The Wisdom of the Few

A Collaborative Filtering Approach Based on Expert Opinions from the Web

Xavier Amatriain
Telefonica Research
xar@tid.es

Neal Lathia
University College of London
n.lathia@cs.ucl.ac.uk

Josep M. Pujol
Telefonica Research
jmps@tid.es

Haewoon Kwak
KAIST
haewoon@an.kaist.ac.kr

Nuria Oliver
Telefonica Research
nuriao@tid.es

ABSTRACT

Nearest-neighbor collaborative filtering provides a successful means of generating recommendations for web users. However, this approach suffers from several shortcomings, including data sparsity and noise, the cold-start problem, and scalability. In this work, we present a novel method for recommending items to users based on *expert* opinions. Our method is a variation of traditional collaborative filtering: rather than applying a nearest neighbor algorithm to the user-rating data, predictions are computed using a set of *expert* neighbors from an *independent* dataset, whose opinions are weighted according to their similarity to the user. This method promises to address some of the weaknesses in traditional collaborative filtering, while maintaining comparable accuracy. We validate our approach by predicting a subset of the Netflix data set. We use ratings crawled from a web portal of expert reviews, measuring results both in terms of prediction accuracy and recommendation list precision. Finally, we explore the ability of our method to generate useful recommendations, by reporting the results of a user-study where users prefer the recommendations generated by our approach.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance, Theory

Keywords

Recommender Systems, Collaborative Filtering, Experts, Cosine Similarity, Nearest Neighbors, Top-N Recommendations

1. INTRODUCTION

Collaborative filtering (CF) is the current mainstream approach used to build web-based recommender systems [1]. CF algorithms assume that in order to recommend items to users, information can be drawn from what other similar users liked in the past. The Nearest Neighbor algorithm, for instance, does so by finding, for each user, a number of *similar* users whose profiles can then be used to predict recommendations. However, defining similarity between users is not an easy task: it is limited by the sparsity and noise in the data and is computationally expensive.

In this work, we explore how professional raters in a given domain (i.e. *experts*) can predict the behavior of the general population. In recent work [2], we have found that a significant part of the error in explicit feedback-based CF algorithms is due to the noise in the users' explicit feedback. Therefore, we aim at using feedback from less noisy sources (*i.e.* experts in the context of this work) to build recommendations. We define an *expert* as an individual that we can trust to have produced thoughtful, consistent and reliable evaluations (ratings) of items in a given domain.

Our goal is not to *increase* CF accuracy, but rather to: (a) study how preferences of a large population can be predicted by using a very small set of users; (b) understand the potential of an independent and uncorrelated data set to generate recommendations; (c) analyze whether professional raters are good predictors for general users; and (d) discuss how this approach addresses some of the traditional pitfalls in CF.

The contributions of this paper include:

1. Collecting and comparing, in Section 2, the characteristics of two datasets: the Netflix dataset[1] of user-movie ratings, and the opinions collected from the web from over 150 movie critics (experts).

2. Designing an approach to predict personalized user ratings from the opinions of the experts. Section 3 outlines traditional CF and describes the proposed algorithm.

3. Evaluating the use of expert opinions as predictors of user preferences, both in terms of prediction accuracy and recommendation list precision (described in Section 4). In Section 5 we complement these results with a user study where we compare our approach with three baseline methods: random, standard Nearest-Neighbor CF and average popularity in the experts data set.

[1]http://www.netflixprize.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.
Copyright 2009 ACM 978-1-60558-483-6/09/07 ...$5.00.

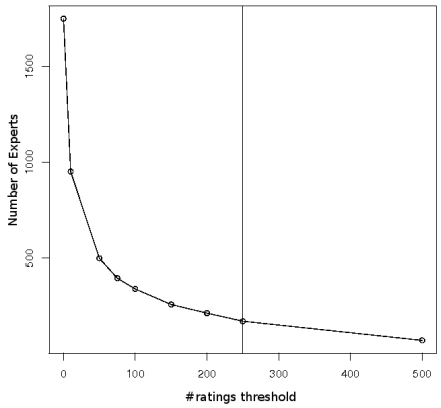


Figure 1: Relation between minimum ratings threshold and number of selected experts

2. MINING THE WEB FOR EXPERT RATINGS

The first step in our approach requires obtaining a set of ratings from a reduced population of *experts* in a given domain. One option is to obtain item evaluations from trusted sources and use a rating inference model [3] or an automatic expert detection model [4]. However, in domains where there are online expert evaluations (*e.g.* movies, books, cars, etc.) that include a quantitative rating, it is feasible to crawl the web in order to gather expert ratings. In our work, we have crawled the Rotten Tomatoes[2] web site – which aggregates the opinions of movie critics from various media sources, to obtain expert ratings of the movies in the Netflix data set. Note that there are other approaches to populate a database of expert ratings, ranging from a manually-maintained database of dedicated experts to the result of crawling and inferring quantitative ratings from online reviews. The focus of our work is not on extracting the expert ratings, but on using such an external and reduced source of ratings to predict the general population.

The ratings extracted from our experts source correspond to $8,000$ of the total of $17,770$ movies in the Netflix data set. The missing movies had significantly different titles in both databases and were difficult to match. For the purpose of this study, we believe that 50% of the Netflix data set movies is a sufficiently large sample.

