Collaborative Filtering Using Random Neighbours in Peer-to-Peer Networks

Arno Bakker
Department of Computer Science
Vrije Universiteit
De Boelelaan 1081a
Amsterdam, The Netherlands
arno@cs.vu.nl

Elth Ogston
Department of Computer Science
University of Warwick
Coventry CV4 7AL, United Kingdom
elth@dcs.warwick.ac.uk

Maarten van Steen
Department of Computer Science
Vrije Universiteit
De Boelelaan 1081a
Amsterdam, The Netherlands
steen@cs.vu.nl

ABSTRACT

Traditionally, collaborative filtering (CF) algorithms used for recommendation operate on complete knowledge. This makes these algorithms hard to employ in a decentralized context where not all users' ratings can be available at all locations. In this paper we investigate how the well-known neighbourhood-based CF algorithm by Herlocker et al. [5] operates on partial knowledge; that is, how many similar users does the algorithm actually need to produce good recommendations for a given user, and how similar must those users be. We show for the popular MovieLens 1,000,000 and Jester datasets that sufficiently good recommendations can be made based on the ratings of a neighbourhood consisting of a relatively small number of randomly selected users.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Performance evaluation (efficiency and effectiveness)*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems*

General Terms

Algorithms, Performance, Measurement

Keywords

Collaborative Filtering, Peer-to-Peer Networking, Recommender Systems, Metrics

1. INTRODUCTION

In the near future, every home will have a personal video recorder (PVR) connected to IP(TV)-based networks that will give its users access to more content than ever before. The overwhelming number of movies, shows and clips creates the need for a recommendation system that helps the user to decide what to watch. For scalability to hundreds of millions of users, this recommendation system should preferably run autonomously on the network of PVRs. However, traditionally, *collaborative filtering (CF) algorithms* [13] used for recommendation operate on complete knowledge, that is, they assume the ratings of all users are known in one location. This makes these algorithms hard to employ in the decentralized PVR context. Due to the size and cost of distributing this information, not all users' ratings can be available on all devices. Hence, in this paper we investigate how CF algorithms operate on partial knowledge; that is, how many similar users does an algorithm actually need to produce good recommendations for a given user, and how similar must those users be. In particular, we study a well-known neighbourhood-based CF algorithm by Herlocker et al. [5]. Here we present only our major results, our full investigation on CF using partial knowledge can be found in [1].

The main contribution of this paper is to show for the well-known MovieLens 1,000,000 and Jester datasets [17, 4] that the CF algorithm can make sufficiently good recommendations based on the ratings of a neighbourhood consisting of a relatively small number of random users. We believe this to be an important result, as it implies that decentralized recommendation can be implemented using very light-weight peer-discovery protocols, rather than complex protocols that attempt to find the best-matching neighbours in the network. We showed this earlier for the MovieLens 100,000 data set [9].

The remainder of the paper is organized as follows. In Sec. 2 we present our system model and methodology. Section 3 describes our experiments studying the effects of the number and type of users on recommendation quality for the MovieLens 1,000,000 dataset. The Jester dataset is analysed in Sec. 4. We present related work in Sec. 5 and conclusions in Sec. 6.

2. SYSTEM MODEL AND METHODOLOGY

At an abstract level the problem of collaborative filtering considers a set of N users, $U = \{u_1, \ldots, u_N\}$ and a set of M items $X = \{x_1, \ldots, x_M\}$. Each user provides ratings, taken from a set of possible values, V, the rating scale, for a subset of the items in X. These ratings form an $N \times M$ user-item matrix, R, where the entry $r_{i,j}$ is the rating of user u_i for item x_j, or empty if that rating is unknown. The basic recommendation task is to predict a rating value for a given empty element $r_{i,j}$ based on the known values in R. This is done by means of a prediction function, f, where $f(R, i, j) \mapsto V$.

In our system model each user has a personal networked video recorder by which he or she rates content. These personal devices can exchange gathered ratings with the devices of other users via the network, and use them to make personal predictions to their respective users. As the network grows, it becomes infeasible to distribute all ratings, i.e. the full matrix R, to all recorders. The prediction function f in the video recorder for user u_i must therefore base its predictions on a submatrix of R denoted R_i. In this paper, this submatrix R_i will consist of u_i's own ratings and the ratings of a specific set of c other users he received ratings from, called u_i's *peer group*. We refer to a PVR storing a particular user's ratings as a *node* in the system.

As prediction function f we use a function that was designed by Herlocker et al. [5, 11] to optimize user-based prediction accuracy, which we refer to as f_1. The f_1 function creates a prediction for a new item x_j for user u_i as follows. It first selects from the submatrix R_i supplied by the peer group the ratings of the z users most similar to u_i *that have rated item* x_j. Note that these z users may not be the Top-z most similar users, as some of those may not have rated x_j. f_1 then determines the relative importance of the ratings chosen by assigning weights to the selected input ratings, based on user similarity. To calculate the similarity between users u_i and u_k it uses Pearson's correlation using significance weighting [5], denoted $d_{i,k}$. The significance weighting is controlled by two parameters, *minCommonItems* and *maxCommonItems*. Finally, from the weighted input it calculates a new prediction $r_{i,j}$ for the given user and item.

