Private Distributed Collaborative Filtering Using Estimated Concordance Measures

Neal Lathia
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
n.lathia@cs.ucl.ac.uk

Stephen Hailes
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
s.hailes@cs.ucl.ac.uk

Licia Capra
Dept. of Computer Science
University College London
London, WC1E 6BT, UK
l.capra@cs.ucl.ac.uk

ABSTRACT

Collaborative filtering has become an established method to measure users' similarity and to make predictions about their interests. However, prediction accuracy comes at the cost of user's privacy: in order to derive accurate similarity measures, users are required to share their rating history with each other. In this work we propose a new measure of similarity, which achieves comparable prediction accuracy to the Pearson correlation coefficient, and that can successfully be estimated without breaking users' privacy. This novel method works by estimating the number of concordant, discordant and tied pairs of ratings between two users with respect to a shared random set of ratings. In doing so, neither the items rated nor the ratings themselves are disclosed, thus achieving strictly-private collaborative filtering. The technique has been evaluated using the recently released Netflix prize dataset.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Security

Keywords

Correlation, Privacy, Recommender Systems

1. INTRODUCTION

Recommendation systems were designed in response to a well known problem: lack of time. People using the web, browsing e-commerce catalogs or news web sites, simply do not have the time to look through (and perhaps sample) the available items to find all the ones they like; they are confronted with the problem of information overload [1]. So instead of letting users get lost in the insurmountable quantity of information they have before them, many web sites present users with recommendations: by collecting *taste information* they create user *profiles*, and use them to make automatic predictions of user preferences. In recent years, collaborative filtering (CF) has established itself as the principle means of generating these recommendations. CF is a method of using the community's behavior to support that of each individual. An example can be taken from Last.fm [2]: if I like the blues musician John Lee Hooker, rather than browsing all the artists tagged "blues," I can focus on what people who also like John Lee Hooker listen to, and find that I may also enjoy Muddy Waters.

Recommender systems traditionally face two conflicting challenges: on the one hand, accuracy (i.e., finding a method to generate recommendations that will closely match the user's actual taste); on the other hand, scalability, since generating these recommendations requires a lot of computational power. As these methods move from centralised servers to distributed, and eventually mobile, platforms, a third issue, that is, *data privacy*, gains more importance than ever before, and adds a new layer of complication to the problem of collaborative filtering [3].

The problem may stem from various sources: on one hand lack of trust in centralised repositories (and their promise of keeping profile information private); or reluctance, in a distributed environment, to collaborate and interact with unknown neighbors. However, collaboration is the key to successful recommendations. Let us move the previous Last.fm example to a distributed, peer-to-peer, environment. If I were asking ten people what they think of Muddy Waters, to predict how much I will like him, then I would prefer to weight the responses I receive based on how similar the recommenders tastes are to my own. This idea is in line with the concept that a recommender's opinion will be more valuable to me if they are like me; someone who listens to the same music I do will provide more insight into how much I will enjoy Muddy Waters rather than, for example, someone who listens to techno. This means each pair of users needs to measure their similarity, or correlation, by comparing each other's profiles to their own. What if I do not want to share my personal tastes with the strangers, since I do not know how they will use this information? Unfortunately, current CF techniques require users to release their entire profile, a requirement that may not only discourage them from partaking in the CF process, but hurt the overall system by reducing the number of users willing to collaborate.

In order to achieve private collaborative filtering in a peer-to-peer environment, privacy should be clearly defined and methods of supporting varying degress of privacy should be

RecSys'07, October 19–20, 2007, Minneapolis, Minnesota, USA.
Copyright 2007 ACM 978-1-59593-730-8/07/0010 ...$5.00.

created. Current methods, described in Section 3, show that it is difficult to find user similarity without compromising their profiles. We propose a new way of computing the correlation between users, based on concordance, in Section 3.2. Our method aims to compute the proportion of agreement between pairs of users, without requiring them to disclose *any* part of their profile. Instead, two users who want to compute their similarity will only have to share a randomly generated set of ratings, and report to each other the number of concordant, discordant, and tied pairs that they share with it. The evaluation of this new method is three-fold: we first compare its prediction accuracy to the Pearson correlation coefficient (Section 3.2), then examine the root mean squared and mean absolute error (RMSE and MAE) measures of the correlation-estimation method on both real and synthetic data in Sections 5.1 and 5.2, lastly, we explore the price incurred on prediction accuracy by using estimated, rather than the actual, similarity measures.

2. PRIVACY

2.1 Definition

Before describing how to achieve privacy within CF, an accurate definition of what privacy means is required. In [4], Alan Westin gives the following definition:

> "Individuals, groups, or institutions have the right to control, edit, manage, and delete information about themselves and decide when, how, and to what extent that information is communicated to others."