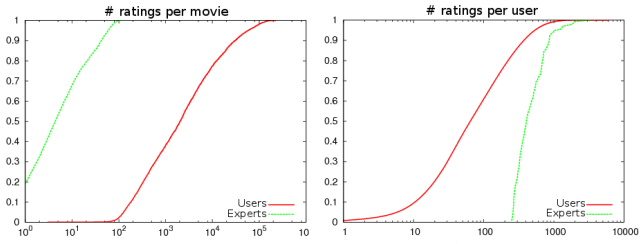
We collected the opinions of $1,750$ experts. However, an initial analysis showed that many of them had very few ratings and were therefore not adding any improvement to our predictions (see per user distribution in the experts data set in Figure 2b). Therefore, we removed those experts who did not contain at least ρ ratings of the Netflix movies. Using a final threshold of $\rho = 250$ minimum ratings, we kept 169 experts. The relation between ρ and the number of selected experts is depicted in Figure 1. This low number of experts number highlights the potential of our method to predict user preferences using a small population as the source.

2.1 Dataset analysis: Users and Experts

Before reporting the results on the performance of our expert-CF approach, we compare next the expert and Netflix datasets.

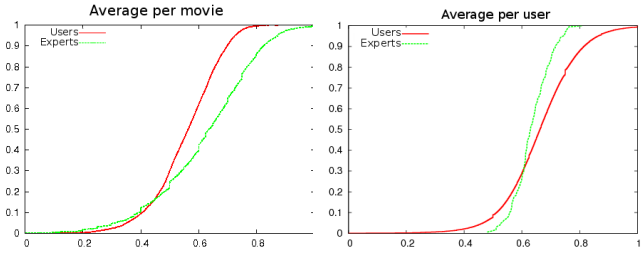
Number of Ratings and Data Sparsity. The sparsity coefficient of the user data set is roughly 0.01, meaning that only 1% of the positions in the user matrix have non-zero values. Figure 2b depicts the distribution of the number of ratings per user and per expert. An average Netflix user has rated less than 100 movies while only 10% have rated over 400 movies. Conversely, the expert set contains around $100,000$ movie ratings, yielding a sparsity coefficient of 0.07. Figure 2b also shows that an average expert has rated around 400 movies and 10% have rated $1,000$ movies or more. Similarly, Figure 2a depicts the distribution of the number of ratings per movie: the average movie has over $1,000$ Netflix user ratings, compared to an average of 100 expert ratings. Note that 20% of the movies in our expert data set only have one rating [3]. However, the expert matrix is less sparse than the user matrix and more evenly distributed, both per user and per movie.

[2]http://www.rottentomatoes.com



(a) (b)

Figure 2: Comparison of the CDF of ratings per (a) movie and (b) user in Netflix and Rotten Tomatoes (experts) Datasets



(a) (b)

Figure 3: Average rating per movie (a) and user (b) in Netflix and Rotten Tomatoes

Average Rating Distribution. Figures 3a and 3b depict the distribution of the mean score of the ratings per movie (a) and user (b) in the Netflix (red line) and Rotten Tomatoes or expert (green line) datasets. As seen in Figure 3a, the average rating in Netflix is around 0.55 (or 3.2⋆); 10% of the movies have a rating of 0.45 (2.8⋆) or less, while 10% of the movies have an average rating of 0.7 (4⋆) or higher. Conversely, experts rate movies with an average score slightly larger than 0.6 (3.5⋆); 10% of the movies have a mean rating $\leq$ 0.4 (2⋆), but the range $\geq$ 0.8 to 1 also accounts for 10% of the movies.

In a similar way, Figure 3b shows that the user ratings have a normal distribution centered around 0.7 (4⋆) while expert ratings are centered around 0.6 (3.5⋆). Note that in this case, only 10% of the users have a mean score $\leq$ 0.55 and another 10% $\geq$ 0.7. In terms of how the average rating is distributed, we see that experts show greater variability per movie than per user: while experts tend to behave similarly in terms of their average rating, their overall opinion

[3]This was not a result of our filtering of experts with few ratings, but a limitation of our original data set

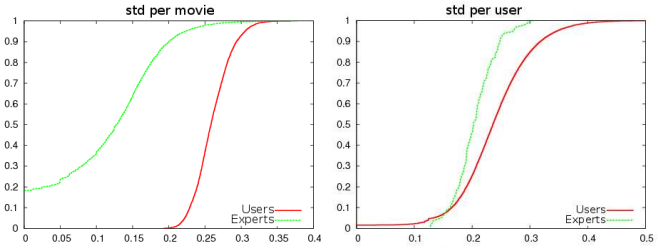


Figure 4: Per movie (a) and user (b) standard deviation (std) in Netflix and Rotten Tomatoes

on the movies is more varied. This could be due to underlying incentives to rate movies, since experts are likely to watch and rate movies regardless of whether they like them or not, while users tend to be biased towards positively rating movies [5]. We also detect a much larger proportion of movies in the highest rating range for experts. Experts seem to consistently agree on what the "excellent" movies are.

