendations. The ideas and techniques used in these two works are very different. Comparing [4] with our method in terms of recommendation quality is a direction for future work.

Evaluating recommender systems in a reliable way is certainly very important. The most authoritative work on the evaluation of recommender systems is by Herlocker et. al. [3]. It is there that the stated barrier of 0.73 on MAE is mentioned. This value is based

[1]Actually, what is available is a slightly different version which can be downloaded from www.trustlet.org

on their experiments as well as experiments performed by other works. Another metric, that they (as well as Massa and Avesani in [7]) suggest, is the recommendations coverage which is the fraction of ratings that the system is able to predict after the ratings are hidden. We evaluate our method using both MAE and recommendations coverage.

3 Outline of our method

The ratings of a set of users for a set of items can be visualized as a matrix M with users as rows and items as columns. The rating that a user u has given for an item i is the value $M[u, i]$. Of course this matrix is very sparse; most of the entries are zero because typically each user has rated only a handful of items. As such, the user-item matrix is never explicitly built, but this does not hinder the similarity computations of Collaborative Filtering.

The ratings of a user u are all the non-zero entries in the corresponding matrix row. Given an *active user* (row) u_a for which we want to generate recommendations, the (user-based) Collaborative Filtering method works by computing the similarity of row u_a with each other row in the matrix. Only the non-zero entries are considered. Typically, the Pearson correlation is used for determining the similarity between two rows, u_1 and u_2. Let the pairs of non-zero ratings for u_1 and u_2 be $(x_1, y_1), \ldots, (x_n, y_n)$, i.e. u_1 and u_2 have n items in common which they have *both* rated. The Pearson correlation for u_1 and u_2 is computed as

$$p_{1,2} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum x_i)^2 - (\sum (y_i)^2}}.$$

The Pearson correlation has been shown to perform well in practice for indentifying users who have similar tastes to a given active user.

By computing the Pearson correlation of the active user u_a against the other users in the database, a set of *similar* users is identified. Let $S = \{u_1, \ldots, u_m\}$ be this set. Then, the recommendation that u_a gets for an item i is a weighted average of the ratings that the users in S have for i. Specifically, if we denote by $u_k[i]$ the rating of user u_k for item i, then the recommendation is

$$r_{a,i} = \frac{\sum_{u_k \in S \wedge u_k[i] \neq 0} u_k[i] \cdot p_{a,k}}{\sum_{u_k \in S \wedge u_k[i] \neq 0} p_{a,k}}. \quad (1)$$

The **problem** is:

> When $u_k[i]$ is zero, user u_k, who is similar to u_a, cannot participate in the computation of recommendation $r_{a,i}$, and this harms the quality of the recommendation.

To alleviate this problem we will use the available trust network to derive an opinion for a user u_k on item i. Our desiderata in using a trust network are as follows.

1. The opinions of (trusted) friends at distance d (in the trust graph) matter *infinitely* more than the opinions of friends at distance $d + 1$. In other words, friends at distance $d + 1$ start to matter only if the friends at distance d cannot come up with an opinion on item i.

2. If f_1 and f_2 are two friends of u_k at distance $d + 1$ in the trust graph, and f_1 is trusted by more friends at distance d than f_2, then the opinion of f_1 should be more important than the opinion of f_2. In other words, even if the friends at distance d cannot come up with an opinion on item i, they can still influence the weighting of the opinions of the friends at distance $d+1$. This is recursive. As such, the friends at distance d can in fact influence the weighting of the opinions of friends at distance $d + k$, for $k \geq 1$.

In order to capture these desiderata we propose a framework based on infinitessimal numbers and polynomials.

4 Hyperreal numbers

Hyperreal numbers were introduced in calculus to capture "infinitesimal" quantities which are infinitely small and yet not equal to zero. Formally, a number ϵ is said to be *infinitely small* or *infinitesimal* (cf. [5]) iff -a< ϵ <a for every positive *real* number a. Hyperreal numbers contain all the real numbers and also all the infinitesimal numbers. There are principles (or axioms) for hyperreal numbers (cf. [5]) of which we mention:

Extension Principle.

1. The real numbers form a subset of the hyperreal numbers, and the order relation $x < y$ for the real numbers is a subset of the order relation for the hyperreal numbers.