The highlight of this definition is that privacy is about control, an idea that resounds in legislation on privacy and use of personal data, such as in the 95/46/EC European directive[1]. Rather than being about withholding or concealing information, supporting privacy means allowing users to exercise full authority over all components and the aggregate of the information that constitutes their profile, and constructing means to allow for collaboration between users when full privacy is desired.

2.2 Private and Public Data

A user in a collaborative filtering environment will generate a set of ratings about some items, such as movies or songs. The set corresponds to the user's profile, and will be used as a means for providing recommendations to neighboring users, or other users who are also creating rating sets in the same environment. With regards to this definition of user's profile, we call private information the following:

- The fact that user a has rated item i.
- A rating $r_{a,i}$ by user a for item i.
- The mean rating $\bar{r}_a$ for user a over all items it has rated.
- The total number of items that n that user a has rated.

In decentralised environments, such as the ones proposed in [5], users will store this information on their local devices. In the same setting, we define public information as:

- The total number of items N that can be rated in the current set. As items are added and removed, all users will be informed of the new size of N.
- The difference between a rating for item i and the user mean rating, $(r_{a,i} - \bar{r}_a)$.

The second value is what neighbors will use when they are creating a predicted value for an unrated item. Although it shows whether item i was rated above or below the mean rating by user a, given this value, it is hard to find $r_{a,i}$ and $\bar{r}_a$. However, by relying on this value, the system becomes vulnerable to different attacks, which will be discussed in Section 7.

3. COLLABORATIVE FILTERING

3.1 Current Methods: No Privacy

The current methods of conducting neighborhood-based collaborative filtering require correlation coefficients to be calculated between every pair of users. These coefficients roughly correspond to how much users will trust, and therefore how much they will weight, each other when aggregating many recommendations together. The most widely cited method to compute such coefficients is the Pearson correlation coefficient (equation 1) [6], [7] although variations and other measures exist [8].

$$w_{a,b} = \frac{\Sigma_{i=1}^{N}(r_{a,i} - \bar{r}_a)(r_{b,i} - \bar{r}_b)}{\sqrt{\Sigma_{i=1}^{N}(r_{a,i} - \bar{r}_a)^2 \Sigma_{i=1}^{N}(r_{b,i} - \bar{r}_b)^2}} \tag{1}$$

This method finds a similarity measure based on the intersection of the user ratings (i.e. the items that have been rated by both users); unrated items, or those rated by only one of the two sides, are not considered. In other words, the input vectors to these measures are of size $|r_a| \cap_i |r_b|$.

The measures are then used by user a as weights when generating a predicted rating $p_{a,i}$ for an item. If there are no neighbors who can provide a recommendation, then the returned predicted value is the current user's mean, a small heuristic that neither punishes nor rewards the predicted ratings of items for which there is no recommendation. Otherwise, recommendations from many neighbors are combined together by calculating a weighted average of the deviations from each neighbor's mean [6], as follows:

$$p_{a,i} = \bar{r}_a + \frac{\Sigma_{u=1}^{M}(r_{u,i} - \bar{r}_u)w_{a,u}}{\Sigma_{u=1}^{M} w_{a,u}} \tag{2}$$

Where M is the number of recommendations received. Ideally, we would like to be able to use the above measures with the defined privacy requirements. To do so would entail decomposing the similarity equations into two parts, and defining a new operation to compose the two halves together, in such a way that the two rating sets would never have to be directly compared to one another. In other words, two users, wishing to find their similarity, would perform an operation on their own rating set and, after reporting these values to each other, they would be able to find their actual correlation by composing the two values. However, the current method to generate these coefficients are such that it is difficult to compute these coefficients without requiring users to share their rating history with each other.

[1] http://en.wikipedia.org/wiki/Directive_95/46/EC_on_the_protection_of_personal_data

Table 1: Prediction Error using Somers and Pearson coefficients

	RMSE		MAE		Uncovered	
Neighbors	Pearson	Somers	Pearson	Somers	Pearson	Somers
1	1.405	1.171	1.096	0.895	0.971	0.930
10	1.298	1.132	1.001	0.870	0.743	0.696
30	1.200	1.113	0.930	0.857	0.485	0.537
50	1.153	1.098	0.893	0.847	0.391	0.461
100	1.110	1.088	0.859	0.839	0.291	0.361
500	1.066	1.066	0.825	0.825	0.140	0.150
999	1.067	1.063	0.826	0.824	0.125	0.136