Rating Standard Deviation (std). Figures 4a and 4b plot the distribution of the std per movie (a) and user (a) in Netflix and Rotten Tomatoes, respectively. In the case of users, the std per movie (Figure 4a) is centered around 0.25 (1⋆) with very little variation, while the expert data set has significantly lower std (0.15) and larger variation. Note that in the expert data, 20% of the items show no std as there is only one rating. The std per user (Figure 4b) is centered around 0.25 for the Netflix data with larger variability than in the per movie case. When looking at the expert data, the average std per user is 0.2 with small variability.

The above analysis highlights the large differences between our user and expert sets. The expert data set is much less sparse than the users'. Experts rate movies all over the rating scale instead of being biased towards rating only popular movies. However, they seem to consistently agree on the good movies. Experts also have a lower overall standard deviation per movie: they tend to agree more than regular users. Also, the per-expert standard deviation is lower than that seen between users, meaning that they tend to deviate less from their personal average rating.

3. EXPERT NEAREST-NEIGHBORS

The generic CF method applies the kNN algorithm to predict user ratings. The algorithm computes a prediction for a user-item pair, based on a number k of *nearest neighbors*, which can either be user- or item-based [6]. Although it is possible to use either approach, we choose user-based CF for its transparent applicability to experts. Both approaches can be decomposed into a sequence of stages. In the first stage, a matrix of user-item ratings is populated. Then, the similarity between all pairs of users is computed, based on a pre-determined measure of similarity. In a similar way to what Herlocker *et. al* propose [7], we use a variation of the cosine similarity which includes an adjusting factor to take into account the number of items co-rated by both users. Given users a and b, item i, user-item ratings r_{ai} and r_{bi}, the number of items N_a and N_b rated by each user, and the number of co-rated items $N_{a\cup b}$ the similarity is computed as:

$$sim(a,b) = \frac{\sum^i (r_{ai} r_{bi})}{\sqrt{\sum^i r_{ai}^2}\sqrt{\sum^i r_{bi}^2}} \cdot \frac{2N_{a\cup b}}{N_a + N_b} \qquad (1)$$

We propose an approach to CF that *only uses expert opinions* to predict user ratings. Therefore, our approach does not require the user-user similarity to be computed; instead, we build a similarity matrix between each user and the expert set. We take a slightly different approach than regular k-NN CF. Our *expert*-CF method is closely related to Ma *et al.*'s method for neighborhood selection [8]. In order to predict a user's rating for a particular item, we look for the experts whose similarity to the given user is greater than δ. Formally: given a space V of users and experts and a similarity measure $sim\colon V \times V \rightarrow \mathbb{R}$, we define a set of experts $E = \{e_1, ..., e_k\} \subseteq V$ and a set of users $U = \{u_1, ..., u_N\} \subseteq V$. Given a particular user $u \subseteq U$ and a value δ, we find the set of experts $E' \subseteq E$ such that: $\forall e \subseteq E' \Rightarrow sim(u,e) \geq \delta$.

One of the drawbacks of using a fixed-threshold δ is the risk of finding very few neighbors; furthermore, the ones that are found may not have rated the current item. In order to deal with this problem, we define a *confidence threshold* τ as the *minimum* number of expert neighbors who must have rated the item in order to trust their prediction. Given the set of experts E' found in the previous step and an item i, we find the subset $E'' \subseteq E'$ such that $\forall e \subseteq E'' \Rightarrow r_{ei} \neq \circ$, where r_{ei} is the rating of item i by expert $e \subseteq E'$, and $\circ$ is the value of the *unrated item*.

Once this subset of experts $E'' = e_1 ... e_n$ has been identified, if $n < \tau$, no prediction can be made and the user mean is returned. On the other hand, if $n \geq \tau$, a predicted rating can be computed. This is done by means of a similarity-weighted average of the ratings input from each expert e in E'' [9]:

$$r_{ai} = \sigma_u + \frac{\sum_{e\subseteq E''} (r_{ei} - \sigma_e) sim(e,a)}{\sum sim(e,a)} \qquad (2)$$

where r_{ai} is the predicted rating of item i for user a, r_{ei} is the known rating for expert e to item i, and σ_u and σ_e are the respective mean ratings.

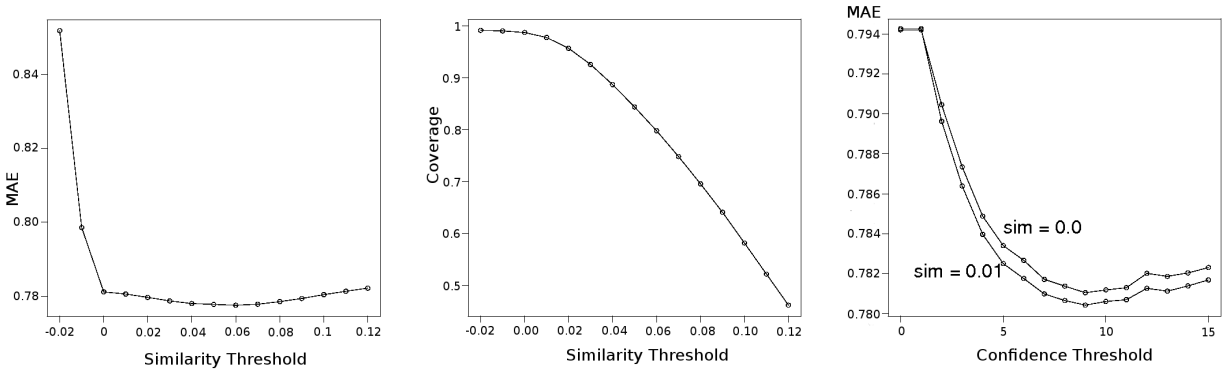
In the next section, we report on the interplay of the two parameters δ and τ. The optimal setting of these parameters depends on the data set and the application in mind, as it is the case with other state-of-the-art CF algorithms. Also, and in order to make our results comparable, note that we use the same threshold-based nearest-neighbor approach when comparing with the standard CF method.

