Being Confident about the Quality of the Predictions in Recommender Systems

Sergio Cleger-Tamayo[1], Juan M. Fernández-Luna[2], Juan F. Huete[2], and Nava Tintarev[3]

[1] Departamento de Informatica Universidad de Holguin, Cuba
sergio@facinf.uho.edu.cu
[2] Departamento de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR Universidad de Granada
{jmfluna,jhg}@decsai.ugr.es
[3] University of Aberdeen, UK
n.tintarev@abdn.ac.uk

Abstract. Recommender systems suggest new items to users to try or buy based on their previous preferences or behavior. Many times the information used to recommend these items is limited. An explanation such as *"I believe you will like this item, but I do not have enough information to be fully confident about it."* may mitigate the issue, but can also damage user trust because it alerts users to the fact that the system might be wrong. The findings in this paper suggest that there is a way of modelling recommendation confidence that is related to accuracy (MAE, RMSE and NDCG) and user rating behaviour (rated vs unrated items). In particular, it was found that unrated items have lower confidence compared to the entire item set - highlighting the importance of explanations for novel but risky recommendations.

1 Introduction

There are numerous resources available on the WWW offering their products to the users, such as Amazon, Netflix, eBay, YouTube or Last.fm. In these systems users have a selection of an unsurveyably large number of items to buy, listen to, or watch. From early on *Recommender Systems (RS)* emerged as a possible solution for this information overload problem, helping users to find relevant items [22] in different ways. Two of the most common actions performed by RS are rating prediction and finding (a ranked list of) good items [7]. For *rating prediction* the system has to predict how much a user would like an unseen item. In this case, the usual output is a rating, on a given scale (for instance, from 1 to 5 stars). For the second task of *finding good items*, this is more of a ranking problem: the RS has to find the best items to be recommended to the active user, ranking the results by taking into account the predicted ratings.

Independently of the task that a RS is designed for, and in order to provide recommendations, these systems have to learn from users' behavior with the aim of discovering their preferences. Often it is hard to make reliable predictions

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 411–422, 2013.

for some users or items. There are several reasons why this maybe the case, such as data sparsity or it may because there is a lot of noise or variation in the information available [18]. An explanation such as *"I believe you will like this item, but I do not have enough information to be fully confident about it."* may mitigate the issue. Explanations can help the user to differentiate between sound and inadequate recommendations, understand why a bad one has been made and even compensate for it. In our previous research [3], we have found that, after observing an explanation, the users modified the proposed ratings frequently (around 35% of the time) and that by means of these actions the recommendations can be improved. If explanations help win users trust, this may have longer term benefits [19].

On one hand, users may appreciate that the system is "frank" and admits that it is not confident about a particular recommendation. On the other hand, it has been found that bringing low confidence recommendations to the attention of experienced users can lower their satisfaction of the system, most likely because it alerts these users to the fact that the system might be wrong [17]. Therefore there may be merit in detecting high confidence recommendations, and using explanations that pinpoint low confidence sparingly. Therefore, this paper aims to answer the following question:

Is it possible for a RS to be able to automatically determine whether a prediction is reliable (has high confidence) or not?

The first problem that we have to tackle is to define what a reliable (confident) prediction means. In the literature unconfident recommendations are usually associated with those situations in which there is not too much data to support it (see Section 2). Note that this definition is related to the item (or even the user), but not to the particular value predicted (for instance, 3.4 stars). In this paper we explore a different alternative that relates reliability to the error in the prediction. In this sense, we will say that a prediction is reliable if we think that the expected error (the difference between the predicted and the ground rating) is under reasonable threshold δ. In order to determine that a prediction is reliable a machine learning approach, looking at the properties in the data used to explain recommendations for a set of observed items, will be used.

This paper is organized as follows: Section 2 presents some related research on recommendation confidence and explanations; the next part of the paper, Section 3, describes our approach to learn from the explanations in order to make decisions about how the reliability of the prediction. The experimental design and its results are showed in Section 4. The paper concludes with some remarks and further research directions, in Section 5.

2 Related Work

In this section we discuss related work in the areas of recommendation confidence and explanations of recommendations.

Confidence: A great deal of previous research has focused on increasing the accuracy of recommendations, but the growing consensus in the recommender

systems community is that accuracy metrics are no longer enough to evaluate the quality of recommendations [16]. Recommendation quality as perceived by the user is however affected by other varied factors such as the effort, usage and social context, and diversity [10,13,20].