There are three things to observe in this model. First, there are two mechanisms for selecting which users' ratings are used in the prediction. The mechanism that constructs the peer group, which we call *network user preselection* and the selection of users from the submatrix R_i by prediction function f, which we call *local user selection.* We are particularly interested in the relative importance of these mechanisms: (1) how many users must the network user preselection select and how similar must they be, and (2) can the local user selection perhaps compensate for less strict network preselection?

The reason we are interested is because there are different costs associated with each mechanism. To construct a peer group that consists of the most similar users, a network protocol needs to be run that discovers these peers in the network of video recorders, and it needs to be run continuously to accommodate for changes in a user's taste. Such a protocol has high complexity in terms of the network resources (bandwidth, number of messages exchanged) used. If local user selection, which itself has zero cost in terms of network resources, is effective, it may be possible to run a less complex protocol to construct the peer group. We will show that this is indeed the case, and sufficiently good quality recommendations can be made with less strict requirements on the peer group.

The second observation to make is that in this model the quality of the predictions made depends on three factors. First, the size and, second, the composition of the user's peer group, as these determine the content of submatrix R_i. Thirdly, the quality depends on the prediction function f that is used, and how it selects and aggregates the information in R_i into a new prediction. The prediction function is fixed at f_1 in this paper, and we will focus on the two remaining factors which represent the amount and type of partial knowledge required for good recommendations, respectively.

Finally, note that peer-group size c and the parameter z that selects the number of similar users that rated the item x_j are related. If there are $c < z$ peers in the peer group, f_1 cannot select the desired number of neighbours. If $c > z$, the f_1 function will select at most z users from the peer group. However, this selection is not necessarily the Top-z most similar users in the group because some of the most similar users may have not rated the item. Hence, the recommendation process benefits from considering $c > z$ peers. The benefit ends once c is large enough such that f_1 will find z similar users that rated the item (or whichever is the maximum number of suitable raters that exist given u_i and x_j).

Methodology

To test the influence of the size and composition factors and the effects of the two mechanisms for user selection we organize the nodes into a virtual peer-to-peer overlay. Each node (PVR) is linked to a set of c other nodes via the peer group. We assume neighbourhood links are uni-directional, implying that a user can have only c neighbours in its peer group, but he or she can appear in the peer group of more than c other nodes.

In addition to varying the peer-group size c, we consider two contrasting peer-to-peer overlay topologies. In the first topology, the *random overlay*, the peer group consists of randomly selected nodes. In the second topology, the *best-neighbours overlay*, the peer group consists of the users to which the user is most similar, given the similarity function $d_{i,k}$ that compares the users' past ratings. Given that ratings from similar users should provide the best quality recommendations, these two topologies, that is, methods of network user preselection, represent a sub-optimal and a best-case scenario, respectively.

In the past, decentralized recommendation systems have constructed this best-neighbours overlay in various ways [3, 16, 10, 2, 12, 7, 14]. Generalizing, the nodes exchange their rating data and compute the similarity to the other peers using a similarity function. By remembering the best candidates so far, while continuing to exchange past ratings with other peers, it has been shown that each node will eventually fill its peer group with the nodes most similar to it. Our main point for this paper is that, in general, discovering these similar peers, in whatever way, is more expensive in terms of network usage than just finding some random peers. Hence, the best-neighbour overlay represents the heavy-weight solution that should be avoided and the random overlay the light-weight solution that is to be preferred.

In summary, in our model, computing a prediction $r_{i,j}$ consists of the following steps:

1. Construct an overlay that connects each node u_i to its peer group of c users; either random users or the users most similar to u_i according to $d_{i,k}$.

2. Create matrix R_i from connected peer group's ratings.

3. From R_i, select the set K ($|K| \leq z$) of users most similar to u_i that (a) have rated x_j and that (b) have rated at least *minCommonItems* of the same items.