2. There exists a hyperreal number that is greater than zero but less than every positive real number.

3. For every real function f, we are given a corresponding hyperreal function f^* which is called the *natural extension* of f.

Transfer Principle. Every real statement that holds for one or more particular real functions holds for the hyperreal natural extensions of these functions.

In short, the Extension Principle gives the *hyperreal* numbers and the Transfer Principle enables carrying out computation on them. The Extension Principle says that there does exist an infinitesimal number, for example ϵ. Other examples of hyperreals numbers, created using ϵ, are: ϵ^3, $100\epsilon^2 + 51\epsilon$, $\epsilon/300$.

For $a, b, r, s \in \mathbb{R}^+$ and $r > s$, we have $a\epsilon^r < b\epsilon^s$, regardless of the relationship between a and b.

If $a\epsilon^r$ and $b\epsilon^s$ are used for example to denote two preference weights, then $a\epsilon^r$ is "infinitely better" than $b\epsilon^s$ even though a might be much bigger than b, i.e. coefficients a and b are insignificant when the powers of ϵ are different. On the other hand, when comparing two preferential weights of the same power, as for example $a\epsilon^r$ and $b\epsilon^r$, the magnitudes of coefficients a and b become important. Namely, $a\epsilon^r \leq b\epsilon^r$ ($a\epsilon^r > b\epsilon^r$) iff $a \leq b$ ($a > b$).

For our purposes in this paper we fix an infinitesimal number, say ϵ, and consider polynomials (or finite series) of the powers of ϵ. Such polynomials are compared using the above rules. Namely, given two polynomials p and q we first compare their terms of the lowest ϵ power. If the coefficients of these terms are equal, then we continue by comparing the terms of the second lowest ϵ power, and so on. For example, $3\epsilon + 4\epsilon^2 + 10\epsilon^3$ is smaller than $3\epsilon + 5\epsilon^2 + 6\epsilon^3$ because although their coefficients of the lowest ϵ power are equal (value 3), the coefficients of the next lowest ϵ power, ϵ^2, are not, namely the coefficient in the first polynomial is 4 and thus, smaller than 5, which is the corresponding coefficient in the second polynomial. In this example, the third terms of the polynomials are not used in the comparison. They would have been used only if all the coefficients of the previous terms were respectively equal.

Given two polynomials $p = a_1\epsilon^{r_1} + \ldots + a_n\epsilon^{r_n}$ and $q = b_1\epsilon^{r_1} + \ldots + b_n\epsilon^{r_n}$, where $a_1, \ldots, a_n \in \mathbb{R}$, $b_1, \ldots, b_n \in \mathbb{R}$ and non-zero, and $r_1, \ldots, r_n \in \mathbb{N}$, the Hadamard quotient of p by q is defined as

$$p /\!\!/ q = (a_1/b_1)\epsilon^{r_1} + \ldots + (a_n/b_n)\epsilon^{r_n}.$$

We will use the Hadamard quotient of polynomials when deriving the rating predictions later on in the paper.

5 Data Structures

A social network is a social structure made of nodes that are tied by one or more specific types of interdependency, such as values, visions, ideas, friendship, etc; resultant is a graph-based structure.

In this paper we consider social networks based on trust statements that users have issued for other users. We call such a social network a trust graph and denote it by TG. This is a directed graph with nodes representing users, and edges representing trust statements. Specifically an edge (u, u') going from (user) node u to node u' expresses the fact that user u trusts user u'.

The other structure that is involved is a nested hash table representing the sparse matrix of the user ratings. This hash table contains user-item-rating triples (u, i, r), which are first hashed with respect to u and then i. We denote this set of user ratings by UR.

Given a (user) node u in TG, we denote the set of its immediate neighbor nodes by N_u.

6 Trust-Based Recommendation Polynomials

Now suppose that a user u is asked for an opinion on item i. We introduce a recursive formula which computes this opinion based on the opinions, or ratings, for i of the neighbors of u in trust graph TG. Our formula captures the intuition that for a user u the opinions of neighbors at distance d are infinitely more important than the opinions of neighbors at distance $d + 1$. Thus, the opinions of u's immediate neighbors in TG are inifinitely more important than the opinions of the neighbors-of-neighbors, and so on.