4. RESULTS

Based on the previously described data, we measure how well the 169 experts predict the ratings of the 10,000 Netflix users. In order to validate our approach, we set up two different experiments: in the first experiment, we measure the mean error and coverage of the predicted recommendations. In the second experiment, we measure the precision of the recommendation lists generated for the users.

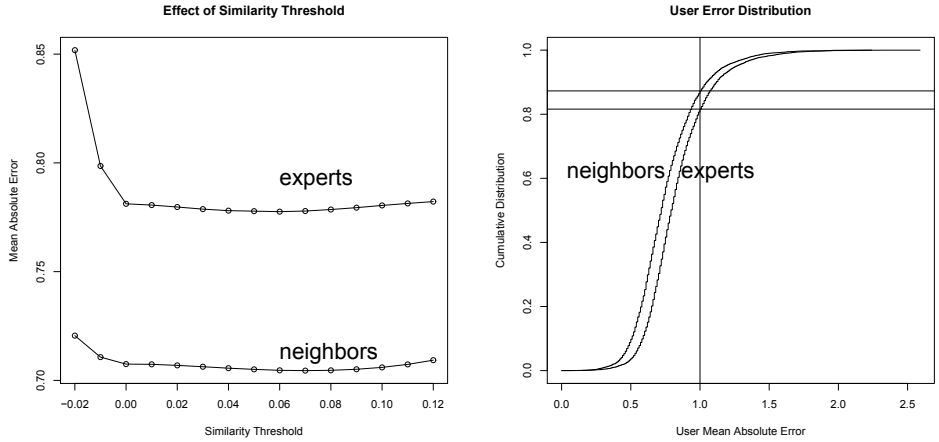
4.1 Error in Predicted Recommendations

In order to evaluate the predictive potential of expert ratings, we divided our user data set (by random sampling) into 80% training - 20% testing sets and report the average results of a 5-fold cross-validation. We use the average for all experts over each given item as a worst-case baseline measure. This is equivalent to a non-personalized "critics' choice" recommendation, which produces a Mean Average



(a) (b) (c)

Figure 5: (a) Mean absolute error (MAE) and (b) coverage of expert-predicted ratings as a function of the minimum similarity (δ) and confidence (τ) thresholds; (c) MAE versus confidence threshold.



(a) (b)

Figure 6: Comparison between Expert CF and Nearest-Neighbor CF: (a) MAE and (b) per user error.

Method	MAE	Coverage
Critics' Choice	0.885	100%
Expert-CF	**0.781**	**97.7%**
Neighbor-CF	0.704	92.9%

Table 1: Summary of the MAE and Coverage in our Expert-based CF approach compared to Critics' Choice and Neighbor CF

Error (MAE) of 0.885 and full coverage. Setting our parameters to $\tau = 10$ and $\delta = 0.01$, we obtain a MAE of 0.781 and a coverage of 97.7%. Therefore, expert-CF yields a significant accuracy improvement with respect to using the experts' average. As far as coverage is concerned, the setting of the parameters represents a small loss. We shall turn next to the details of how the two parameters in our approach (δ or minimum similarity and τ or confidence threshold) interplay.

Figure 5a shows that the MAE in the expert-CF approach is inversely proportional to the similarity threshold (δ) until the 0.06 mark, when it starts to increase as we move to higher δ values. The accuracy below the 0.0 threshold degrades rapidly[4], as we are taking into account too many experts; above 0.06, though, we have too few experts in the neighborhood to make a good prediction. If we look at Figure 5b we see how it decreases as we increase δ. For the optimal MAE point of 0.06, coverage is still above 70%. The ultimate trade-off between MAE and coverage will be application-specific. Turning to Figure 5c, we see how the MAE evolves as a function of the confidence threshold (τ). The two depicted curves correspond to $\delta = 0.0$ and $\delta = 0.01$. We choose these two values of our similarity threshold as they produce a reasonable tradeoff between accuracy and coverage.

Standard neighbor-CF (ignoring the expert data set) offers a second baseline measure of performance. A side-to-side comparison gives us a first intuition.

Using the Netflix users as neighbors, we measure a MAE of 0.704 and 92.9% coverage when the $\delta = 0.01$ and $\tau = 10$ (see summary in Table 1) [5]. Figure 6 includes a detailed

[4]Note that this threshold values are dependent on the chosen similarity measure. In our case, we are using a symmetric cosine similarity that can yield values between $[-1, +1]$.

[5]It should be noted, for consistency sake, that using this parameters yields very similar results to standard kNN CF

comparison of the accuracy and coverage of the expert-CF (experts line) and NN-CF (neighbors) methods as a function of the similarity threshold. While NN-CF has a MAE 10% lower than expert-CF, the difference in their coverage is also around 10%, favoring the experts in this case.