4. Subsequently calculate the prediction using Herlocker et al.'s formulae (from [11]):

$$r_{i,j} = \bar{r}_i + \frac{\sum_{k=1}^{|K|}(r_{k,j} - \bar{r}_k) \cdot d_{i,k}}{\sum_{k=1}^{z} d_{i,k}} \tag{1}$$

$$d_{i,k} = \alpha \cdot \frac{\sum_{s=1}^{|S|}(r_{i,s} - \hat{r}_{i,S})(r_{k,s} - \hat{r}_{k,S})}{\sum_{s=1}^{|S|}(r_{i,s} - \hat{r}_{i,S})^2(r_{k,s} - \hat{r}_{k,S})^2} \tag{2}$$

$$S = X_i \cap X_k;\ \text{the set of items rated by both } u_i \text{ and } u_k \tag{3}$$

$$\alpha = \begin{cases} \dfrac{|S|}{maxCommonItems} & \text{if } |S| < maxCommonItems; \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

$$\bar{r}_i = \text{average rating of } u_i \text{ over all items he rated} \tag{5}$$

$$\hat{r}_{i,S} = \text{average rating of } u_i \text{ over all items in } S \tag{6}$$

3. MOVIELENS 1 MILLION EXPERIMENTS

In this section we analyse the MovieLens 1 million ratings dataset (ML1M) [17]. It consists of 1,000,209 ratings on a scale of 1 to 5 stars of 3593 movies by 6040 users. For evaluating the performance we partition this data into a training set and a test set. We use the same procedure as was used to create the `ua` training and test sets for the smaller MovieLens 100,000 (ML100K) dataset [11]. This is to ensure that these experiments are comparable to the ones we did earlier with this smaller dataset, described in [9].

This means the test set consists of 10 randomly chosen movies per user. The resulting remaining training set forms the matrix R, constituting the users' ratings used to populate the nodes in the overlay. Because ML1M has a different number of users, the procedure leads to a division where 94% of the ratings is used for training and just 6% for testing, as opposed to 90%, respectively, 10% for the `ua` division of ML100K. As a result, the outcomes in this section may be biased towards best-neighbours overlays, as more information is available about users.

Each experiment consists of constructing the two peer-to-peer overlays for a given peer-group size c using the training set. Each node will then attempt to predict the rating its user u_i would give to the 10 withheld items based on its R_i matrix. The resulting predictions are compared to the 60400 actual ratings in the test set using (initially) the *mean absolute error (MAE)* metric [15]. Associated with MAE is the *coverage* metric which measures what fraction of the predictions attempted actually returned a result. Predictions for user u_i and movie x_j may fail because, for example, none of the user's neighbours actually rated x_j.

The different parameters of the similarity function $d_{i,k}$ and prediction function f_1 for all experiments are set following Herlocker et al.'s conclusions [5] for the MovieLens 100K dataset, which we confirmed. Using these parameters again allows comparison with our earlier work. We set the significance weighting parameters of $d_{i,k}$ to *minCommonItems*=2 and *maxCommonItems*=100. Negative correlations are not considered. For the parameter z, the number of users to select using local user selection, we use 60. These parameters resulted in the lowest MAE for the MovieLens 100K `ua` sets when using complete knowledge. The experiments are conducted using the CoFE collaborative filtering engine [11].

Figure 1(a) shows the MAE performance of f_1 on a best-neighbours and random overlay on the ML1M dataset. In general, the random overlay performs worse than the best-neighbours one. The reason is that with a random overlay, f_1 bases predictions only on the more similar users in the random input set, i.e., uses only local user selection. For small group sizes this results in very little data with which to make predictions, but it is very effective for larger group sizes. Doing network user preselection by using a best-neighbours overlay provides f_1 with higher quality input thus improving recommendation, especially for smaller groups.

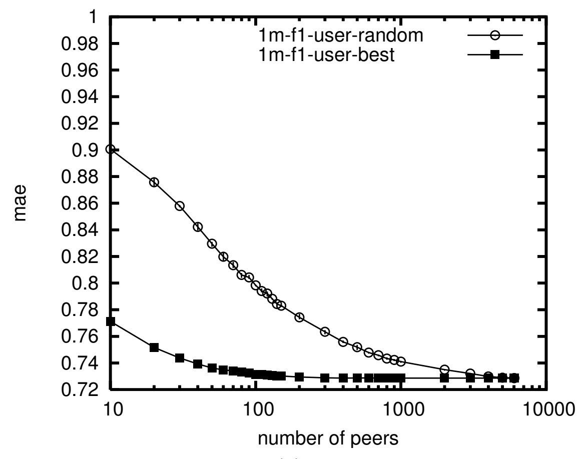
Looking in more detail, however, several important observations can be made. The first observation to make is that knowing the ratings of 3% of the user population yields practically the same results as knowing the whole population. In particular, the performance of both overlays for a group size of 200 is at most 0.05 stars worse on average than with the maximum group size of 6039.

The second (even more significant) observation is that the difference between using a best-neighbours overlay and a random overlay is also very small for small group sizes. The absolute MAE difference between 200 best or random neighbours is just 0.04 stars. In other words, selecting the 200 best out of 6039 candidates or 200 random ones has little influence on prediction quality.

