Hoeffding-CF: Neighbourhood-Based Recommendations on Reliably Similar Users

Pawel Matuszyk and Myra Spiliopoulou

Otto-von-Guericke-University Magdeburg,
Universitätsplatz 2,
D-39106 Magdeburg, Germany
{pawel.matuszyk,myra}@iti.cs.uni-magdeburg.de

Abstract. Neighbourhood-based collaborative filtering recommenders exploit the common ratings among users to identify a user's most similar neighbours. It is known that decisions made on a naive computation of user similarity are unreliable, because the number of co-ratings varies strongly among users. In this paper, we formalize the notion of *reliable similarity* between two users and propose a method that constructs a user's neighbourhood by selecting only those users that are reliably similar to her. Our method combines a statistical test and the notion of a *baseline user*. We report our results on typical benchmark datasets.

Keywords: Reliable User Similarity, Reliable Recommendations, Reliability, Hoeffding Bound, Collaborative Filtering, Recommenders

1 Introduction

Neighbourhood-based collaborative filtering (CF) engines return recommendations on the basis of user similarity. As shown in [4], similarity values computed from too few co-ratings cannot be trusted. In this study, we assert that even similarities between users with *many* ratings in common cannot always be trusted, and we introduce the concept of *reliable similarity* between users. We propose a mechanism that builds a user's neighbourhood by selecting only users, whose similarity is *reliably* useful for making recommendations to that user, no matter whether the common ratings are few or many.

User similarity on the basis of few co-ratings is unreliable [4]. Researchers have already proposed solutions to this problem, namely thresholds on the number of ratings two users should share to be considered similar, or assigning lower weights to users that have too few ratings in common [4, 2, 7]. Their inherent assumption is that similarity based on many co-ratings is informative. To see why this assumption does not always hold, assume a database with seven items $j_1, \ldots, j_7$ and assume that the average rating for j_1, j_2, j_6 is 4, the average rating for j_4 is 5, for j_4 is 2 and the average for j_3, j_7 is 3. Table 1 shows the ratings of four users for these items. Note that u_2, u_4 have given a rating of 1 to j_7 (lower than the average for the item), while u_3 gave the highest possible rating.

This is an author's copy. The final publication is available at Springer via:
http://dx.doi.org/10.1007/978-3-319-08786-3_13

Given the similarity values between u_1 and each other user (last column of Table 1), should j_7 be recommended to u_1? The similarity of u_1 to u_2 and to u_4 is 1 but is based on too few ratings. If we set the threshold to 4 co-ratings, then u_2 and u_4 will be ignored or assigned very low weights, whereupon j_7 will be recommended, since the similarity of u_1 to u_3 is more than 0.99. However, u_3 assigns to each item the average rating for this item, while u_1 (similarly to u_2, u_4) rated j_2 higher than the average. What if people who love j_2 find j_7 intolerable, as *both* u_2 and u_4 do? Heuristics that assign higher weight to users with many ratings exacerbate this problem. Instead of a heuristic, we propose a significance-based solution, in which we decide whether a user (no matter how many co-ratings she has) is informative for a recommendation.

$Users\backslash Items$	j_1	j_2	j_3	j_4	j_5	j_6	j_7	cosine similarity to u_1
u_1	4	5	3	5	2	4	?	–
u_2	4	5	?	?	?	?	1	1
u_3	4	4	3	5	2	4	5	0.9957
u_4	?	5	3	?	?	?	1	1

Table 1: Ratings of four users (best rating: 5, worst: 1), and their cosine similarity to user u_1, for whom recommendations must be computed

In our approach, we first formalize the concept of *baseline user* – informally, the average user for the population under observation. Then, we introduce the concept of *reliable similarity*: we use the Hoeffding Bound (HB), derived from Hoeffding's Inequality [5], to test whether a given user is more similar to the active user than the baseline user is; we then consider as neighbours to the active user only those users whose similarity to her satisfies the bound. Hence, the recommender decides on statistical grounds whether it can make a recommendation on neighbourhood-based similarity, no matter how small this neighbourhood is.