Another area that is less understood, is prediction confidence [17]. This differs from prediction strength, which can be defined as the degree to which the item is recommended to the user, i.e. 4.5 stars out of 5. *Prediction confidence* on the other hand, is based on the supporting knowledge for this prediction either in terms of what is known about the user or the item.

For example, predictions for items with many ratings are likely to be more accurate than those with few ratings (inverse to the item cold start problem) [17]. Likewise, it is hard to make predictions for users who have not yet entered many ratings (the user cold start problem) and, in the same way, polarizing items may be more risky recommendations. Also, another aspect that should affect confidence is to consider whether a recommendation comes from trustable users or not (trust-aware RSs) [15].

In previous studies of confidence, a simplified metric of the number of data points, such as the number of ratings given to a movie, has been used [14,17]. In [8] an explanation interface that used a measure of confidence is proposed, but did not disclose how it was computed. This particular study only looked at a single high confidence value, but this explanation interface did not significantly affect the users' perceived likelihood of watching a movie. This suggests that (positive) confidence information can be used without detrimental effects on intended purchase behavior.

This paper proposes a novel way of classifying confident predictions in a collaborative filtering approach, which mainly differs in the type of information used. Our approach considers both: a) the predicted rating for an item and how it was computed and b) the errors found when analyzing similar observed recommendations. Thus, if in similar cases the predictions were not reliable (large errors were obtained), we will be unconfident about this recommendation.

Explanations: Despite the popularity of RSs, few supply explanations of their recommendations. These systems are usually seen as black boxes in which there is no other choice but to simply accept the recommendations [8]. A possible solution is the inclusion of explanations facilities in the recommendation processes. Previous investigations of explanations in RS were focused on showing the effects of explanations on users under several aims, such as increasing effectiveness (helping the user to make good decisions) or trustworthiness (increase the user trust in the recommendations) [1,3,5,6,8,12,19,24]. One of the roles of explanations concerning prediction confidence could be to gain the users' trust by setting the correct level of expectation from the user. In one qualitative study it was found that users felt that explanations could increase their acceptance upon viewing, and save potential disappointment [23].

However, by explicitly calling out the possibility that not all recommendations are of the same accuracy, a seed of doubt may be planted into users mind about all the recommendations they receive. One paper which surveyed the roll

of confidence displays in recommender systems, found that the introduction of a confidence display had an overall positive effect on the satisfaction with the system of its users compared to a control. However, training on the confidence display had an adverse effect on experienced users (but not novices) [17]. Explanations similar to a confidence display will bring the issue of confidence to the forefront, but may also offer the user a sense of control in terms of improving the predictions given to them.

Another study evaluated the effects of certainty rating on perceived transparency, and acceptance of the system and its recommendations. In this study, these explanations did not cause an increase in any of these factors. However the authors also stated that these explanations where not always understood by users, and suggested that would have been better to use terms such as "sure" and "unsure" rather than supplying confidence as a percentage e.g. "50.0% sure" [6]. It is also true that confidence based explanations may be better suited at improving the trustworthiness of a recommender system rather than its transparency, or persuasive capabilities.

In the seminal study [8] evaluating 21 explanation interfaces, the most persuasive interface was histogram with the following text *"the system suggests 3 stars because it has been rated by other similar users as ..."*. In this explanation we can see, for instance, that the number of neighbors that rated the target item with 1,2,3,4 and 5$\star$ are $\{3, 5, 5, 4, 3\}$, respectively. These explanations are valuable because they indicate that in some way *how* the predicted rating is computed (which is related to the transparency of recommendations) but also they could give some information about the quality of the prediction, increasing the user's confidence in the system (related to trust).

This paper describes an approach that (among other factors) considers variability in user ratings, which can be used to display a histogram-based explanation such as the one described above. The next sections describe the model used for predicting confidence, and how it is related to recommendation accuracy.

3 Learning from Rating Patterns

In our approach we will try to classify a recommendations as confident or not using information gathered from the properties of the data used while explaining a set of already known items. But, the given explanation strongly depends on the used recommendation model. Several recommending strategies can be found in the literature [22], but in this paper we shall focus on a nearest-neighborhood-based collaborative filtering algorithm [4] which computes the predictions by considering how similar users rated a target item. In this model, two main steps can be considered: a) Neighborhood selection: Among all the users who rated that target item, I_t, select the most similar ones with respect to the active user preferences, $N_t(a)$. b) Rating prediction: computes the predicted rating, $\hat{r}_{a,t}$, as a weighted combination that takes into account the rating given by these users to the target item.