Finally, we are interested in measuring whether the difference in prediction accuracy is equally *distributed* among all target users. Figure 6b includes the cumulative distribution of per-user error for both methods. Note how both curves separate at around MAE = 0.5 and they run almost parallel with a separation of around 0.1 until they start joining again at around the point of MAE = 1. This means that neighbor NN works better for the minority of users with a low MAE average of less than 0.5, which represent around 10% of our population. Both methods perform equally the same for the rest of the users – with the expert-CF approach performing even slightly better for users with a MAE higher than one. This is a positive feature of our approach that only performs worse on users that are highly predictable, in which a slight increase in error should be acceptable.

4.2 Top-N Recommendation Precision

Although measuring the mean error on all predictions for a test data set is currently an accepted measure of success for recommender systems, we are interested in evaluating our results in a more realistic setting. A "real-world" recommender system only recommends the set of items that the user may like. This is similar to the idea of top-N recommendation lists [11].

The evaluation of recommendation algorithms through top-N measures has been addressed before [12, 13]. All of these approaches rely on the use of the well-known precision and recall measures. Ziegler *et al.* show [14] that evaluating recommender algorithms through top-N lists measures does not map directly to the user's utility function. However, it does address some of the limitations of the more commonly accepted accuracy measures, such as MAE.

We propose a variation as an extension to the previously described approaches. In our case, we do not construct the list of *recommendable* items by fixing N, but rather classify items as being *recommendable* or *not recommendable* given a threshold: if there is no item in the test set that is worth recommending to a given target user, we simply return an empty list. We believe that using a *recommendable threshold* is a more appropriate measure for top-N recommendations than, for instance, the ones proposed by Deshpande and Karipis [11], where the user rating was not taken into account in the evaluation process. The most similar approach to ours is that proposed by Basu *et al.* [15], where they use the *top quartile* of a user's ratings in order to decide whether a movie is *recommendable*. We do not use the *modified precision* measure proposed in [13] for two reasons: First, because we believe it is unfair with algorithms that promote serendipity. And second, and most important, because we will later do a final validation with a user study in which we will use the same procedure for generating the lists. In that setting, we will aim at recommending previously unrated items. Therefore, penalizing unrated items as proposed in the *modified precision* would make both results not comparable.

with $k = 50$, which is a common setting [10]. In this case we measure a MAE of 0.705.

We define an item to be *recommendable* if its predicted value is greater than σ. With this definition in mind, we evaluate our system as a 2-class classification problem:

1. For a given user, compute all predictions and present those greater or equal than σ to the user
2. For all predicted items that are present in the user's test set, look at whether it is a true positive (actual user rating greater or equal to σ) or a false positive (actual user rating less than σ).
3. Compute the *precision* of our classifications using the classical definition of this measure

We measure a precision of 0.76 using the same parameter values reported in Section 4.1 and setting $\sigma = 4$. This means that 76% of the items recommended by the experts found in the user test set are qualified as recommendable by the users. Note, however, that a *recommendable threshold* of 4 is quite restrictive. If we lower σ to 3, we measure a precision of 89%. Figure 4.2 depicts the variation of the precision of NN-CF (neighbors line) and expert-CF (experts line) with respect to σ. For $\sigma = 4$, the baseline method clearly outperforms our expert-based approach with a precision of 0.85. However, for $\sigma = 3$ the precision in both methods is similar. Therefore, for users willing to accept recommendations for any *above average* item, the expert-based method appears to behave as well as a standard NN-CF.