The paper is organised as follows. The next section contains related work. In section 3 we present our method. In section 4, we evaluate our method on real datasets, focussing on the interplay among reliability, neighbourhood size and number of users with an empty neighbourhood. The last section summarizes the findings and discusses open issues.

2 Related Work

Neighbourhood-based collaborative filtering has been studied thoroughly in numerous publications. An overview and several studies on the most important aspects can be found in [9] (Chapters 1, 4, 5), where much emphasis is put on the predictive quality of a recommender's output. We, however, do not focus on further improvement of this criterion, but rather investigate how to improve the "reliability of recommendations". This should not be confused with the reliability of conclusions made about recommender systems, in the process of evaluation using hypothesis testing, as described in [10]. In contrast to statistical testing in the evaluation that aims to measure the significance of the error measures, we investigate the significance of the neighbourhood of a user.

Herlocker et al. introduce the term of significance weighting [4]. They recognize that similarity on only few co-ratings is not representative and the amount of trust in this value should be limited. To limit the influence of those unreliable similarity values they weight them with a term $n/50$, where n is the number of co-ratings. Ma et al. [7] change the weighting schema defined by Herlocker et al. Bell et al. also define a different weighting schema, which they call "shrinkage" [2]: they shrink the similarities towards a null-value to an extent that is inversely proportional to the number of co-ratings. The fewer co-ratings between users exist, the less influence does the particular similarity value have on the predicted rating value. However, none of these methods answers the question "how many co-ratings are enough?", nor addresses the more important underlying question "whose co-ratings are *useful* enough?". We formalize the latter question. Instead of using weights, we can decide whether a similarity between two users can be relied upon, independently of the exact number. We stress that our method is not a solution to the cold start problem, where no enough information about users is known. Our goal is to quantify the reliability of the known information.

3 Building Reliable Neighbourhoods of Users

Our approach consists of a formal model on *reliable similarity* of a user, an adjusted CF-based recommendation engine and a mechanism that builds a user's neighbourhood by only considering users that are truly similar to the peer user and ignoring all other users. We concentrate on user-user collaborative filtering, but our approach can be used for item-item CF as well.

3.1 Baseline Users

To compute the *neighborhood* of the active user u_A, for whom recommendations must be formulated, we first introduce the notion of a "baseline user" u_B – a default, fictive user. Informally, a user x is *reliably* similar to u_A, if u_A is more similar to x than to u_B; then, the neighbourhood of u_A consists of the users who are *reliably similar* to her. Formally, u_B is a vector:

$$u_B = [ir_1, ir_2, ..., ir_{n-1}, ir_n] \tag{1}$$

where ir_j is a rating of the item j and n is the total number of items. We consider three types of baseline users: the *average user*, the *random Gaussian user* and the *random uniform user*. For the computation of the baseline users, we use an initial sample of ratings R_{train} for training.

Average user: This baseline is computed by defining ir_j for an item j as the average rating on j in R_{train}:

$$ir_j = \frac{1}{|U(j)|} \sum_{x \in U(j)} r_{x,j} \tag{2}$$

where $r_{x,j} \in R_{train}$ is the rating of user x for item j and $U(j) = \{x | r_{x,j} \in R_{train}\}$ is the set of users who rated j.

Random Gaussian user: This baseline is computed by specifying that the ratings for each item j follow the normal distribution with parameters μ and σ approximated on R_{train}. The value of ir_j for any j is generated from this distribution:

$$ir_j \sim \mathcal{N}(\mu, \sigma^2) \tag{3}$$

Random uniform user: This baseline is computed by specifying that the ratings for each item j follow the discrete uniform distribution with r_{min} and r_{max} being the extreme rating values. Hence:

$$ir_j \sim \mathcal{U}\{r_{min}, ..., r_{max}\} \tag{4}$$

For example, if a rating can be one to 5 stars, then $r_{min} = 1$ and $r_{max} = 5$.

We use the term of a baseline user to define the concept of *reliable similarity*, which is based on a significance test.