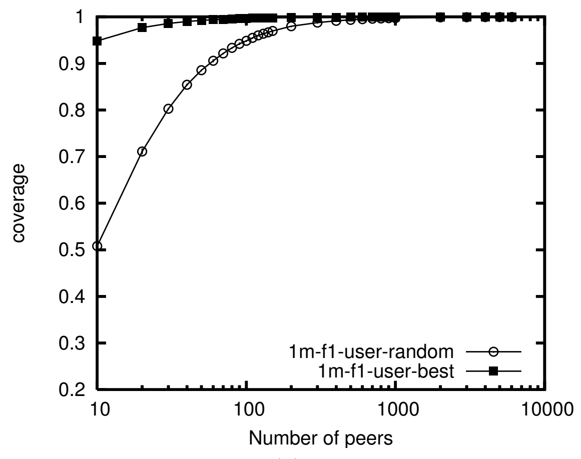
These surprising results are not due to low coverage. MAE is calculated over just the made predictions, which would benefit an algorithm that made just a few, but correct predictions. However, coverage is good at 90% or more for group sizes over 100 for all algorithms, as shown in Fig. 1(b). This behavior is also not explained by the value of the parameter z in these experiments. As in the ML100K experiments, z was set to 60, which means that even when group sizes are large, the algorithm will use the opinion of just 60 users. This parameter could therefore prohibit performance improvements at larger group sizes. This is not the case, however, as rerunning the experiment with $z = 120$ and up to 1000 peers gave hardly any improvement in terms of MAE, or precision and recall, as described in [1].

So the MAE results indicate that even relatively small groups of randomly chosen neighbours produce sufficiently good recommendations. In other words, it appears that local user selection from a small random sample of the user population are a match for network user preselection, as already shown for the smaller ML100K dataset in [9].

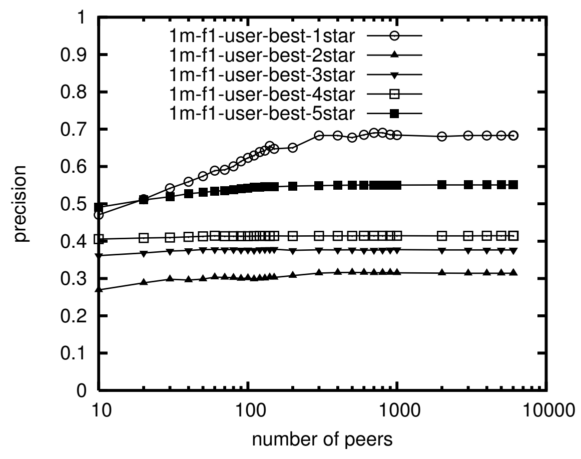
Unfortunately, the MAE metric has several problems [6]. First, small differences in MAE can indicate huge differences in recommendation quality. Second, it considers errors in any part of the ratings scale to be equal. However, errors at the extremes of the scale are more important than errors elsewhere, as users are "most interested in suggestions of items they would love or hate, not of items about which they would be ambivalent" [15]. To more accurately measure how the different overlays affect the quality of recommendation for these extremes we turn to the standard information-



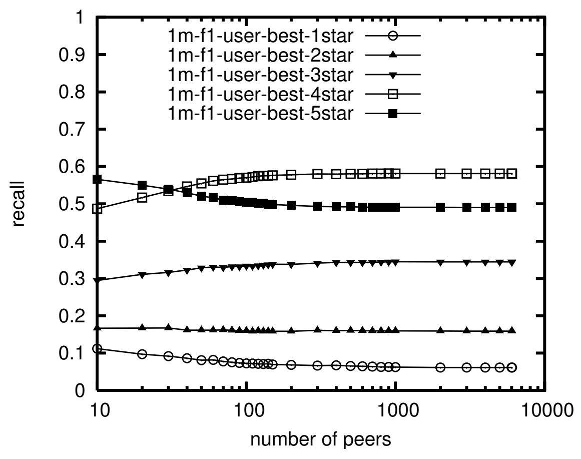
(a)



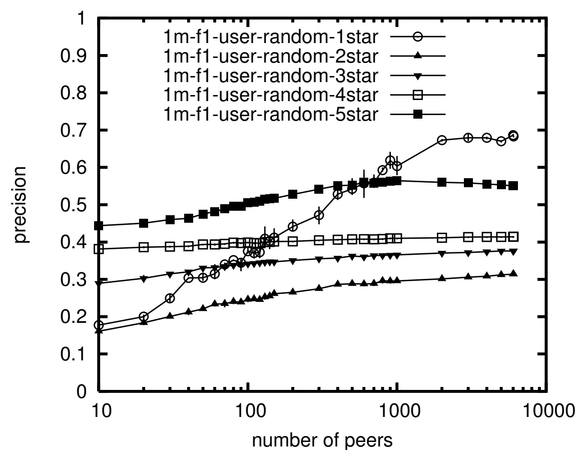
(b)