3.2 Concordance

An alternative way of defining the correlation between two users within a single rating context (e.g. movies), which can be used to achieve the data privacy described above, is using the proportion of concordant, discordant, and tied pairs of ratings that users share. If we define the difference between a user a's rating for an item i and a's mean rating as $f_{a,i} = r_{a,i} - \bar{r}_a$, then given a pair of ratings, $r_{a,i}$ and $r_{b,i}$, by different users a and b for the same item i, such ratings would be:

- Concordant, if
  - $f_{a,i} > 0$ and $f_{b,i} > 0$, or
  - $f_{a,i} < 0$ and $f_{b,i} < 0$
- Discordant, if
  - $f_{a,i} > 0$ and $f_{b,i} < 0$, or
  - $f_{a,i} < 0$ and $f_{b,i} > 0$
- Tied, if
  - $f_{a,i} = 0$ or $f_{b,i} = 0$,
  - $r_{a,i}$ does not exist (the item is unrated), or
  - $r_{b,i}$ does not exist

Given a total number N of items, and the numbers of concordant (C), discordant (D), and tied (T) ratings between two users a and b we can define a new *measure of association* between the two users. This measure is called Somers' d, and is defined as follows [9]:

$$d_{a,b} = \frac{C - D}{N - T} \qquad (3)$$

This value can be interpreted as finding the proportion of how much users tend to agree with each other, and can be used a a weight when aggregating recommendations, exactly as $w_{a,b}$ or $cos(a, b)$.

Before moving on to show how this coefficient allows for private collaborative filtering, it is useful to put it into perspective by showing how it compares, in terms of prediction accuracy, to the more commonly seen coefficients, namely the Pearson correlation coefficient. We selected the Pearson correlation coefficient as a base line for comparison (instead of, say, vector similarity) since it To do this we used a subset of the recently released Netflix competition dataset [10]. The entire dataset is reported to contain 100 million ratings, made by 480,000 users on 17,770 movies, making it among the largest real user-rating datasets available to the public. The subset that we used is a set of 999 customers who had rated between 100 and 500 movies each. The subset was then split into training and prediction sets, by ordering each user's ratings chronologically, and splitting at the midpoint. We chose this subset with the goal of using it to compare the two algorithms on a plausible dataset, that is, a dataset that is neither too sparse (only a couple of ratings per user) nor too dense (there are a handful of customers who have rated all of the movies). In other words, we did not test the "extreme" behaviors of the algorithms, but selected a subset that displays the data sparsity feature that is a common characteristic to most rating datasets. This decision removed the need for us to consider modified versions of the Pearson coefficient that address the inaccuracy of measuring similarity when users have very small profiles, such as the $n/50$ significance-weighting method [6]. It also allows us to differentiate between the problem of estimating profile similarity between users who have constructed a profile, and the separate problem of bootstrapping similarity measures between users who have provided very little information to the system.

We first used the training set to generate both the Pearson correlation and Somers' d coefficients between every pair of users. We then used these coefficients as weights in the prediction method (equation 2) with varying numbers of neighbors for each set of coefficients. The results are reported in both root mean squared error (RMSE) and mean absolute error (MAE) between the actual and predicted ratings, as shown in Table 1. We also include another error measure, that is, the proportion of uncovered items. This corresponds to the number of movies for which no recommendation could be found (within the given set of neighbors) divided by the number of movies in the test set. These are the two most common error measures for collaborative filtering algorithms; however, in these experiments, we measured them separately. In other words, the RMSE and MAE measures were derived only on the predicted ratings that the system could generate, and answers the question "when the system can generate recommendations, how accurate are they?" Similarly, the coverage measure looks to answer the question "what proportion of the dataset can the system generate predictions for?"

One of the most notable differences between the two methods is that the Somers' d measure only looks for agreement within the ratings, thus reducing the rating scale to 3-values (like/dislike/no opinion). This would seem to imply a loss of information with respect to the Pearson coefficient, where ratings are weighted by how much they deviate from the mean. However, as Table 1 demonstrates, the error computed using the two methods is very similar, thus suggesting no meaningful loss of information, or perhaps counting the number of tied pairs compensates for what was disregarded. The operations involved in computing the Somers' d coefficient are however computationally much clearer than those required by the Pearson method (which include power and square root); the Somers' d method does not entail additional overhead.

4. USING CONCORDANCE TO ACHIEVE PRIVACY

4.1 Transitivity of Concordance

The interesting property about concordance measures is that a pair of ratings $r_{a,i}$ and $r_{b,i}$ can be categorized as either concordant, discordant, or tied, without ever directly comparing them, but rather comparing them to a third rating $r_{c,i}$.