3.2 Reliable Similarity between Users

To define *reliable similarity*, we begin with an arbitrary similarity function $sim()$. We will specify $sim()$ explicitly later.

Definition 1 (Reliable similarity). *Let $sim()$ be a similarity function, and let u_B be the baseline user learned on R_{train}. We define the "reliable similarity" sim_{rel} between a user u_A , for whom recommendations must be formulated, and an arbitrary other user x as*

$$sim_{rel}(u_A, u_B, x) = \begin{cases} sim(u_A, x) & , \text{ if } sim(u_A, x) \gg sim(u_A, u_B) \\ 0 & , \text{ otherwise} \end{cases} \tag{5}$$

where we use the symbol $\gg$ for "significantly greater than". User x is "reliably similar" to u_A if $sim_{rel}(u_A, u_B, x) > 0$.

Checking for significance. We implement the "significantly greater than"-test of Def. 1 with help of the Hoeffding Inequality [5]:

$$Pr(\widehat{X} - \overline{X} \geq \varepsilon) \leq exp(\frac{-2n\varepsilon^2}{R^2}) \tag{6}$$

The Hoeffding Inequality quantifies the probability that the deviation of an observed average $\widehat{X}$ from the real average $\overline{X}$ of a random variable X is greater than or equal to ε. It takes as inputs the range R of the random variable and the number of observed instances n. The Hoeffding Inequality is independent of any probability distribution, however, it is thereby more conservative than other distribution-specific bounds [3]. The inequality can be transformed into the Hoeffding Bound that specifies the maximal allowed deviation ε given a confidence level of $1-\delta$:

$$\widehat{X} - \overline{X} < \varepsilon \;, where\ \varepsilon = \sqrt{\frac{R^2 \cdot \ln(1/\delta)}{2n}} \tag{7}$$

We apply the Hoeffding Bound to ensure that the true similarity between two users is inside the ε-vicinity of the observed similarity. In particular, let u_1, u_2 be two users. Then, $\widehat{X}$ stands for the observed difference in similarity between them and $\overline{X}$ stands for the difference of their true similarities, thereby demanding that the similarity function is an average, as dictated in [5].

Definition 2 (Similarity Function for Significance Testing). *Let u_1, u_2 be two users and let $I_{co\text{-}rated}(u_1, u_2)$ be the set of items that both have rated. Then, the similarity between u_1, u_2 is the following average (for a rating scale between 0 and 1, otherwise normalization is required):*

$$sim(u_1, u_2) = 1 - \frac{\sum\limits_{j \in I_{co\text{-}rated}(u_1,u_2)} |r_{u_1 j} - r_{u_2 j}|}{|I_{co\text{-}rated}(u_1, u_2)|} \tag{8}$$

On the basis of this similarity function, we state with confidence $1-\delta$ that the non-observable true average similarity, denoted as $\overline{sim}(u_1, u_2)$, is within the ε-vicinity of the observed average similarity, denoted as $\widehat{sim}(u_1, u_2)$. The bound ε represents the uncertainty of the observed information. The fewer co-rated items we have for the two users, the larger is the possible deviation from the true unobserved values. This is captured by the number of observations n, which is here the cardinality of $I_{co\text{-}rated}(u_1, u_2)$. The smaller the value of n, the larger the bound ε (cf. Ineq.7) for a given confidence $1-\delta$.

The use the Hoeffding Bound in the significance test in Def. 1 means the following: when we observe that $\widehat{sim}(u_A, x) > \widehat{sim}(u_A, u_B)$, we want to state with confidence $1-\delta$ that $\overline{sim}(u_A, x) > \overline{sim}(u_A, u_B)$, subject to a bound ε.

To this purpose, we first need to ensure that the same number of observations is used for both the observed similarity $\widehat{sim}(u_A, x)$ and for the observed similarity $\widehat{sim}(u_A, u_B)$. Evidently, the set of co-rated items between u_A, u_B is the set of items rated by u_A, since the baseline user u_B has a rating for every item. Therefore, for each user x, whom we consider as potential neighbor of u_A, we compute $sim(u_A, u_B)$ on $I_{co\text{-}rated}(u_A, x)$ rather than on $I_{co\text{-}rated}(u_A, u_B)$. Thus, the number of observations is fixed to $n = |I_{co\text{-}rated}(u_A, x)|$.