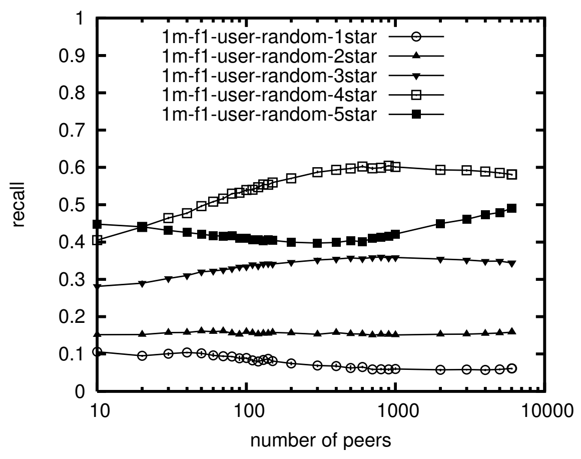
(c)



(d)



(e)



(f)

Figure 1: Measurements for MovieLens 1,000,000 with $z = 60$. For random peers the values are averaged over three runs. The vertical errorbars show the minimum and maximum value obtained in these three runs. The x-axis is on a logarithmic scale. (a) Mean Absolute Error (note that the y-axis starts at 0.72) (b) Coverage (note that the y-axis starts at 0.2) (c) The 1–5 star precision of f_1 for differing numbers of similar peers. (d) The associated 1–5 star recall. (e) The 1–5 star precision of f_1 for differing numbers of random peers. (f) The associated 1–5 star recall.

retrieval metrics *precision* and *recall*.

Visualizing Recommendation Behaviour

We employ the standard information-retrieval metrics precision and recall as follows. For the user-item pairs in the test set, we separate the list of returned predictions P, according to prediction value, into the sublists P_{1^*}, P_{2^*}, P_{3^*}, P_{4^*} and P_{5^*}. We also divide the actual ratings of the test set in a similar manner into A_{1^*}, A_{2^*}, A_{3^*}, A_{4^*} and A_{5^*}. Thus, P_{5^*}, for instance, contains all of the five-star predictions ($r^*_{i_k j_k} = 5$) and A_{5^*} contains all the actual five-star ratings in the test set ($r_{i_k j_k} = 5$). The user-item pairs (u_{i_k}, x_{j_k}) that correspond to the predictions and ratings in P_{5^*} and A_{5^*} can be viewed as a selected-items set S_{5^*} and a relevant-items set T_{5^*}, respectively, for the query "find all five-star movies for each user". This allows us to calculate precision and recall per rating value. Note that precision and recall are computed only over the predictions that could actually be made.

We use these new metrics to analyze the performance of the best-neighbours overlay in Fig. 1(c,d). As with MAE, precision and recall values vary little beyond group sizes of 200 neighbours, only one-star precision still rises 3 percentage points. Figure 1(e,f) shows precision and recall for the random overlay. Again there is little change as the group sizes grow. The exceptions are one-star precision that continues to increase by 23 percentage points, and one-star recall that grows 10 percentage points.

Comparing the two overlays we see that they have similar five-star precision. For the other extreme, one-star ratings, the best-neighbours overlay yields better precision: 17 percentage points on average, at most 31 for a peer group of 20. It also has slightly better two and three star precision (3 resp. 4 percentage points on average). Recall for the five stars extreme is better, up to 12 percentage points and 8 percentage points on average. Recall for four stars is on average 2 percentage points worse, at most 8 for the smallest groups.

In sum, the best-neighbours overlay has the best recommendation quality, especially at the extremes of the scale, but the results for the random overlay are not considerably worse, as MAE already indicated. And, as noted in the beginning, the experiment may be biased towards best-neighbours overlays. Hence, we conclude that using the ratings from a small group of random users to make predictions already leads to sufficiently good quality for this dataset.

4. JESTER EXPERIMENTS

The Jester dataset is the largest dataset we analysed and consists of 4,136,360 ratings of 100 jokes by 73,421 users on a continuous scale of -10 to 10 [4]. So apart from the domain, this dataset differs in most aspects from the two MovieLens datasets. First, the dataset is much less sparse, only 44% of the user-item space is not rated, as opposed to approx. 95% for MovieLens. Second, the number of items is much smaller (100 vs. 1682 or 3593 for MovieLens). Third, the number of users is much larger with 73,421 as opposed to 943 or 6040. Finally, the rating scale is much more fine-grained than just five discrete values [4]. We refer to Jester ratings as the number of *smiles* assigned to a joke.

We split this dataset into a training and test set as before. To achieve the same approximate division as in the previous experiments, we withhold 6 ratings as opposed to 10 in the MovieLens datasets, which results in 89% of the ratings being used for training and 11% for testing. The training and test sets are summarized in [1], Fig. 27.