It is relatively simple to show (using the definitions in given in Section 3.2) that if both relationships between ($r_{a,i}$, $r_{c,i}$) and ($r_{b,i}$, $r_{c,i}$) are either concordant or discordant, then the relationship ($r_{a,i}$, $r_{b,i}$) is concordant. Similarly, if one is concordant and the other discordant, then ($r_{a,i}$, $r_{b,i}$) must be discordant. Lastly, if we impose the restriction that the value $r_{c,i}$ will not equal the mean $\bar{r}_c$ or be "unrated," then if one of the two relationships is tied, ($r_{a,i}$, $r_{b,i}$) will also be tied.

We can show how this works with a small example. Suppose a, who has mean 3.5, rates an item 4 stars, and b, who has mean 2, rates the same item 3 stars. The differences between the ratings and the means, $f_{a,i}$ and $f_{b,i}$ are both positive: it is clear that these two ratings are concordant with each other. Now pick a random rating $r_{r,i}$, say 1, and suppose the mean of the random set is 2.5. When a compares this rating to its own, it will conclude that the two are discordant, and b will similarly achieve the same result. If a tells b that it is discordant with $r_{r,i}$, then, by the definition of discordant pairs, there are two possible cases that could have arisen:

- $f_{a,i} < 0$, and $f_{r,i} > 0$, or
- $f_{a,i} > 0$, and $f_{r,i} < 0$, or

Since b also has access to $r_{r,i}$, it knows that $f_{r,i} < 0$, and for a to have chosen discordant means that the second case must be true, $f_{a,i} > 0$, and so a's rating is concordant to its own.

4.2 Concordance, with Privacy

We use the indirect comparisons to introduce full privacy into the collaborative filtering process. Given two users, a and b, our method computes a privacy-preserving similarity measure as follows:

1. Generate r_r, a set of ratings of size N such that no rating $r_{r,i}$ is equal to the set's mean, $\bar{r}_r$, and each rating is a random number that is uniformly distributed on the rating scale.
2. a finds and reports:
    - C_{ar}, the number of concordant pairs it shares with the random set,
    - D_{ar}, the number of discordant pairs it shares with the random set, and
    - T_{ar}, the number of tied pairs.
3. Similarly, b finds and reports the values C_{br}, D_{br}, and T_{br} to a.
4. Each pair of values is used to find an upper and lower bound on the actual, unknown values C_{ab}, D_{ab}, T_{ab}.
5. The bounds are then used to generate an estimate of d_{ab}.

In order to precisely define how the lower and upper bounds of Point 4 are computed, we consider each set in more detail.

4.2.1 Tied Pairs

If there were only 2 items, x and y, that could be rated, and both a and b found that they are tied on 1 of those pairs with r, then they could both be tied on the same item x, or a could be tied on x and b tied on y. This implies that the lower bound to the number of tied items is $max(T_{ar}, T_{br})$. The upper bound is $(T_{ar} + T_{br})$, unless this sum is greater than N, in which case the upper bound is N. In this example, $1 \leq T_{ab} \leq 2$. More generally:

$$max(T_{ar}, T_{br}) \leq T_{ab} \leq (T_{ar} + T_{br}) \tag{4}$$

4.2.2 Concordant Pairs

Estimating the bounds on the number of concordant pairs is not as straightforward. It is helpful to first define two additional values, *minOverlap* and *maxOverlap*. Let us consider the same example as above: given that both users report that they are tied on one item, then (although we do not know how many they are tied on with each other), the maximum possible overlap, that is, the maximum number of items they could agree upon, is the minimum of the two values:

$$maxOverlap(C_{ar}, C_{br}) = min(C_{ar}, C_{br}) \tag{5}$$

Consider a similar example, where there are 5 items that can be rated, and a reports having 2 concordant pairs with r, while b reports 4 concordant pairs. Again, we do not know which pairs are the concordant ones, but we know the number of pairs. From above, we deduce that the *maximum* overlap is 2, and this would be the case if both the items that a is concordant on are also concordant for b. So what if the contrary were true, and both a and b were concordant with r on different items- what would the *minimum* overlap be? There are a total of 5 items, and 4 of them have been labelled concordant by b. That leaves 1 item that a can rate concordant, but a has 2 concordant pairs. Therefore, there must be a minimum overlap of one pair between the two.

$$\begin{aligned} minOverlap(C_{ar}, C_{br}, N) = min(C_{ar}, C_{br}) - \\ (N - max(C_{ar}, C_{br})) \end{aligned} \tag{6}$$

Using these equations, we can define the lower bound on the number of concordant pairs as the minimum overlap between the two values. As described above, the properties of concordance transitivity show that two concordant pairs and two discordant pairs result in a concordant pair. Therefore, an upper bound to the total number of concordant pairs is the maximum overlap between the two values plus the maximum overlap between the number of discordant pairs. In other words:

$$minOverlap(C_{ar}, C_{br}) \leq C_{ab} \leq \\ maxOverlap(C_{ar}, C_{br}) + maxOverlap(D_{ar}, D_{br}) \quad (7)$$