In the left part of Figure 1, we depict the relative positions of $\widehat{sim}(u_A, x)$, $\overline{sim}(u_A, x)$, $\widehat{sim}(u_A, u_B)$, $\overline{sim}(u_A, u_B)$ in a case where both the observed and the true average similarity between u_A, x is larger than the corresponding values for u_A, u_B. In the right part of Figure 1, we depict again the relative positions in a case where the observed average similarity between u_A, x is larger than the observed similarity between u_A, u_B, but the true similarity between u_A, x is smaller than the true similarity between u_A, u_B. Clearly, this is undesirable.

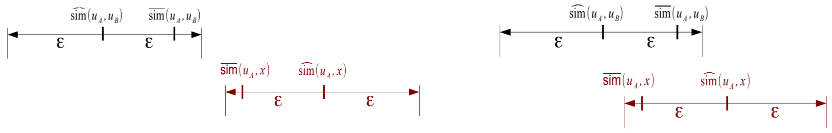
$\widehat{sim}(u_A, u_B)$
$\overline{sim}(u_A, u_B)$
ε
ε
$\overline{sim}(u_A, x)$
$\widehat{sim}(u_A, x)$
ε
ε
$\widehat{sim}(u_A, u_B)$
$\overline{sim}(u_A, u_B)$
ε
ε
$\overline{sim}(u_A, x)$
$\widehat{sim}(u_A, x)$
ε
ε

Fig. 1: Relative positions of the observed similarity between u_A, x and between u_A, u_B and true similarity within the ε-vicinity of the corresponding observed similarity; the observed similarities on the left allow the conclusion that the true similarity between u_A, x is larger than the true similarity between u_A, u_B; the observed similarities on the right lead to an erroneous conclusion, though.

Hence, we need a bound ϱ such that it holds:

$$\text{if } \widehat{sim}(u_A, x) - \widehat{sim}(u_A, u_B) > \varrho \text{ then } \overline{sim}(u_A, x) > \overline{sim}(u_A, u_B)$$

To ensure with confidence $1 - \delta$ that $\overline{sim}(u_A, x) > \overline{sim}(u_A, u_B)$ for any values of $\widehat{sim}(u_A, x)$, $\widehat{sim}(u_A, u_B)$, we consider the extreme case, where $\widehat{sim}(u_A, x)$ is smallest and $\widehat{sim}(u_A, u_B)$ is largest, i.e. $\overline{sim}(u_A, x) = \widehat{sim}(u_A, x) - \varepsilon$ and $\overline{sim}(u_A, u_B) = \widehat{sim}(u_A, u_B) + \varepsilon$. Then, to ensure that $\overline{sim}(u_A, x) > \overline{sim}(u_A, u_B)$, following must hold:

$$\left(\widehat{sim}(u_A, x) - \varepsilon\right) - \left(\widehat{sim}(u_A, u_B) + \varepsilon\right) > 0 \text{ i.e. } \widehat{sim}(u_A, x) - \widehat{sim}(u_A, u_B) > 2\varepsilon$$

This means that $\varrho = 2\varepsilon$. Thus, we specify that:

$$sim(u_A, x) \gg sim(u_A, u_B) \iff \widehat{sim}(u_A, x) - \widehat{sim}(u_A, u_B) > 2\varepsilon \tag{9}$$

Definition 3 (Reliable Neighbourhood). *Let u_A be an active user. Subject to Def. 1, the similarity function of Eq. 8 and the two invocations of the Hoeffding Bound, we define her reliable neighbourhood as:*

$$relNeighbourhood(u_A, \theta) = \{x \in U | sim_{rel}(u_A, u_B, x) > \theta\} \tag{10}$$

where U is a set of users and the similarity threshold θ is applied on reliable neighbours only. All unreliable neighbours are excluded, even if their similarity to u_A is larger than θ.