We measure the performance of the f_1 prediction function for the two overlays on the Jester dataset with two different parameter settings. First, we use the same parameters that were shown to be optimal for MovieLens 100,000; in particular, the number of opinions considered for each prediction, the parameter z, is set to 60. Second, we measure performance for $z = 80000$, which means that the opinions of all peers in a peer group on an item are taken into account when making a prediction.

Mean Absolute Error

We measured the recommendation performance of the two overlays in terms of Mean Absolute Error up to the maximum peer group size of 73,420 peers (with $z = 60$). The first significant result is that the MAE performance hardly changes after 10,000 peers. At 10,000 peers the MAE is 3.29 for the best-neighbours and 3.32 for the random overlay. At the maximum of 73,420 peers for both overlays the MAE is 3.29 smiles.

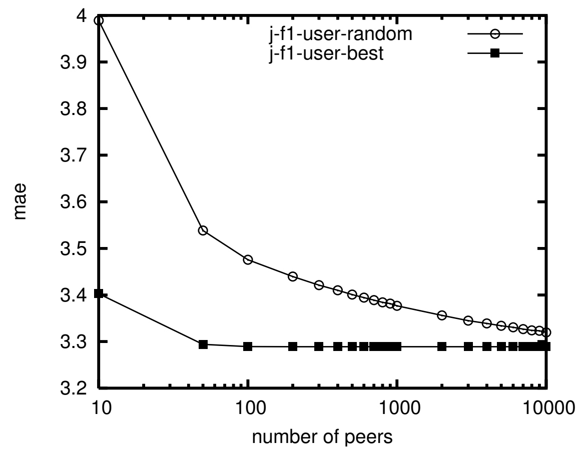
Even more importantly, if we look at the MAE results for up to 10,000 peers shown in Fig. 2(a), the MAE performance hardly changes after 200 peers. The (absolute) difference in performance between 200 and 73,420 peers is 0.00 for the best-neighbours overlay and 0.15 for the random overlay. As MAE is in terms of the rating scale, this means the average prediction is at most 0.15 smiles off if we use only 200 neighbours. Given the large (-10.0…10.0 smiles) rating scale, this difference is negligible.

Most significantly, the difference between a random peer group and a best-neighbours peer groups is small at 200 peers. Consequently, we conclude that knowing a 0.27% random sample of the user population yields basically the same results on MAE as using the most similar users in taste from the whole user population. This conclusion can also be drawn when parameter z is set to 80000 instead 60, as described in [1], Fig. 29. The only noticeable changes compared to $z = 60$ are that the overall performance of f_1 gets slightly worse for groups of more than 300 peers.

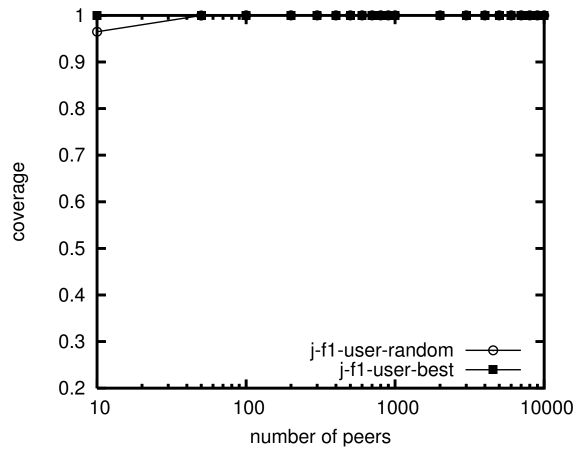
Virtual 1–5-star Precision and Recall

As the Jester rating scale is continuous the precision and recall metrics cannot be applied directly. For easy analysis, we translate the continuous rating scale into a discrete, virtual 1–5 star rating scale as follows. The interval [-10.0,-6.5] is mapped to a rating of 1 virtual star (vstar), the interval (-6.5,-2.5] is mapped to 2 virtual stars, (-2.5,2.5) is mapped to 3 virtual stars, [2,5 to 6.5) is mapped to 4 virtual stars and [6.5,10] is mapped to a rating of 5 virtual stars. Note that the (-2.5,2.5) interval is larger than the others, so this translation is biased towards 3 virtual stars.

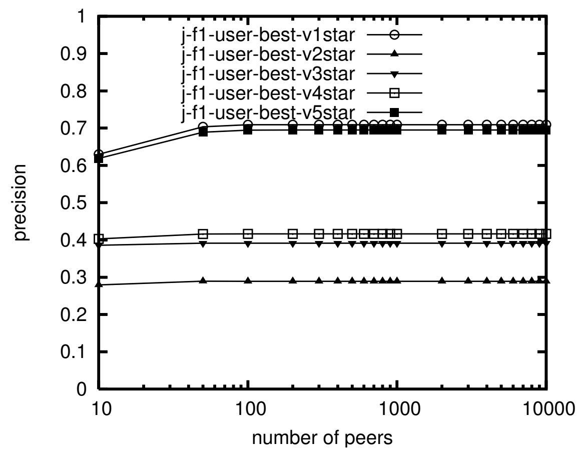
The precision and recall for the two overlays on Jester is shown in Fig. 2(c,d) and 2(e,f), respectively. First, the performance of both overlays changes particularly little as peer-group size increases. Second, the differences between using a best-neighbours overlay and a random overlay are also small. For precision, they differ marginally only below approx. group sizes of 500 peers. For recall, the best-neighbours has slightly better recall across the whole range. For both overlays precision is high at the extremes of the scale and recall is relatively low. Also for $z = 80000$ rec-