4.2.3 Discordant Pairs

To define the bounds on the number of discordant pairs, we use another property of concordance measures. When counting the different numbers of pairs, every entry is labelled as either concordant, discordant, or tied, even if an item is unrated. This implies that the sum of the concordant, discordant, and tied pairs will equal the size N, or the total number of items that can be rated. The lower bound on the number of discordant pairs can therefore be defined as N minus the upper bounds for the concordant and tied pairs, $C_{ab}^{max}, T_{ab}^{max}$. Similarly, the upper bound is N minus the lower bounds for the number of concordant and tied pairs, $C_{ab}^{min}, T_{ab}^{min}$:

$$N - (C_{ab}^{max} + T_{ab}^{max}) \leq D_{ab} \leq N - (C_{ab}^{min} + T_{ab}^{min}) \quad (8)$$

4.2.4 Estimated Concordance

Now that we have the bounds, we can generate a prediction of the actual correlation by using the midpoints of each. If the midpoints for the number of shared concordant and discordant pairs are the same, then the resulting similarity will be 0, a result that (if widespread throughout the community) will lead to very poor coverage. To counteract this behavior, we weight the number of concordant pair midpoint higher than the discordant midpoint, thus favoring the degree to which two entities estimate they will agree rather than disagree.

$$predicted(d_{a,b}) = \frac{\bar{C}_{ab} - 0.5\bar{D}_{ab}}{N - \bar{T}_{ab}} \quad (9)$$

4.3 Privacy, with Concordance

How does this method achieve private collaborative filtering? Returning to the definition of private data, our goal was to of find a similarity measure without requiring users to release the information listed Section in 2.2. That is, what items they have rated and the ratings that constitute their profile.

Concordance-based measures of similarity are not concerned with the actual ratings that users give their items, but rather the relationship that these ratings have with the mean rating. If a particular user reported that $T_{ar} = 0$, or, in other words, that all of its ratings were either concordant or discordant with the random set, then we would know that the user has rated all the items. On the other hand, we would not be able to precisely match the concordant and discordant pairs to the actual items i, and, given that these values represent agreement relative to the mean rather than ratings, we would still have no information on how likely individual ratings are within that user's set.

Having imposed the requirement that no rating in the random set can equal the random set mean, we ensure that the random set will not interfere with the bounds on the number of tied pairs between the users, therefore not distorting the estimated value. A tied pair is also defined as one of two possibilities: when an item is unrated, or when the item rating equals the user mean. Therefore, two users, one who has not rated any items, and the other who has rated all items the same, can not be differentiated. The estimation method we proposed uses abstracted information derived from the profile and the shared random set. In other words, if the random set is only shared between the two users who want to estimate their similarity, then the resulting pair values will not be useful profiling information to any third party (that does not have the random set).

However, there are ways to leak profile information. Consider the worst-case scenario, where user a is fully concordant with the generated random set. Although the actual ratings remain unknown, this implies that the user has rated all available items, and the actual similarity can be derived using the random set. If, for example, the rating scale is 1 to 5 stars, the random set had mean $\bar{r}_r = 2.5$, and item i had rating, $r_{r,i} = 4$, then, to be concordant on this item, user a would have had to have rated item i above $\bar{r}_a$. On the other hand, had $r_{r,i} = 1$, we would know that $r_{a,i}$ is less than $\bar{r}_a$.

The user performing the estimation method with a would be able to build a profile of a using binary values. It would contain a +1 for every above-the-mean rating, and a -1 for every below-the-mean rating. The profile is fully revealed since this is enough information to be able to compute the actual Somers' d between the two. The same case would arise if user a were discordant with the entire random set.

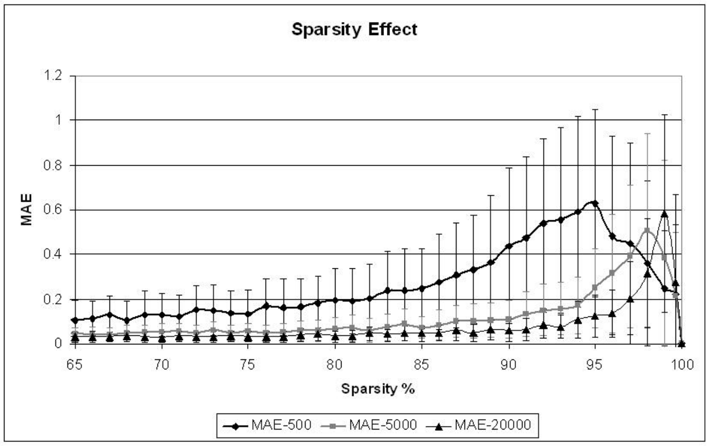
The probability of the worst-case happening is dependent on the set of items that can be rated, and is roughly equivalent to the probability that a particular user has both rated the entire set and the that the relationships generated in the random set will be a perfect match to the user's set. Given that in real life, the majority of available rating datasets are highly sparse, and recommendation systems were designed to confront large sets of items, the likelihood of this ever occurring remains very small. Extra security measures could also be enforced. For example, half of the random set could generated by a, and the other half by b, and if either of the two users finds that they are fully concordant on the half they generated, they could re-generate their half before sharing it with the other user. Therefore, even though a may still find that it is fully concordant with the half that b generated, a would ensure that the possibility of being fully concordant with the entire set could never happen.