3.3 Algorithms

Algorithms 1 and 2 show a pseudocode of our extensions to collaborative filtering. Algorithm 1 computes a neighbourhood of an active user u_A using our method of checking the reliability of neighbours `isReliableNeighbour`, presented in Algorithm 2. This method requires two parameters: θ is a similarity threshold, also used in conventional CF, and δ controls the confidence of the Hoeffding Bound used for checking the reliability. Since the criterion of the reliable similarity is much stricter than the conventional similarity, it can happen that no neighbours

Algorithm 1 Reliable CF

```
reliableNeighbourhood($u_A$) ← {}
$u_B \leftarrow initializeBaseline(R_{train}, baseline\_type)$
for all $\{x \in U | x \neq u_A\}$ do
  $x$ reliable ← isReliable($u_A, u_B, x$)
  if $x$ reliable then
    reliableNeighbourhood($u_A$).add($x$)
  end if
end for
if reliableNeighbourhood($u_A$) $== \emptyset$ then
  abstain or recommend most popular items
else
  for all item $i \in missingValues(u_A)$ do
    predict $\widehat{r}_{u_A,i}$(reliableNeighbourhood($u_A$))
  end for
  return top-k ranked items
end if
```

Algorithm 2 isReliable(u_A, u_B, x)

```
$x$ reliable ← true
if $sim(u_A, x) \leq \theta$ then
  $x$ reliable ← false
end if
$\varepsilon$ ← computeHoeffdingBound($\delta$, Range, numberCoRatings($u_A, x$)) (cf. Ineq. 7 and Eq. 10)
if $\widehat{sim}(u_A, x) - \widehat{sim}(u_A, u_B) \leq 2\varepsilon$ then
  $x$ reliable ← false
end if
if $x$ reliable then
  return true
else
  return false
end if
```

for an active user can be found at all. For this case we also adjusted the conventional CF algorithm. Our method can either abstain from recommending any items until more information about the given user is collected, or it provides non-personal recommendations e.g. the most popular items from the trainings dataset. We state that it is beneficial to make fewer, but reliable recommendations, than to recommend items that will cause a negative attitude or a distrust of the user towards the recommender.

4 Experiments

We evaluate our method on the datasets MovieLens (100k), Flixter, Netflix and Epinions [8], comparing it to: a conventional user-based collaborative filtering recommender with cosine similarity, denoted as CF, to the method by Bell et al. called "shrinkage"[2] and to "significance weighting" by Herlocker et al.[4]. Since our goal is to compare different ways of building a neighbourhood, we implemented only the weighting schemas from the methods described in [2] and [4] and coupled them with the conventional CF algorithm. To ensure a fair comparison, all methods use the same core CF algorithm with no further extensions, so that only the way they build and weight their neighbourhoods differs.

We term our method "Hoeffding-CF", abbreviated hereafter as H-CF. We consider one variant of our method per type of baseline user, denoted as H-CF_Gauss (Gaussian user), H-CF_Uniform (uniform user) and H-CF_Avg (average user). To optimise the parameters of the methods we run multiple experiments using a grid search over the parameter space. Since the number of experiments in the grid search is high, we chose a sample of users per dataset,

Table 2: Samples of users on four datasets.

Dataset	total Ratings	sampled ratings
Flixter	572531	59560
MovieLens 100k	100k	100k (no sampling)
Netflix	100 M	216329
Epinions	550823	165578

taking over all their ratings. The evaluation settings are detailed below. Further information regarding datasets and our samples is summarized in Table 2.

4.1 Evaluation Settings

As basis for our evaluation we use (a) the RMSE of the predictions made by each method, and (b) the number of cases where the method encounters an empty neighbourhood and cannot make a neighbourhood-based prediction; this is denoted as *Missing Predictions*. However, a prediction is still provided using a fallback-strategy explained later. We further compute the *Average Neighbourhood Size*, the average size of non-empty neighbourhoods built by each method.