There are several possibilities to obtain a given prediction and the user can view a histogram-based explanation in order to determine its reliability. For

example, let us assume a prediction of 3⋆ for two different items, say I_1 and I_2, and two different explanations (histograms) expressing that twenty neighbors rated item I_1 as $\{4, 4, 4, 4, 4\}$ and item I_2 as $\{0, 0, 20, 0, 0\}$. In this case, the user should have a greater certainty about the prediction for the item I_2 than item I_1, since all twenty neighbors gave the same rating of 3 stars. There are several likely factors that lead a user to be confident (or not) with a prediction. Our hypotheses in this paper are two fold: First, that confidence depends on the user, the target item but also it depends of his/her past experiences with the system; and second, that we can analyze these experiences with the aim of predicting the confidence of recommendations.

We propose the use of machine learning strategies to check our intuition. But, which data are necessary for training? One possibility might be to force this neighborhood, $N_t(a)$, to predict already known ratings (those for the observed items, $\mathcal{O}$). So for each item I_o in $\mathcal{O}$ we know (see the example in Table 1) the rating given by the user ($r_{a,o}$) and the rating predicted by the neighborhood ($\hat{r}_{a,o}$). Therefore, we can state whether we obtain a confident prediction or not. In this paper we shall consider that if $abs(r_{a,o} - \hat{r}_{a,o}) \leq 0.5$ we obtain a confident prediction, i.e. $C = yes$, otherwise it is unconfident $C = no$. Also, for each predictions we have its histogram-based explanation. Note that in this case it is possible that there exists some neighbors who did not rate the observed item[1] I_o so, we decided to include an extra row in the explanation representing the how many of them did not rate I_o (denoted with the value #0).

Focusing on the histogram, we believe that their raw values can not help to determine the confidence of the prediction. For example, knowing that a neighbor rated other items with 2⋆ or 3⋆ does not give us information about the confidence of this prediction. This value depends on whether the item fulfils the particular tastes of this neighbor, or not. But what we shall consider relevant to predict the quality of a recommendation is the fact that there exists some agreement among the different ratings of the neighborhood.

In order to measure the lack of agreement we shall consider the entropy measure, defined as $H(o) = \sum_j p(r_j) \log_2(1/p(r_j))$, where $p(r_j)$ is the probability that the neighbors used the rating r_j for this item[2]. In this case, the more ratings concentrated in a specific value, the lower the entropy (the greater the agreement). Also, entropy takes the maximum value when all the ratings are equally likely (uncertainty is highest when all possible events are equiprobable).

From all these data, we can obtain the inputs for a machine learning algorithm. The dependent variable is the confidence in the predictions, C, and $\hat{r}_{a,o}$, #0 and $H(o)$ are the features selected as the independent variables. The inclusion of #0 as learning feature will allow us to detect higher errors in the training set that might be due to large values of #0. In order to predict the confidence of a prediction, we use the decision tree-based $J48$ classifier [21] since it has been

[1] Because the neighbors were selected among those who rated the target item I_t.

[2] We opted to consider the fact that a neighbor did not rate the item as a piece of information about the uncertainty in the recommendation process, and will be included as an attribute in the computations of the entropy values.

Table 1. Example. I_o - item number, $r_{a,o}$ - actual rating, $\hat{r}_{a,o}$ - predicted rating, #0 - rating not given, $x\star$ - number of neighbors who rated the item with x stars, H(o) - entropy of neighborhood ratings, C - confidence prediction (yes or no)

			Histogram							
I_o	$r_{a,o}$	$\hat{r}_{a,o}$	#0	**1⋆**	**2⋆**	**3⋆**	**4⋆**	**5⋆**	$H(o)$	C
i_1	3	3,3	2	2	3	8	6	1	2,01	y
i_2	5	4,0	2	0	0	0	20	0	0	n
i_2	4	3,1	2	0	0	16	4	0	0,72	n
..	.	...	..	..	..	..	..	..	...	.
i_m	5	4,1	0	1	2	2	12	5	1,49	n

successfully applied for solving classification problems in many applications being also easily interpretable by the users. J48 builds decision trees[3] by identifying the attributes that discriminate the various instances most clearly (using the concept of information entropy). After learning the classifier, it will be finally used to predict the confidence on the prediction for the target item , $\hat{c}_t$, taking into account that we know $\hat{r}_{a,t}$, #0 and $H(t)$.