This method is useful for estimating similarity with strict privacy enforced, but given that these systems will operate over time (as recommendations are exchanged between peers), it does not prevent peers from inferring each other's profiles based on the received recommendations: a user would have to communicate with a wide range of other users to minimise the possibility of this happening. It is also worth noting that if a group of users shared with each other their similarity to a target user, then some work could be done to infer the general tastes and opinions of the target. The problem of identifying colluding users, and evaluating the trustworthiness of recommenders in the community, deserves closer inspection and are left as matters of future study.

5. EVALUATION

Previous work [11] has reported that the results of many collaborative filtering techniques are sensitive to the dataset they operate on. Moreover, each dataset is characterised by different size and sparsity parameters, corresponding to the number of items that can be rated and the number of ratings that users have made. The aim of privacy-enhanced CF is to introduce novel methods that can operate across a wide range of datasets, not methods designed to suit a particular

Figure 1: MAE Estimation Error with Fixed Dataset Size



case. That is why, to test our method, we first ran experiments using three randomly generated sets, A, R, and B. The value of these simulations is that it allowed us to test our method over a wide range of size and sparsity inputs, even though the ratings themselves may not model actual human behavior. If we abstract away from the idea that a vector A represents a user profile, and instead consider it to be a set of numbers that will have a fixed, observable correlation with a different set B, then the aim of using simulated data is to verify how well this technique can estimate the *actual* correlation using the values reported in common with the middle set R. The details of the experiments and the results obtained can be found in Section 5.1. We then tested the accuracy of our method against a real dataset, and an analysis of the results can be found in Section 5.2.

5.1 Results with Random Data

We split the experiment on random data into two parts; in the first we kept the sparsity constant and varied the size. The experiment using fixed sparsity proportions was run as follows:

```
$Sparsity = x$
for $N = 1000$ to 20000 do
    error = double[100]
    for $i = 0$ to 100 do
        1. Generate $r_r$ such that:
               $\forall_{r,i}$ = random number uniformly distributed
               between 1 and 5
               $\forall r_{r,i}$, $r_{r,i} \neq \bar{r}_r$
        2. Generate $a_r$ and $b_r$, with size $N$ such that:
               Each rating is random, uniformly distributed
               between 1 and 5
               $x\%$ of the ratings are set to -1 (unrated)
        3. Compare $a_r$ and $r_r$ to find:
               $C_{ar}$, the number of concordant pairs
               $D_{ar}$, the number of discordant pairs
               $T_{ar}$, the number of tied pairs
        4. Compare $b_r$ with $r_r$ to find $C_{br}$, $D_{br}$, and $T_{br}$
        5. Compute
               $T_{min} = max(T_{ar}, T_{br})$
               $T_{max} = sum(T_{ar}, T_{br})$
               $C_{max} = maxOverlap(C_{ar}, C_{br})$
               $C_{min} = minOverlap(C_{ar}, C_{br})$
                   $+maxOverlap(D_{ar}, D_{br})$
               $D_{min} = N - (C_{max} + T_{max})$
               $D_{max} = N - (C_{min} + T_{min})$

               $actual(a,b) = somers(C_{ab}, D_{ab}, T_{ab}, N)$
               $predicted(a,b)$
               $= somers(\frac{C_{min}+C_{max}}{2}, \frac{D_{min}+D_{max}}{2}, \frac{T_{min}+T_{max}}{2}, N)$
               error[i] = $|predicted(a,b) - actual(a,b)|$
    end for
    $MAE = sum_j(error[])/100$
end for
```

In the second experiment we varied the sparsity over fixed size, by replacing the sparsity input above with a size definition, and changing the for loop to iterate over sparsity proportions. The purpose of these two experiments was to observe the behavior of the estimation process without having to combine both inputs. In each run of the experiments, we would generate A, R, and B according to the size and sparsity inputs, and then record both the actual and estimated similarity between A and B, to generate MAE measures of the estimation process. The results are plotted in Figure 2 and 1. The results using synthetic data are helpful in determining the applicability of our solution as a method of estimating similarities between datasets. They show that our method can work with varying degrees of error, but provide no insight into the effect of such error on the prediction accuracy when using the estimated coefficients, which can only be analysed by using real data.