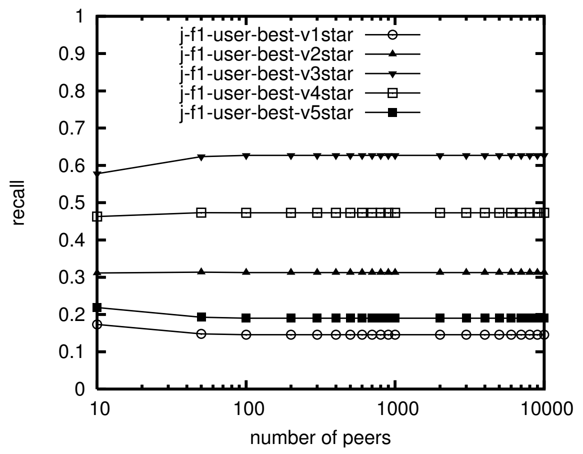
(a)



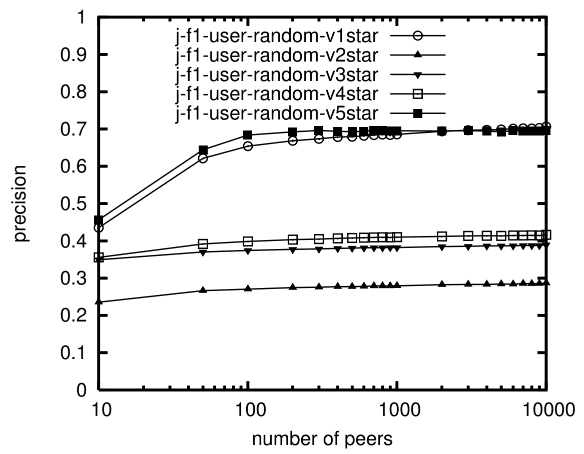
(b)



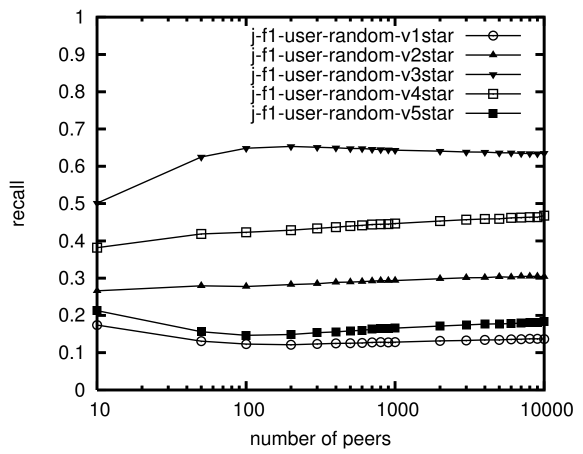
(c)



(d)



(e)



(f)

Figure 2: Measurements for Jester with $z = 60$. For the random overlays, values are averaged over three runs. The vertical errorbars show the minimum and maximum value obtained in these three runs. The x-axis is on a logarithmic scale, and ends at 10000 in all graphs. (a) Mean Average Error. (note that the y-axis starts at 3.2.) For comparison, the MAE of the trivial algorithm that always predicts the average rating value for Jester (0.74 smiles) is 4.48. (b) Coverage (note that the y-axis starts at 0.2.) (c) The virtual 1–5 star precision of f_1 for similar peers. (d) The associated virtual 1–5 star recall. (e) The virtual 1–5 star precision of f_1 for random peers. (f) The associated virtual 1–5 star recall.

ommendation quality does not change much as group sizes become over 200 peers, as described in [1].

From these results we conclude that for the Jester dataset, using a small, random group of users rather than a group of similar peers of any size for recommendation has little effect on the quality of the recommendations measured with our precision and recall metrics. With parameter z set to 60 there is no noticeable effect and when set to 80000 the effect is small.

5. RELATED WORK

PocketLens is an item-based prediction algorithm designed specifically for a peer-to-peer setting. In [8], Miller et al. evaluated the performance of PocketLens using several different underlying overlays: a Gnutella-based random overlay, a best-neighbors overlay, and two Distributed-Hash Table-based overlays. The performance of each overlay was tested using a non-standard version of the MovieLens 100K dataset with twice as many items. They found the best MAE performance was achieved by the random overlay and with sufficient coverage (already 90% for groups of just 65 peers). Their measurements thus support our conclusion that random overlays can be used for decentralized CF algorithms. In [9]) we showed it also holds for PocketLens on the standard MovieLens 100K dataset. Here we showed it holds for an user-based algorithm and two additional datasets, and when measured using more expressive metrics.