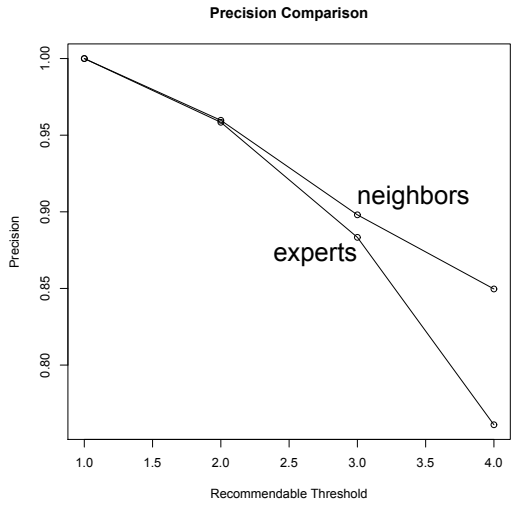
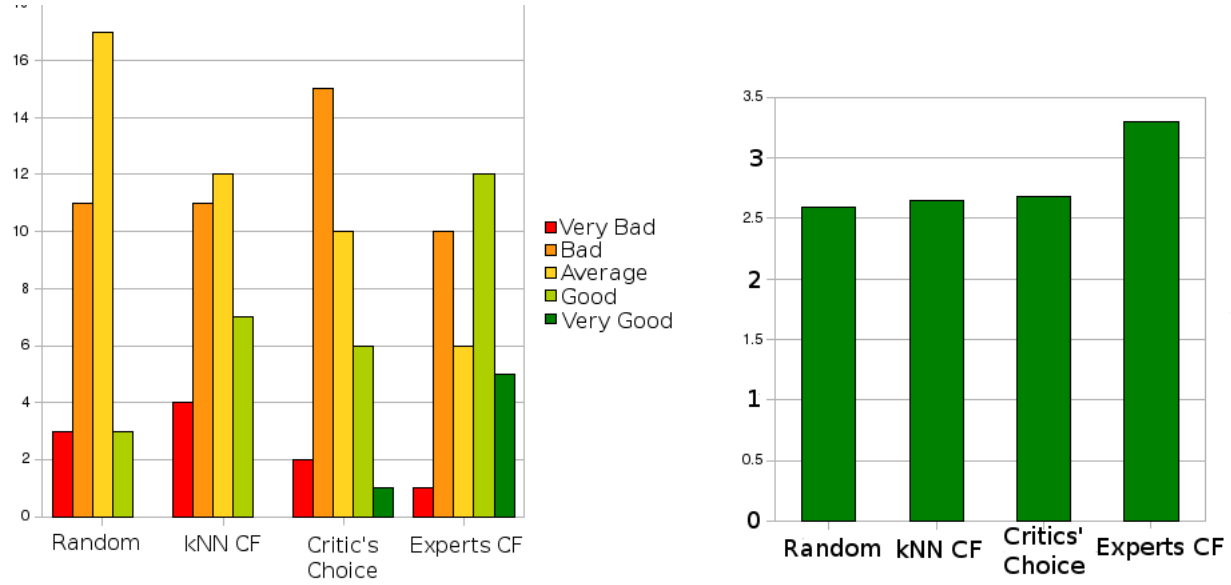


Figure 7: Precision of Expert CF (experts line) as compared to the baseline NN CF (neighbors line) as a function of the recommendable threshold σ.

5. USER STUDY

Although error and recommendation list precision are prominent metrics used for CF evaluation, their relation to the actual *quality* of the recommendation is unclear. Therefore, we designed a user study to further validate our findings. We designed a web interface that asked users to rate 100 preselected movies. The selection was done by using a stratified random sample on the movie popularity curve: we divided the 500,000 Netflix movies into 10 equal-density bins and random sampled 10 movies from each bin. We provided a "have not seen" button so users could voluntarily decide which movies to rate.



(a) Comparison of the responses to overall quality of the recommendation lists generated by 4 methods.

(b) Overall quality of the expert-CF method as compared to others, average response

Figure 8: Overall quality of different recommendation strategies as assessed by our user study participants.

57 participants were recruited via email advertisement in a large telecommunications company. The participants' age ranged from 22 to 47 years, with an average age of 31.2 years. Almost 90% of our participants were in the 22 to 37 age group and most of them were male (79.12%). Note, however, that this demographic group corresponds to the most active group in online applications such as recommender systems [16].

Using the collected ratings and the 8,000 movies that we considered in the above experiments, we generated 4 top-10 recommendation lists: **(i) Random List**: A random sequence of movie titles; **(ii) Critics choice**: The movies with the highest mean rating given by the experts. If two movies had the same mean, we ranked them based on how many experts had rated the movie; **(iii) Neighbor-CF**: Each survey respondents' profile was compared to the ratings of the users in the Netflix data subset: the top-10 list was derived by ordering the unrated movies with predictions higher than the recommendable threshold; and **(iv) Expert-CF**: Similar to (iii), but using the expert dataset instead of the Netflix ratings.

Both neighbor-CF and expert-CF used the same settings: a similarity threshold of 0.01 and a confidence measure of 10. Note that, as we saw in Figure 6, the setting of a given similarity threshold is not unfair to any of the two approaches. Therefore, we choose a value that gives us enough coverage so as to produce enough recommendations. For the settings we chose, the RMSE values are the ones we included in Table 1. Increasing the similarity threshold would reduce coverage, while the ratio between both RMSE's would remain the same. Increasing the confidence threshold would also reduce coverage. Note that the neighbor-CF algorithm would be especially sensitive to changes in the confidence threshold. The coverage of neighbor-CF is 92.9% with the current setting of the threshold, compared to 97.7% in the case of expert-CF. Also note that, as already explained in the previous section, these settings for the neighbor-CF yield an error comparable to standard kNN with $k = 50$.

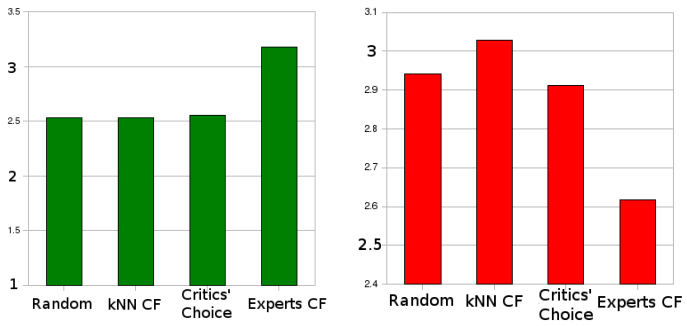
We then asked our participants to evaluate: (i) The *overall quality* of the recommendation lists; (ii) whether they included items that the user *liked*; (iii) whether they included items that the user *did not like*; and (iii) whether they included *surprising* items.