5.2 Results with Real Data

After experimenting with synthetic data, we returned to the Netflix dataset to test our methodology on real data. Analysis of the subset of customers that we had selected showed that the proportion of unrated items was about 99%, in other words, the dataset is very sparse, and falls into the worst-performance area of the simulation results. For each pair of customers, we generated the actual concordance similarity, and then created a random set of ratings to find the predicted similarity.

Generating coefficients with small error values in the simulation is encouraging, but not sufficient to show that designing a means for preserving strict privacy is helpful to

Figure 2: MAE Estimation Error with Fixed Dataset Sparsity

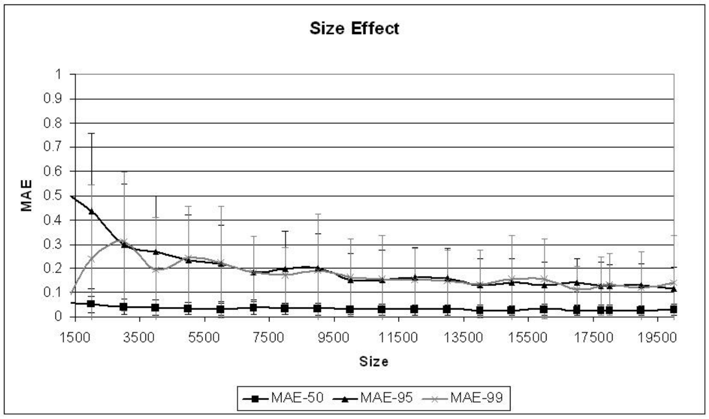


Table 2: Prediction Error Using Estimated and Actual Somers Coefficients

	RMSE		MAE		Uncovered	
Neighbors	Actual	Estimated	Actual	Estimated	Actual	Estimated
1	1.171	1.255	0.895	0.983	0.930	0.925
10	1.132	1.265	0.870	0.981	0.696	0.689
30	1.113	1.221	0.857	0.942	0.537	0.514
50	1.098	1.190	0.847	0.915	0.461	0.425
100	1.088	1.145	0.839	0.881	0.361	0.316
500	1.066	1.086	0.825	0.840	0.150	0.141
999	1.063	1.061	0.823	0.824	0.136	0.088

collaborative filtering; after all, the main task is to address information overload and maintaining privacy could be a barrier to the solution to this problem. The last experiment we ran was aimed at discovering to what extent the error in the predicted coefficients impacted the prediction accuracy. To do so, we ran the same prediction method used in section 3.2: using the same training and test sets as before, we generated predicted ratings using the estimated coefficients. The results are shown in Table 2, side by side with the previous results from using the actual Somers' coefficients.

The surprising result is that the effect, in terms of prediction accuracy and coverage, of using estimated coefficients is not as strong as may have been anticipated. The results, in some cases, are marginally better than using the actual coefficients, although the extent to which a 0.02 improvement will have on the system as a whole has yet to be fully explored. Just as collaborative filtering algorithms behave differently according to the the dataset they operate on, they are also subject to the parameters used in aggregating recommendations. In these experiments we implemented the top-k neighbors, without using any correlation threshold. However, it is relatively simple to see that if we opted to use the top-k *recommenders*, i.e. users who had rated the item in question, the coverage results would have been significantly improved.

6. PREVIOUS WORK

There has been a wide range of work done on preserving privacy in collaborative filtering environments. Amongst the first were anonymization techniques [12], [13]. As [7] discusses, these techniques are not appropriate for collaborative filtering since there is no guarantee of the quality of the dataset- allowing users to operate anonymously opens the system up to a number of attacks, which will quickly overcome the benefits of having a collaborative filtering system in place.

Polat and Du [7] describe a method of disguising user rating data by means of randomized perturbation techniques. The aim of their work is to modify the user ratings in such a way that a centralised data collector (which is performing the collaborative filtering) cannot derive the truthful user ratings, but can still use the relationships between the items in a meaningful way. This is a very interesting way of concealing private data, but the approach uses a more relaxed definition of privacy than we have defined above, since customers would still be divulging the information as to what items they have rated (even though it is disguised). This difference is highlighted by their conclusions, where they propose to increase the accuracy of their collaborative filtering system by divulging even more information, such as the user rating mean. This kind of information may not compromise the user's profile, but falls short of the Westin's definition of privacy by not allowing users to be in control of all information that is relevant to their profile.