User u can already have an opinion (rating) about product i. We consider the personal opinions of users to be infinitely more important than the opinions of their neighbors. When user u has already an opinion for item i, it might be tempting to think that u does not need a recommendation for i. However we argue that providing trust-based recommendations, even in such cases, is nevertheless useful. These recommendations have the potential to *influence* or *calibrate* the opinion of a user for an item against some other, competing item, for which the user had the *same* opinion initially. For example suppose that a user had tried in the past both an HP laptop and an LG one, and had created the same opinion (rating), say 4, for both of them. Now this user wants to buy an HP or LG laptop, but does not have a definitive "winner" in her head. If she gets from the neighbors the recommendations of 2 and 5 for these products, respectively, then she might better prefer the LG laptop as opposed to the HP one.

Let $o_{u,i}$ be the opinion (or rating) user u has for item i. If u does not know or has not created yet an opinion about i, then this value is 0.

Our formula for computing/calibrating u's opinion on item i is

$$q_{u,i} \quad = \quad o_{u,i} + \epsilon \cdot \sum_{v \in N_u} q_{v,i} \qquad (2)$$

where ϵ is the infinitessimal number that we fixed in Section 4, and N_u is the set of u's immediate neighbors in TG. We compute the above for each user u and each item i. This is "one pass" or "one iteration" over the data. We repeat this procedure several times. Initially, the $q_{u,i}$ values are equal to the $o_{u,i}$ values. After each iteration the opinions get updated with the (aggregated) opinion of neighbors, then the opinion of the neighbors-of-neighbors, and so on.

Example 1 *Consider the trust graph given in Figure 1. There are four users represented by the nodes 1,*

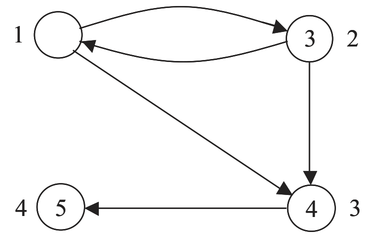

Figure 1: Example of a trust graph. The numbers outside the nodes are the node ids. The numbers inside the nodes are the ratings that the users have given to some item. These ratings correspond to the same item under consideration.

2, 3, and 4. These node ids are shown in the exterior of the circles representing the nodes. In this example we consider one item only. The ratings that the users have given to this item are shown in the interior of the nodes. Node 1 does not have a rating for the item on focus.

In the following computations we have kept only the subscript for the user and dropped the subscript for the item as we are considering only a single item in this example.

Initially, the q-opinions are the same as the corresponding user ratings, i.e. $q_1 = o_1 = 0$, $q_2 = o_2 = 3$, $q_3 = o_3 = 4$, *and* $q_4 = o_4 = 5$.

Now, formula 2, gives rise to the following equations for nodes 1, 2, 3, and 4, respectively.

$$\begin{aligned} q_1 &= o_1 + \epsilon \cdot (q_2 + q_3) \\ q_2 &= o_2 + \epsilon \cdot (q_1 + q_3) \\ q_3 &= o_3 + \epsilon \cdot q_4 \\ q_4 &= o_4. \end{aligned}$$

After the first iteration, the q-opinion values are

$$\begin{aligned} q_1 &= 0 + \epsilon \cdot (3 + 4) = 7\epsilon \\ q_2 &= 3 + \epsilon \cdot (0 + 4) = 3 + 4\epsilon \\ q_3 &= 4 + 5\epsilon \\ q_4 &= 5, \end{aligned}$$

whereas, after the second iteration, the q-opinion values are

$$\begin{aligned} q_1 &= 0 + \epsilon \cdot (3 + 4\epsilon + 4 + 5\epsilon) = 7\epsilon + 9\epsilon^2 \\ q_2 &= 3 + \epsilon \cdot (7\epsilon + 4 + 5\epsilon) = 3 + 4\epsilon + 12\epsilon^2 \\ q_3 &= 4 + 5\epsilon \\ q_4 &= 5. \end{aligned}$$

■

As it can be observed, the q-opinions are polynomials of the powers of ϵ with real numbers as coefficients. As such they can be compared, and thus ranked, using the rules described in Section 4. Namely, if there is more than one item, then when comparing the respective recommendation polynomials, the coefficients of an ϵ power become relevant only when the coefficients for all the lesser powers of ϵ are respectively the same in the polynomials that are being compared.