It is evident that the RMSE values for the three variants of our method are not directly comparable, because the value of *Missing Predictions* varies among the methods. Hence, we refine RMSE into following measures:

- *Neighbourhood-based RMSE*: the RMSE of the predictions made using the neighbourhoods of the users; limited to users with non-empty neighbourhoods (abbreviated hereafter as CF-RMSE)
- *Fallback-strategy RMSE*: the RMSE of the predictions made using the fallback strategy; limited to users with empty neighbourhoods
- *Global RMSE*: total RMSE by both *Neighbourhood-based RMSE* and *Fallback-strategy RMSE*

As fallback strategy we use the recommendation of the most popular items not rated by the active user. The impact of this strategy is encapsulated in *Fallback-strategy RMSE.*

For the variants of our method, we vary δ: the lower the value, the more restrictive is the confidence level of the Hoeffding Inequality and the less users are considered reliably similar to a given user. Hence, we expect that a decrease of δ will negatively affect the *Average Neighbourhood Size* and the *Missing Predictions*. For shrinkage and significance weighting we also optimize β and γ.

We further consider different similarity threshold values. As we have seen in section 1, setting the threshold to a high value is not adequate for prohibiting recommendations on the basis of unreliable neighbourhoods. It must be noted that the CF may also fail to build neighbourhoods for some users, if the threshold is set very restrictively. In total, we performed more than 250 experiments, all of which were evaluated using 5-fold cross validation.

4.2 Results

In Table 3, we present our results on each of the four datasets. For each of the methods we present only the best value found by the grid search in course of the optimization. The symbol ”— ” indicates that there are no applicable values for this position (e.g. delta is not applicable for the CF). The sizes of the neighbourhood in Table 3 is seemingly high, however, these are the values found as approximatively optimal by the grid search.

The best result on on the Movie Lens 100k dataset was achieved by our method (1^{st} row in the Table) with a setting of $\delta = 0.999$, a uniform baseline user, and distance threshold of 0.25. The best value of global RMSE was 0.9864. The best result achieved by the conventional CF was 1.0207 (5th row in the table). This is a stable improvement verified using the 5-fold cross validation. Shrinkage and significance weighting yielded a result close to the conventional CF. When we compare our method with e.g. shrinkage with respect to the average neighbourhood size (row 1 and 3), then we notice an essential reduction from ca. 898 to 447 users. This means that our method reduced the neighbour-

Table 3: Results on four benchmark datasets sorted with respect to global RMSE (lower values are better) and grouped by the dataset.

Row	Method	Distance Threshold	Setting	Missing Predictions	avgNeigh-borhoodSize	global RMSE	CF-RMSE	fallback-RMSE
MovieLens 100k								
1	H-CF_Uniform	0.25	$\delta = 0.999$	2905	447	0.9864	0.9683	1.4929
2	H-CF_Gauss	0.25	$\delta = 0.95$	3229	259.55	0.9875	0.9684	1.4693
3	Shrinkage	0.2	$\beta = 500$	215	898.08	1.0192	1.0192	—
4	Sig. Weighting	0.2	$\gamma = 200$	215	898.08	1.0192	1.0192	—
5	CF	0.2	—	215	898.08	1.0207	1.0207	—
6	H-CF_Avg	0.4	$\delta = 0.999$	13079	132.38	1.0321	1.0390	0.9839
Flixter (sample of 1000 users)								
7	H-CF_Gauss	0.8	$\delta = 0.95$	7047	78.14	1.0149	1.0133	1.0381
8	H-CF_Avg	0.8	$\delta = 0.95$	49918	5.84	1.0221	1.1355	0.9969
9	H-CF_Uniform	0.4	$\delta = 0.95$	4357	241.3580	1.0549	1.0532	1.1576
10	CF	0.7	—	3998	442.7564	1.0856	1.0856	—
11	Shrinkage	0.7	$\beta = 50$	3998	442.7564	1.0872	1.0872	—
12	Sig. Weighting	0.7	$\gamma = 50$	3998	442.7564	1.0889	1.0889	—
Netflix (sample of 1000 users)								
13	H-CF_Gauss	0.2	$\delta = 0.95$	13601	199.66	0.9619	0.9511	1.1551
14	H-CF_Uniform	0.2	$\delta = 0.95$	11171	382.74	0.9622	0.9529	1.1849
15	H-CF_Avg	0.2	$\delta = 0.999$	60394	96.94	1.0075	1.0225	0.9669
16	Shrinkage	0.2	$\beta = 200$	4023	916.2519	1.0210	1.0210	—
17	Sig. Weighting	0.2	$\gamma = 100$	4023	916.2519	1.0214	1.0214	—
18	CF	0.2	—	4023	916.2519	1.0233	1.0233	—
Epinions (sample of 10 000 users)								
19	H-CF_Avg	0.3	$\delta = 0.5$	165578	0	1.0074	—	1.0074
20	H-CF_Gauss	0.8	$\delta = 0.5$	164948	0.2770	1.01100	1.3964	1.0106
21	H-CF_Uniform	0.4	$\delta = 0.5$	159842	1.5113	1.0279	1.3215	1.0109
22	CF	0.7	—	113117	461.39	1.2843	1.2843	—
23	Shrinkage	0.7	$\beta = 50$	113117	461.39	1.2894	1.2894	—
24	Sig. Weighting	0.7	$\gamma = 100$	113117	461.39	1.2907	1.2907	—