The average number of ratings was of 14.5 per participant. We had to generate the recommendation lists based on such limited user feedback. Therefore, our test conditions are similar to a real setting where the user has recently started using the recommendation service. Our results are thus assessing how well the methods perform in cold-start conditions.

Figure 8 includes the responses to the overall list quality. Participants were asked to respond in a 1 to 5 Likert scale as the one included in Figure 8a. Note that we also report the averaged results in Figure 8b. We can see that the expert-CF approach is the only method that obtains an average rating higher than 3. Interestingly, the standard neighbor-CF approach performs almost as badly as a randomly generated list. The only approaches that are qualified as very good are expert based: critics' choice and expert-CF. In addition, 50% of the users considered the expert-CF recommendations to be good or very good. Finally, it is important to stress that although an average rate of 3.4 might seem low, this is in a cold-start situation with very limited information from each user.

In Figure 9 we report on the replies to the question of whether participants felt the generated lists contained items they knew they would like/dislike. In these cases, we used a 1 to 4 Likert scale, removing the neutral response. We report average results for clarity. In Figure 9a, we again see that the expert-CF approach outperforms the rest of the methods. More than 80% of the users agreed (or fully agreed) that this method generates lists that include movies they like. It is interesting to note that this figure is similar to the assessed overall list quality in Figure 8b.

Figure 9b summarizes the ratings to the question "the list contains movies I think I would not like". This question



(a) (b)

Figure 9: User study responses to whether the list contains movies users would like (a) or would not like (b) .

is very important: recommending wrong items mines the user's assessment of the system and compromises its usability [17]. Therefore, an important aspect on the evaluation of a recommender system is how often the user is disappointed by the results. The expert-CF approach generates the least negative response when compared to the other methods.

Finally, we performed an analysis of variance (anova) to test whether the differences between the four recommendation lists are statistically significant or not. The null hypothesis is that the average user evaluation for the four different lists is the same. The confidence level is set to 99%, such that p-values smaller than 0.01 imply a rejection of the null hypothesis. The p-value for the fours lists is $5.1e-05$ and consequently the null hypothesis is rejected. If we leave out the expert-CF algorithm from the analysis, we measure a p-value of 0.42. In this case, we conclude that the differences on the user satisfaction from the three baseline methods is not statistically significant. The cross-comparison between the expert-CF and the other three algorithms gives p-values of $2.14e-05$, $2.4e-03$ and $8e-03$ for Random, neighbor-CF and Critics' Choice respectively. From these results, we conclude that the differences on the participants' satisfaction produced by the expert-CF algorithm cannot be attributed to the sampling of the user study.

6. DISCUSSION

In this paper, we have introduced a recommender system that uses a small set of expert ratings to generate predictions for a large population of users. Our experiments are based on using online reviews from movie critics. However, we could also think of a small set of "professional" raters maintaining a rating database. This is reminiscent but less demanding than content-based recommender systems that rely on experts to manually categorize and label content [18].

The use of experts to generate predictions in a recommender system has been explored by Cho *et al.* [19]. Their approach, however, is focused on identifying expert users from *within* a closed community of users, by deriving a "domain authority" reputation-like score for each user in the data set.

The idea of *expertise* is also related to that of *trust*. In trust-aware recommender systems the influence of neighbors is weighed by a measure of how trustworthy they are for the current user. The trust measure can be defined and obtained in different ways. For instance, O'Donovan and Smyth [20] compute a measure of trust by looking at how well a neighbor has predicted past ratings.

As shown in Section 4, our approach does not outperform a "naive" neighbor-CF approach[6]. However, our focus when using external data sources is not as much on prediction accuracy as it is on addressing some of the common problems found in traditional CF recommender systems. In the next paragraphs, we shall describe a few of these problems and discuss how they might be addressed by using a limited set of external experts to generate the predictions (*i.e.* wisdom of the few).

Data Sparsity: In a standard collaborative recommender system, the user-rating data is very sparse. Although dimensionality reduction techniques offer some help, this problem is still a source of inconsistency and noise in the predictions. Using the *wisdom of the few* addresses this issue since domain experts are more likely to have rated a large percentage of the items, as shown in Section 2.

Noise and Malicious Ratings. Users introduce noise when giving their feedback to a recommender system, both in the form of careless ratings [21] and malicious entries [22, 23], which will affect the quality of predictions. Experts are expected to be more consistent and conscious with their ratings, thus reducing noise. In addition, an expert data set can be immune to malicious, profile-injection attacks as it is an easy to control and stable data set.

Cold Start Problem: In a CF system, new items lack rating data and can not be recommended; the same is true when a new user enters the system [24]. Motivated expert users typically rate a new item entering the collection as soon as they know of its existence and therefore minimize item cold-start. In addition, experts should create a less sparse and noisy dataset which should improve the user cold-start problem, as shown in our user study.

Scalability: Computing the similarity matrix for N users in an M-item collection is an $O(N^2M)$ problem. This matrix needs to be updated on a regular basis, as new items and/or users enter the system. Therefore, CF based approaches typically suffer from scalability limitations. While there are several ways to address this issue – such as k-means clustering [25], scalability is still an open research problem in CF systems. The *wisdom of the few* approach is less sensitive to scale, as it creates recommendations from a very reduced set of experts (*e.g.* 169 experts *vs.* 500,000 potential neighbors in the Netflix database).