It can be verified that the immediate neighbors of a (user) node contribute their ratings to the coefficent of ϵ; the neighbors-of-neighbors contribute their ratings to the coefficient of ϵ^2, and so on. In general, a neighbor at distance d contributes to the coefficient of ϵ^d.

Also, it can be verified that if a neighbor at distance $d+1$ is a direct neighbor of k neighbors at distance d, then this neighbor will contribute k times its opinion (rating) in the calculation of the coefficient of ϵ^{d+1}. Thus, the second of the desiderata in Section 3 is satisfied.

Several iterations of applying formula 2 give us polynomials which can be compared and thus ranked. What is needed next is to normalize the coefficients of these polynomials, in order for the coefficients to be in the proper rating scale (for instance, from one to five).

For this, during an iteration, we not only calculate formula 2, but also calculate the following formula for accumulating the counts of nodes that contribute to the coefficients of ϵ.

$$c_{u,i} = e_{u,i} + \epsilon \cdot \sum_{v \in N_u} c_{v,i} \qquad (3)$$

where $e_{u,i}$ is a 0/1 integer which is 1 if there exists a rating for item i in node u, and 0 otherwise.

Finally, the coefficients of the $q_{u,i}$ polynomials are normalized by taking their Hadamard quotient with the corresponding $c_{u,i}$ polynomials.

Example 2 *Consider the trust network of the previous example (see Figure 1). As before we have kept only the subscript for the user and dropped the subscript for the item as there is only a single item that we are considering.*

Initially, $c_1 = e_1 = 0$ *and* $c_2 = e_2 = 1$, $c_3 = e_3 = 1$, *and* $c_4 = e_4 = 1$.

Formula 3, gives rise to the following equations for nodes 1, 2, 3, and 4, respectively.

$$\begin{aligned} c_1 &= \epsilon \cdot (c_2 + c_3) \\ c_2 &= 1 + \epsilon \cdot (c_1 + c_3) \\ c_3 &= 1 + \epsilon \cdot c_4 \\ c_4 &= 1. \end{aligned}$$

After the first pass, we have

$$\begin{aligned} c_1 &= 0 + \epsilon \cdot (1 + 1) = 2\epsilon \\ c_2 &= 1 + \epsilon \cdot (0 + 1) = 1 + \epsilon \\ c_3 &= 1 + \epsilon \\ c_4 &= 1, \end{aligned}$$

whereas, after the second pass, we have

$$\begin{aligned}
c_1 &= 0 + \epsilon \cdot (1 + \epsilon + 1 + \epsilon) = 2\epsilon + 2\epsilon^2 \\
c_2 &= 1 + \epsilon \cdot (2\epsilon + 1 + \epsilon) = 1 + \epsilon + 3\epsilon^2 \\
c_3 &= 1 + \epsilon \\
c_4 &= 1.
\end{aligned}$$

Finally, the normalized polynomials are obtained by taking the Hadamard quotients of the q_u *polynomials with the corresponding* c_u *polynomials. We have*

$$\begin{aligned}
q_1 /\!\!/ c_1 &= (7\epsilon + 9\epsilon^2) /\!\!/ (2\epsilon + 2\epsilon^2) = 3.5\epsilon + 4.5\epsilon^2 \\
q_2 /\!\!/ c_2 &= (3 + 4\epsilon + 12\epsilon^2) /\!\!/ (1 + \epsilon + 3\epsilon^2) = 3 + 4\epsilon + 4\epsilon^2 \\
q_3 /\!\!/ c_3 &= (4 + 5\epsilon) /\!\!/ (1 + \epsilon) = 4 + 5\epsilon \\
q_4 /\!\!/ c_4 &= 5 /\!\!/ 1 = 5.
\end{aligned}$$

■

If only a single real number is needed as output for the q-opinion of a user u on some item i, then the coefficient of the smallest ϵ power in the normalized polynomial $q_{u,i} /\!\!/ c_{u,i}$ can be considered. For example, if this polynomial is $3.5\epsilon + 4.5\epsilon^2$, then 3.5 is produced, whereas if the polynomial is $3\epsilon^2 + 2\epsilon^3$, then 3 is produced.