hoods by 451 users on average and still performed better than the conventional CF. Regarding the baseline users on the Movie Lens dataset, the best results were achieved by the uniform random baseline. The average user baseline led to small neighbourhoods. This can be explained by the fact that many users in the MovieLens dataset are similar to the average user. Using this baseline makes the differences between user vectors insignificant and, consequently, many of the users are not considered as reliable neighbours to the active user.

If no reliable neighbours of an active user can be found, then computing a rating prediction is not possible. We counted the occurrences of this case in our method (column "missing predictions"). In this situation a fallback-strategy (e.g. popular items) takes over the task of providing a recommendation (prediction error is included in global RMSE). We observed that those cases become more frequent when delta is low. This causes a more extensive pruning behaviour of our method, because more neighbourhoods are considered unreliable. If we allow our method to abstain from recommendation instead of using the fallback-strategy the improvement of RMSE is even higher (0.9683; row 1, column CF-RMSE). Also the conventional CF, shrinkage and significance weighting exhibit some missing predictions. They are caused by either new users or new items that are not known from the training dataset.

We performed similar experiments on a random sample of 1000 users on the Flixter dataset. Also on this dataset our method achieved the best *globalRMSE* value of 1.0149 this time using a Gaussian baseline. The conventional CF (row 10) yielded a value of 1.0856 using neighbourhoods bigger by 365 users on average. Shrinkage and significance weighting were not able to outperform CF.

Also on a random sample of 1000 users from the Netflix dataset our method outperformed other approaches with respect to global RMSE, reaching the level of 0.9619 using the Gaussian user baseline. When abstention was allowed the improvement was even more substantial and reached the level of 0.9511, compared to e.g. shrinkage with 1.0210 (row 16). Also here we observed an essential reduction of the neighbourhood cardinality from ca. 916 by the shrinkage method down to ca. 200 by our approach. This proves that our approach selects the reliable neighbours, who are more informative for the preferences of an active user than the competitive methods.

The last dataset we performed our experiments on is the (small) Epinions dataset (cf. Table 3). Here our method clearly dominated the conventional CF. Hoeffding-CF achieved an RMSE of 1.0074 compared to 1.2843 by the conventional CF. Significance weighting and shrinkage performed worse than CF. Our approach recognized unreliable neighbourhoods and switched from the neighbourhood-based recommendation to the fallback-strategy that performs better on this dataset (cf. the columns *CF-RMSE* and *fallback-RMSE*). The average number of neighbours in the first row shows that the neighbourhood was limited to the minimum and this yielded the best result. Differently than on the other datasets, here the average user baseline performed the best. The statement about its strictness in the significance testing still holds. This very strictness was beneficial on this dataset. In row 19 we see that the neighbour-