Canny has also described methods of preserving privacy in [14] and [15]; these methods focus on creating a public aggregate of user data without violating any individual's privacy, based on distributed singular value decomposition of the user rating matrix. This entails creating communities of users, so that users can seek recommendations from the most appropriate group, thus opening up the system to encompass not only specific similarity-based recommendations (derived from personal preferences, such as movie recommendations), but also allowing heterogeneous recommendations, spanning many different contexts (a scenario that we do not consider here). The model, however, assumes that all users will participate in the distributed algorithm, and as collaborative filtering moves to a more ubiquitous environment, this assumption may not hold any longer. The model we have proposed based on concordance allows users to interact with selected neighbors, without requiring participation from them all.

Given that mainstream collaborative filtering systems run on centralised servers, the hazard becomes storing all usersŠ data in one location; instead, [5] proposes storing each client profile on the client side, and only transmitting similarity measures over the network, using methodologies that link the ideas of Polat and Du [7], which were designed with centralised servers in mind, and the peer-to-peer solution described by Canny [14].

7. CONCLUSION AND FUTURE WORK

Concordance is a new, promising measure of similarity between users, which allows for distributed collaborative filtering without requiring any profile release from individuals or contributions by all members of the community to function. The essence of concordance-based measures is that it is not the precise rating values, but the proportion by which two rating sets tend to agree, that is useful when comparing people. Our next step in evaluating the new similarity measure and the estimation process will be to test it against more datasets, each with different size and sparsity characteristics, to increase the validity of the results obtained with the simulation and Netflix dataset. In [9], a number of other concordance-based measures are discussed, that deserve closer inspection, both as alternatives to the well-established methods of collaborative filtering and potential candidates to support porting these systems to peer-to-peer environments; the first step in our future work involves a more comprehensive analysis of the performance of these coefficients, on a wider range of datasets and recommendation aggregation parameters.

There are a number of aspects to this method that are also open to improvement; one of these is how to cope with users who have incredibly sparse profiles, or are new to the system and have provided very little information. However, the basis of exchanging abstracted profile information in order to estimate profile similarity is already in place. It is very tempting to offer to improve this estimation method by relaxing the definition of privacy; however the focus of distributed recommendation system research should not only be accuracy but supporting means that encourage all users to participate, including those that are most concerned about their privacy.

No matter how strong the privacy level supported by a collaborative filtering system is, the security of user's profiles is still vulnerable, due to the nature of cooperation between users to generate recommendations. An attacker could build a model of a user's profile by means of a probing attack, requesting recommendations for every item. The information received would allow the attacker to have a good idea of the user's tastes, even though it would be based on values ranging from -1 to 1, which is a linear transformation of the actual rating values, 1 to 5 (for the Netflix dataset). Therefore, users will also require means of protecting their profile as they operate in the collaborative environment; evaluating their similarity to others without releasing their profile is a first step that precedes support for evaluating other user's behavior, to find those that are trustworthy recommenders.

8. REFERENCES

[1] J.B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the ACM Conference on Electronic Commerce*, 1999.

[2] Last.fm website. http://www.last.fm, 2007.

[3] S. K. Lam, D. Frankowski, and J. Riedl. Do you Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. In *Proceedings of the 2006 International Conference on Emerging Trends in Information and Communication Security (ETRICS)*, 2006.

[4] A. F. Westin. *Privacy and Freedom*. The Bodley Head Ltd, 1967.

[5] B. Miller, J.A. Konstan, and J. Riedl. Pocketlens: Toward a personal recommender system. In *ACM Transactions on Information Systems*, volume 22, July.

[6] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 230–237, 1999.

[7] H. Polat and W. Du. Privacy-Preserving Collaborative Filtering using Randomized Perturbation Techniques. In *The Third IEEE International Conference on Data Mining (ICDM'03)*, Melbourne, FL, November 2003.

[8] G. Linden, B. Smith, and Y. York. Amazon.com recommendations: Item-to-item collaborative filtering. In *IEEE Internet Computing*, pages 76–80, 2003.

[9] A. Agresti. *Analysis of Ordinal Categorical Data*. John Wiley and Sons, 1984.

[10] Netflix prize. http://www.netflixprize.com, 2007.

[11] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. In *ACM Transactions on Information Systems*, volume 22, pages 5–53. ACM Press, 2004.

[12] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. In *ACM Transactions on Information and System Security*, volume 1, pages 66–92. ACM Press, 1998.

[13] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[14] J. Canny. Collaborative filtering with privacy via factor analysis. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, New York, NY, USA, 2002. ACM Press.

[15] J. Canny. Collaborative filtering with privacy. In *IEEE Symposium on Security and Privacy*, 2002.