We note that in practice the polynomials are more useful as a whole because as such they enable a more fine grained comparison of recommendations for different items than just the approximation by their most important coefficient. Thus, we can substitute directly these polynomials for $u_k[i]$'s in formula (1) and then perform symbolic computations with polynomials generating in the end recommendations which are polynomials as opposed to single numbers. However, we use the above most-important coefficient approximation in order to be able to compare with other methods which only produce single numerical predictions for user/item ratings.

7 Evaluation

In this section we present the evaluation of our system. We have used the same dataset that Massa and Avesani have collected for testing their trust-based recommendation methods (http://www.trustlet.org).

7.1 Dataset

The dataset has been crawled from the *Epinions.com* Web site. Epinions is a site where the users can write and read reviews about items such as movies, cars, books, etc. The users also assign numeric ratings to the items of interest. These ratings are in a scale from 1 (min) to 5 (max).

What is interesting about *Epinions.com* is that the users can create their web of trust which is *accessible*, and thus, the dataset we consider contains a trust network that we can use. To the best of our knowledge, this is the only dataset which provides both user ratings, as well as a trust network.

The dataset has the following specific characteristics:

1. There are
   (a) 45,819 users,
   (b) 139,738 items,
   (c) 664,823 ratings, and
   (d) 487,183 trust statements.
2. The number of customers who trust someone is 33,961.
3. The number of users who have rated at least one item is 40,163.

7.2 Evaluation Metrics

There are several types of metrics that have been used to evaluate recommendation systems. One metric we use is the well known Mean Absolute Error (MAE). This is the ubiquitous metric in most of the works in recommender system and the easiest to interpret for measuring the effectiveness of prediction. MAE measures the deviation of predictions from the true user-specified ratings. The technique for calculating MAE is *leave-one-out*, i.e. the true user rating is hidden and the prediction is calculated.

For each rating-prediction pair (r, p), the absolute error $|r - p|$ is calculated. The procedure is repeated for all the rating-prediction pairs and the average is finally calculated. Formally,

$$\text{MAE} = \frac{\sum_{(r,p)\in P} |r - p|}{|P|}, \tag{4}$$

where P is the set of rating-prediction pairs and $|P|$ is the cardinality of this set.

If we qualify a rating-prediction pair as $(r_{u,i}, p_{u,i})$ to express by the subscripts the particular user and item for the pair, and also introduce a 0/1 variable $a_{u,i}$ which is 1 when the system is able to predict a value for user u and item i, and 0 otherwise, then formula 4 is equivalently reexpressed as

$$\text{MAE} = \frac{\sum_{u\in U} \sum_{i\in I_u} |r_{u,i} - p_{u,i}|}{\sum_{u\in U} \sum_{i\in I_u} 1}, \tag{5}$$

where U is the set of all users, whereas I_u is the set of items for which user u has provided a rating and the system was able to produce a prediction. The denumerator of the fraction can also be expressed as $\sum_{u\in U} |I_u|$.

Clearly, the lower the MAE, the more accurate is the system.

Another metric which is argued to be very important in the study of recommender systems (cf. Herlocker et. al.) is the *Coverage*. The coverage is the fraction of ratings, which after being hidden, the system is able to produce a prediction. This metric is important because the recommender systems are not always able to give recommendations for a given user and a given item.

We also measure the *rank accuracy* (RA) of the considered methods. The RA is the ability to generate an ordering of recommendations that matches how the user would have ordered his/her opinions (ratings) [3]. As a rank accuracy metric we consider the Spearman ρ (cf. [3]), which is computed as the Pearson correlation, except that ratings and predictions are first transformed into ranks.

Since the dataset we consider is significantly large, as proposed by Massa and Avesani in [7], we also study in detail the following segments or views on the data:

1. *Heavy raters*, who provided more than 10 ratings;

2. *Opinionated users*, who provided more than 4 ratings and whose standard deviation of ratings is greater than 1.5;

3. *Black sheep users*, who provided more than 4 ratings and for which the average distance of their rating on item i with respect to mean rating of item i is greater than 1;

4. *Cold start users*, who provided from 1 to 4 ratings;

5. *Controversial items*, which received ratings whose standard deviation is greater than 1.5.

6. *Niche items*, which received less than 5 ratings;

Evidently, the performance numbers obtained on these segments of the data provide better insights regarding the strengths of various methods.

7.3 Experimental Results

In this section we present our experimental results comparing our recommendation method, TCF[2], with the following methods.