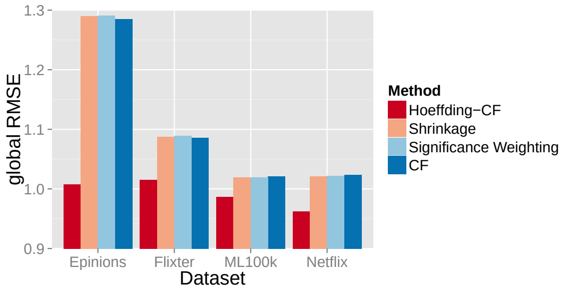

Fig. 2: Best results achieved by each method. Lower values of global RMSE are better. Our method, Hoeffding-CF, achieves best results on each dataset.

hood was reduced to 0 i.e. there was no neighbourhood-based recommendations. All recommendations were provided by the fallback-strategy that, in this case, performed better.

4.3 Summary of Findings

Our experiments show that Hoeffding-CF is capable of recognizing unreliable neighbourhoods and selecting neighbours that are informative for the preferences of an active user. It outperformed the conventional collaborative filtering, shrinkage and significance weighting on all datasets. When abstention from providing recommendations was allowed, the improvement in terms of RMSE was often even more substantial. All of the best results were achieved using a smaller neighbourhood than in case of conventional CF and remaining approaches. A summary of the best results by each method is presented in Figure 2.

We also observed that the parameter delta plays an important role in finding the optimal results. The lower its value, the stricter is the testing of the neighbourhood and the smaller is the average neighbourhood. Consequently, the number of predictions provided by the baseline method rises. The optimal value of delta varies across different dataset around 0.95. Cross-validation can be used for tuning on each dataset.

The choice of the baseline user has also an effect on the performance. We observed that the random-based user (Gaussian and uniform baseline) perform better than the average user baseline on most datasets. The reason for that is that many users are similar to the average user, so it is difficult to identify a user that is significantly more similar to a given user than the average. Hence, when the average user is the baseline, each user has only a few significant neighbours. On the Epinions dataset, however, this led to an improvement of accuracy.

5 Conclusions

We investigated the problem of neighbourhood-based recommendations when the similarity between users cannot be fully trusted. This problem does not emanate solely from data sparsity: even users with many ratings may be uninformative. We introduced the concepts of *baseline user* and of *reliable similarity*, and we use statistical testing to select, for a given user, those users who are informative, truly similar neighbours to her, ignoring users that do not contribute more information than the baseline user. To ensure efficient computation, we use the Hoeffding Inequality for statistical testing.

Experiments on real datasets show that the use of reliable similarity improves recommendation quality: our method is superior to the conventional CF, shrinkage and significance weighting on all datasets, while the superiority in the forth dataset is mainly owed to a good performance of the fallback-strategy rather than to neighbourhood-based recommendations. Our method outperforms other approaches, although it uses smaller neighbourhoods. This means that the reliability, rather than the size of a neighbourhood is decisive for good predictions.

Our next task is comparing our method to approaches presented in [6], [1] and also formulating reliable recommendations for matrix factorization, which is a popular method in recommenders, but relies on activities of heavy raters. Are heavy raters informative for a specific user, though? We intend to investigate this issue by extending our concepts of baseline user and reliable similarity towards reliable matrix-factorization-based recommenders.

References

1. L. Baltrunas and F. Ricci. Locally Adaptive Neighborhood Selection for Collaborative Filtering Recommendations. volume 5149 of *LNCS*. Springer, 2008.
2. R. Bell, Y. Koren, and C. Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *13th ACM SIGKDD*, 2007.
3. P. Domingos and G. Hulten. Mining High Speed Data Streams. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2000.
4. J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *ACM SIGIR*. ACM, 1999.
5. W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.*, 58:13–30, 1963.
6. R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR '04*, pages 337–344, New York, NY, USA, 2004. ACM Press.
7. H. Ma, I. King, and M. R. Lyu. Effective missing data prediction for collaborative filtering. In *ACM SIGIR*, SIGIR '07.
8. P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI 2006 Workshop on Recommender Systems*, pages 29–33+, 2006.
9. F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.
10. G. Shani and A. Gunawardana. Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*.