Our analysis focused on using Herlocker et al.'s algorithm on partial knowledge of ratings based datasets and two recommendation tasks. First, in this paper, we looked at the task of providing a *given* set of items with an accurate rating, as is required for annotating an Electronic Program Guide with ratings. This is called the "Annotation in Context" task in [6]. Second, in our technical report, we also looked at the specific task of recommending *some* good items, which may be sufficient and more efficiently achievable than finding the global Top-N best items for our PVR context [1]. In this second task, small random overlays also perform similarly to best-neighbours overlays.

There are many existing user-based decentralized collaborative filtering solutions that build a best-neighbours overlay [3, 16, 10, 2, 12, 7, 14]. Unfortunately, they use different similarity functions, recommendation algorithms, data sets and recommendation tasks. Therefore, more work is needed to see to which of these our findings apply.

6. CONCLUSIONS AND FUTURE WORK

Our experiments with the Jester and MovieLens datasets bring us to the conclusion that when Herlocker et al.'s algorithm is used the neighbours from which a peer receives ratings data may not be critical to the quality of peer-to-peer recommendations. That is, neither the number of neighbours nor selecting the most similar really matters. If a peer has access to ratings from a few hundred, randomly chosen other nodes, we see that reasonable recommendations can be obtained, even irrespective of the total size of the population. This is a notable result in light of the various efforts to port existing centralized collaborative-filtering algorithms to peer-to-peer networks. We conjecture that there may be no need to incur the added costs of structuring a network in order to get good quality recommendations. As future work we plan to investigate why such so little and randomly obtained partial knowledge is sufficient.

7. REFERENCES

[1] A. Bakker, E. Ogston, and M. van Steen. Random Opinions in Decentralized Recommendation. Technical report, Department of Computer Science, Vrije Universiteit Amsterdam, Sept. 2008 (updated).

[2] E. Díaz-Avilés, L. Schmidt-Thieme, and C.-N. Ziegler. Emergence of Spontaneous Order Through Neighorhood Formation in Peer-to-Peer Recommender Systems. In *Proceedings WWW '05 International Workshop on Innovations in Web Infrastructure (IWI '05)*, Chiba, Japan, May 2005.

[3] L. Foner. A Multi-Agent Referral System for Matchmaking. In *Proceedings First International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, London, UK, Apr. 1996.

[4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*, 4(2):133–141, July 2001.

[5] J. Herlocker, J. Konstan, and J. Riedl. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Information Retrieval*, 5(4):287–310, Oct. 2002.

[6] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, Jan. 2004.

[7] B. Kim, Q. Li, and A. Howe. A Decentralized CF Approach Based on Cooperative Agents. In *Proceedings of the 15th international conference on World Wide Web (WWW'06)*, pages 973–974, Edinburgh, Scotland, UK, May 2006.

[8] B. Miller, J. Konstan, and J. Riedl. PocketLens: Toward a Personal Recommender System. *ACM Transactions on Information Systems*, 22(3):437–476, July 2004.

[9] E. Ogston, A. Bakker, and M. van Steen. On the Value of Random Opinions in Decentralized Recommendation. In *Proceedings 6th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS'06)*, Bologna, Italy, June 2006.

[10] T. Olsson. *Bootstrapping and Decentralizing Recommender Systems.* Licentiate dissertation, Department of Information Technology, Uppsala University, Sweden, June 2003.

[11] Oregon State University. COllaborative Filtering Engine version 0.4. `http://eecs.oregonstate.edu/iis/CoFE/`, Sept. 2005.

[12] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A Social-Based Peer-to-Peer System. In *Proceedings 5th International Workshop on Peer-to-Peer Systems (IPTPS'06)*, Santa Barbara, CA, USA, Feb. 2006.

[13] P. Resnick and H. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.

[14] G. Ruffo and R. Schifanella. A Peer-to-Peer Recommender System Based On Spontaneous Affinities. In *ACM Transactions on Internet Technology*, volume 9, pages 4:1–4:34, Feb. 2009.

[15] U. Shardanand and P. Maes. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proceedings 1995 ACM SIGCHI Conference on Human Factors in Computing Systems*, pages 210–217, Denver, CO, USA, May 1995.

[16] A. Tveit. Peer-to-peer Based Recommendations for Mobile Commerce. In *Proceedings of the ACM Mobile 1th Workshop on Mobile Commerce*, pages 27–29, Rome, Italy, July 2001.

[17] University of Minnesota. GroupLens Home Page. `http://www.grouplens.org/`, Sept. 2005.