1. *Mole Trust* (MT) introduced in [8]: the method is instantiated with *trust horizon* one, two, and three[3], and these instantiations are denoted by MT1, MT2, and MT3, respectively;

2. User-based collaborative filtering (*CF*) as descibed before, and

[2]**T**rust-based Enhanced **C**ollaborative **F**iltering

[3]The trust horizon is parameter of Mole Trust which specifies the depth of neighbors in the trust graph that are considered by the method.

Mean Absolute Error/Ratings Coverage							
Views	**Methods**						
	IB	CF	MT1	MT2	MT3	TCF	
						TCF1	TCF2
All	1.031 45.28%	0.876 51.24%	0.844 26.48%	0.852 57.64%	0.832 71.68%	0.711 78.26%	0.700 86.80%
Heavy raters	1.07 49.8%	0.873 56.89%	0.847 28.97%	0.849 61.96%	0.828 75.06%	0.700 86.52%	0.689 94.66%
Opin. Users	1.619 50.48%	1.138 52.17%	1.071 21.43%	1.142 54.19%	1.135 70.11%	1.029 77.91%	1.030 87.10%
Black sheep	1.435 57.49%	1.25 53.38%	1.19 18.99%	1.2 63.21%	1.26 69.95%	1.199 78.17%	1.201 87.11%
Cold users	1.197 12.52%	1.03 3.22%	0.76 6.57%	0.92 22.06%	0.89 41.73%	0.97 7.21%	0.93 10.53%
Contro. Items	1.469 45.37%	1.597 49.86%	1.495 26.23%	1.676 60.43%	1.831 100%	1.355 77.84%	1.431 86.43%
Niche items	1.031 2.68%	0.835 14.29%	0.744 8.91%	0.814 26.58%	0.825 43.38%	0.277 65.50%	0.314 80.61%

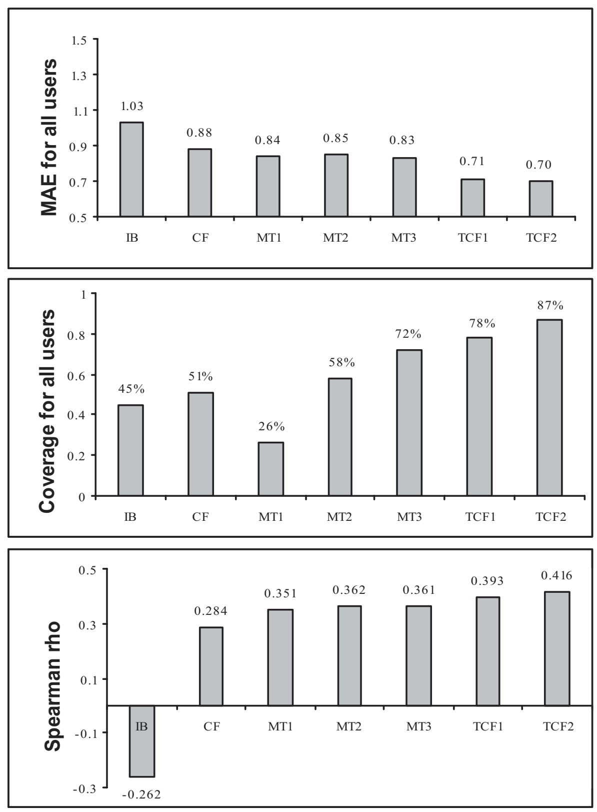

Figure 2: [Top] MAE and Coverage for the different methods. The performance numbers for our TCF method are given in the two rightmost columns of each table. [Bottom] Graphs for MAE, Coverage, and Spearman ρ when considering all users and items.