Privacy. Privacy in CF recommender systems is a growing concern and still an area of research [26]. In order to maintain and update the similarity matrix, the system has to transmit all user ratings to a central node where the matrix is computed. This step is not needed in our approach, since the similarity matrix only includes expert data and the target user. In expert-CF, the current experts ratings can be easily transmitted thanks to the reduced size of the matrix, such that all computation is performed locally on the client. This advantage is particularly relevant in a mobile scenario.

7. CONCLUSIONS

In this paper, we have proposed an approach to recommend content, based on the opinion of an external source: a

[6]It is important to note that there are several improvements to neighbor-CF that would yield better MAE. Similarly, our approach is open to algorithmic improvements that are left for future work.

reduced number of *experts*. The proposed method (*i.e.* the wisdom of the few) is able to predict the ratings of a large population by considering a reduced set of expert ratings. The method's performance is comparable to traditional CF algorithms, even when using an extremely small expert set. We have implemented this approach to predict ratings in the Netflix data set by looking at the ratings of 169 cinema critics, obtaining comparable mean errors. We have also validated the approach in a top-N recommendation setting. Finally, we have shown through a user survey that our method is preferred to standard neighbor-CF in a scenario similar to a cold-start situation. We believe that the proposed approach addresses some of the shortcomings of traditional CF: data sparsity, scalability, noise in user feedback, privacy and the cold-start problem.

Although the approach presented in this paper has shown promising results, it has also opened up opportunities for future improvement. First, we have shown the usefulness of expert-CF to a particular domain, but the approach could be implemented in other domains where expert ratings are available. The use of external experts could also be combined with regular CF in order to improve overall accuracy. We have also restricted our work to a neighbor user-based CF, but the same approach could be implemented using an item-based algorithm or other CF algorithms, such as model-based approaches.

Acknowledgements.

The work reported on this paper has been partially funded by an ICREA grant from the Generalitat de Catalunya.

8. REFERENCES

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[2] X. Amatriain, J.M. Pujol, and N. Oliver. I like it... i like it not: Evaluating user ratings noise in recommender systems. In *Proc. of the 2009 Conference on User Modeling, Adaptation, and Personalization*, 2009.

[3] I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proc. of WWW '08*, 2008.

[4] K. Balog, T. Bogers, L. Azzopardi, M. de Rijke, and A. van den Bosch. Broad expertise retrieval in sparse data environments. In *Proc. of SIGIR '07*, 2007.

[5] M. Harper, X. Li, Y. Chen, and J. Konstan. An economic model of user rating in an online recommender system. In *Proc. of UM 05*, 2005.

[6] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW '01*, 2001.

[7] J. L. Herlocker, J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, (5):287–310, 2002.

[8] H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *Proc. of SIGIR '07*, 2007.

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work*, pages 175–186, 1994.

[10] R. Bell and Y. Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc. of IEEE ICDM*, 2007.

[11] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[12] G. Karypis. Evaluation of item-based top-n recommendation algorithms. In *CIKM '01: Proc. of 10th Int. Conf. on Information and knowledge management*, pages 247–254, 2001.

[13] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proc. of SIGIR '04*, 2004.

[14] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *Proc. of WWW '05*, 2005.

[15] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. In *AAAI Workshop on Recommender Systems*, 1998.

[16] Choicestream. Personalization Survey. Technical report, Choicestream Inc., 2007.

[17] Kirsten Swearingen and Rashmi Sinha. Beyond algorithms: An hci perspective on recommender systems. In *ACM SIGIR 2001 Workshop on Recommender Systems*, 2001.

[18] Science. Rockin' to the Music Genome. *Science*, 311(5765):1223d–, 2006.

[19] J. Cho, K. Kwon, and Y. Park. Collaborative filtering using dual information sources. *IEEE Intelligent Systems*, 22(3):30–38, 2007.

[20] J. O'Donovan and B. Smyth. Trust in recommender systems. In *IUI '05: Proc. of the 10th international conference on Intelligent user interfaces*, pages 167–174, 2005.

[21] M. Mahony, N. Hurley, and G. Silvestre. Detecting noise in recommender system databases. In *Proceedings of the 2006 IUI*, 2006.

[22] J. Ford S. Zhang, Y. Ouyang and F. Makedon. Analysis of a low-dimensional linear model under recommendation attacks. In *Proc. of SIGIR '06*, 2006.

[23] B. Mehta and W. Nejdl. Attack resistant collaborative filtering. In *Proc. of SIGIR '08*, 2008.

[24] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong. Addressing cold-start problem in recommendation systems. In *ICUIMC '08: Proc. of 2nd Int. Conf. on Ubiquitous Inf. Manag. and Comm.*, pages 208–211, New York, NY, USA, 2008.

[25] G. Xue, C. Lin, Q. Yang, W. Xi, H. Zeng, Y. Yu, and Z. Chen. Scalable collaborative filtering using cluster-based smoothing. In *Proc. of SIGIR '05*, 2005.

[26] R. Baraglia, C. Lucchese, S. Orlando, M.o Serrano, and F. Silvestri. A privacy preserving web recommender system. In *ACM SAC*, pages 559–563, New York, NY, USA, 2006.