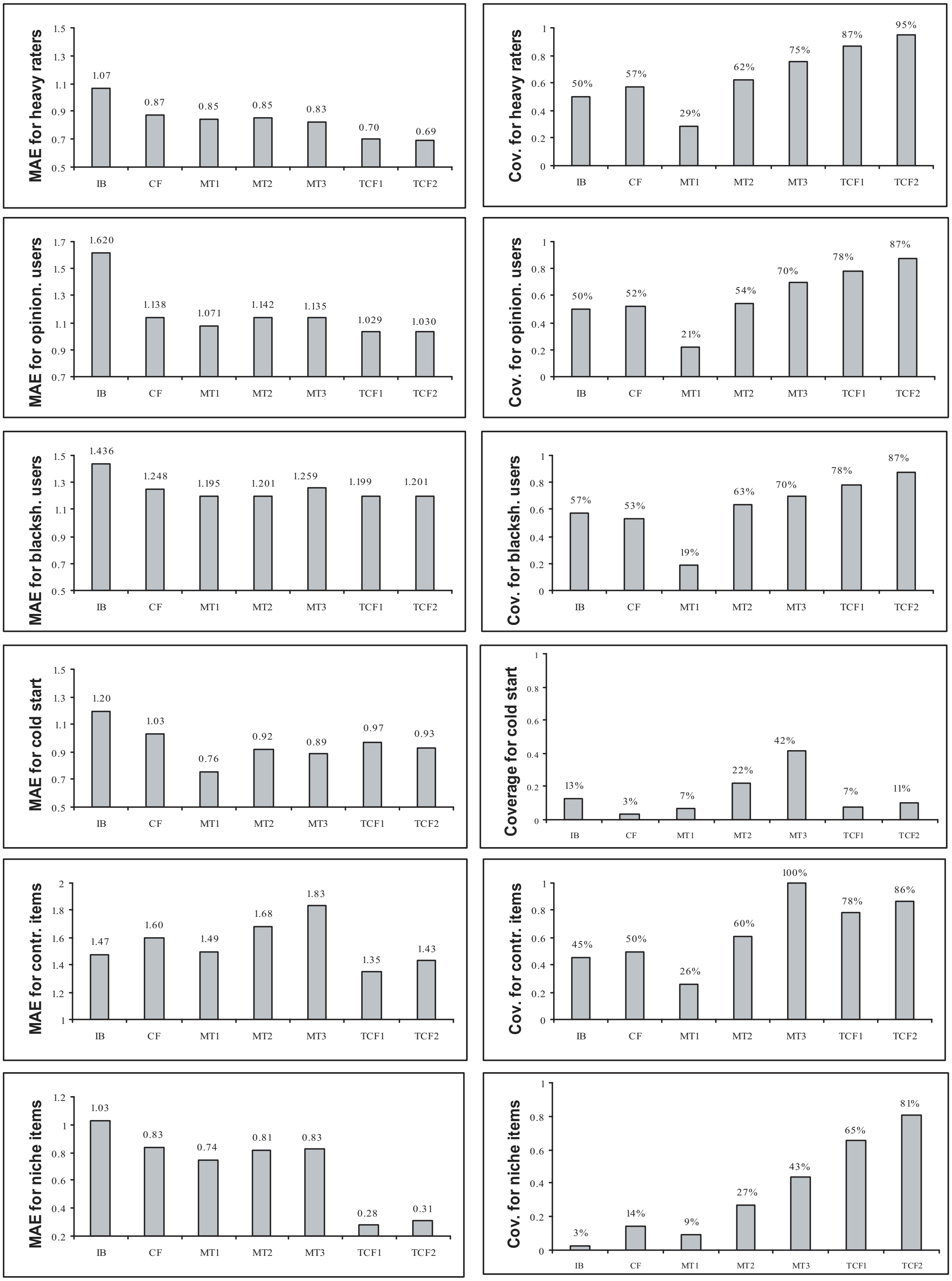

Figure 3: MAE and Coverage when considering the specified data segments.

3. Item-based (*IB*) collaborative filtering, which instead of similar users, finds similar items and then computes the recommendation as a weighted average of the ratings for the similar items (cf. [11]).

Specifically, we present results for our method when performing one and two iterations or passes through the data. We do not perform more iterations because the results reach a fixed point after the second iteration. The results are presented in figures 2 and 3.

In Figure 2 [Top] we show error and coverage numbers for the different methods we consider.

In Figure 2 [Bottom] we graph the performance numbers for the different methods when considering all the users and items. As it can be seen, our TCF method instantiations (TCF1 and TCF2) significantly outperform the other methods in terms accuracy, coverage, and Spearman ρ. Namely, our MAE values are 0.71 and 0.70 for TCF1 and TCF2, respectively, which not only are smaller than the MAE values for the other methods, but also are below the stated barrier of 0.73 in Herlocker et. al. paper [3]. The MAE values for our TCF method when performing two or three iterations are very close to each other. The coverage however is better for TCF when performing two iterations.

In Figure 3, we graph the performance numbers for the different methods when considering different views of users and items. Note that we consider these views when computing the error and coverage, not when computing the predictions. Our TCF method performs better for every view except for cold start users. MoleTrust with trust horizon 1 (MT1) gives a better accuracy by severely sacrificing the ratings coverage. As we go to a greater trust horizon for MoleTrust, and have MT2 and MT3, the coverage improves, but the accuracy decreases.

Interestingly, for cold start users, if we consider a naive recommender, which, for a given item i, always gives the simple average of ratings for i over all users, then the MAE is 0.86 and the coverage is 93%, thus being better than any other method for acceptable levels of coverage. Our conclusion with respect to cold start users is that one does not need to use elaborated methods, but just rely on the *most simple method* considering the aggregate opinion of all the users.

Another interesting feature of our method is that we can increase the coverage by performing more iterations without sacrificing accuracy. This in contrast to MoleTrust which suffers decreased accuracy (higher MAE) as the trust horizon increases. In MoleTrust, the trust horizon represents a tradeoff between accuracy and coverage, but this is not so for our TCF method.

8 Conclusions

We have presented a novel method for utilizing trust-based networks for enhancing collaborative filtering in recommendation systems. Our main idea is to use infinitesimal polynomials for representing the trustworthiness of users when aggregating their opinions for producing recommendations. When aggregating recommendations from trusted neighbors, these polynomials enforce our belief that the opinions of immediate (distance-one) neighbors are infinitely more important than the opinions of distance-two neigbors which in turn are more important than the opinions of distance-three neighbors, and so on. This way of treating opinions is a departure from previous approaches which consider a less drastic aggregation of trust neighbors' opinions.

The results justify our proposed method. We are the first to report a MAE of 0.7 on a large, real life, Epinions dataset, for which the next best MAE values are above 0.8 (for reasonable coverage). We also break the MAE barrier of 0.73, previously reported in the literature. We achieve this without sacrificing our ratings prediction coverage which is also better than that of the other methods.

We believe that our proposed framework based on infinitesimal numbers and polynomials opens the way for expressing special qualitative preferences in recommender systems, and it is an important step towards harnessing the power of social networks.

References

[1] P. Avesani, P. Massa, and R. Tiella. A trust-enhanced recommender system application: Moleskiing. In H. Haddad, L. M. Liebrock, A. Omicini, and R. L. Wainwright, editors, *SAC*, pages 1589–1593. ACM, 2005.

[2] J. A. Golbeck. *Computing and applying trust in web-based social networks*. PhD thesis, College Park, MD, USA, 2005. Chair-Hendler,, James.

[3] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.

[4] M. Jamali and M. Ester. *rustWalker*: a random walk model for combining trust-based and item-based recommendation. In J. F. E. IV, F. Fogelman-Soulié, P. A. Flach, and M. J. Zaki, editors, *KDD*, pages 397–406. ACM, 2009.

[5] J. H. Keisler. *Foundations of Infinitesimal Calculus*. On-line Edition, http://www.math.wisc.edu/∼keisler/foundations.html, 2007.

[6] P. Massa and P. Avesani. Trust-aware collaborative filtering for recommender systems. In R. Meersman and Z. Tari, editors,

CoopIS/DOA/ODBASE (1), volume 3290 of *Lecture Notes in Computer Science*, pages 492–508. Springer, 2004.

[7] P. Massa and P. Avesani. Trust-aware recommender systems. In J. A. Konstan, J. Riedl, and B. Smyth, editors, *RecSys*, pages 17–24. ACM, 2007.

[8] P. Massa and P. Avesani. Trust metrics on controversial users: Balancing between tyranny of the majority and echo chambers. *International Journal on Semantic Web and Information Systems*, 3(1):39–64, 2007.

[9] J. O'Donovan and B. Smyth. Trust in recommender systems. In R. S. Amant, J. Riedl, and A. Jameson, editors, *IUI*, pages 167–174. ACM, 2005.

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

[11] T. Segaran. *Programming Collective Intelligence*. O'Reilly, 2007.

[12] C.-N. Ziegler. *Towards Decentralized Recommender Systems*. Verlag Dr. Mller, Saarbrcken, Germany, May 2008.