As a final remark we want to say that, since usually a user did not rate a large number of items, this process does not require a high computational cost, being transparent for the user (it is a pre-visualization approach).

4 Experiments

We use on offline approach based on MovieLens 100K data set, which was collected by the GroupLens Research Project at the University of Minnesota and contains 100,000 anonymous ratings (on a scale of 1 to 5) of approximately 1,682 movies made by 943 MovieLens users, who joined MovieLens during the seven-month period from September 19th, 1997 through April 22nd, 1998.

The objective of the experiments are to measure the capability of our approach to determine the reliability of the predictions given by a RS. To validate our model, we have decided to divide the data into training and test sets (containing the 80% and 20% of the data, respectively) in such a way that i) no rating belongs to both training and test sets, and ii) all items in the test set have been rated in the training data. Also, for test purposes we have also included as test instances a set of unseen movies. Particularly, for each user in the test set we duplicate the number of items by randomly selecting movies among those which have not been rated by him/her.

With the aim of exploring the effect of the amount of knowledge supporting the predictions, we will also consider three different situations: The first one that considers those users that rated more than 100 items (large support, LS), the second one that considers users with a number of ratings between 40 and

[3] We used the implementation contained in Weka data mining toolkit, `http://www.cs.waikato.ac.nz/ml/weka`

Table 2. Comparison of prediction reliability R and confidence $\hat{c}$. ($R = yes$ if $abs(r - \hat{r}) \leq \delta = 0.5$). Considering three levels of support : large (LS, > 100 items), medium (MS, $40 - 100$) and small (SS, < 40)

	a) Global		b) LS		c) MS		d) SS	
Pred.	$\hat{c} = yes$	$\hat{c} = no$	$\hat{c} = yes$	$\hat{c} = no$	$\hat{c} = yes$	$\hat{c} = no$	$\hat{c} = yes$	$\hat{c} = no$
$R = yes$	5702	2841	4349	1969	963	591	390	281
$R = no$	2451	8614	1685	6732	489	1319	277	563

Table 3. Comparison of the accuracy for confident recommendations compared to all items, for three levels of support : large (LS, > 100 items), medium (MS, $40 - 100$) and small (SS, < 40)

	a) Global		b) LS		c) MS		d) SS	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
All	0.711	0.924	0.716	0.922	0.696	0.933	0.701	0.918
$\hat{c} = yes$	0.480	0.676	0.464	0.654	0.515	0.728	0.557	0.755

100 (medium support, MS) and finally those user with few ratings, less than 40 (small support, SS).

In order to test our hypothesis that it is possible to automatically determine the reliability of a prediction, we have designed our experiments around the two tasks mentioned in the introduction of this paper: rating prediction and finding good items.

4.1 Rating Prediction Task

In order to determine the performance of our model for this task we measure the capability of correctly classifying a prediction as confident or not. A prediction is considered *reliable* if its value differs from the actual value no more than δ which we have set to 0.5 (on a 1-5 scale). So, $R = yes$ if the error ($abs(r - \hat{r})$) is lower than 0.5, and unreliable, $R = no$, otherwise. To determine confidence we have to know the given rating, r, and therefore we will focus on those items in the test set that have been rated by the user.

Thus, for each item we know, on the one hand, if the predicted rating is reliable: R. On the other hand, we have the output of the algorithm that classifies this prediction as confident or not: $\hat{c}$. Table 2 presents the confusion matrix, where columns represent the predicted confidence, while each row represents the real error or reliability. Particularly, we report the number of false positives, false negatives, true positives, and true negatives.

From these tables, we see a high prediction rate for true positive and true negatives - our model made the correct prediction over 73% percent of the cases. The performance does however decreases with the number of ratings used to train the model: the accuracy is $0.75, 0.68$ and 0.63 when considering large, medium and small support, respectively. Now, we would like to focus on those cases in which the system is confident with the recommendation, representing the 42% of

the predictions (this percentage does not vary greatly with the support). Among the set of items classified as confident, $\hat{c} = yes$, we have that on average, the 70% of the times the system did a good job, i.e. the users would obtain a reliable recommendation (when analyzing the support the obtained values are 72%, 66% and 58% for large, medium and small training sizes, respectively). To compare, the number of reliable recommendations received by the user if we did not use confidence classification was 44%.

The previous analysis does not take into account the detailed information provided by the numerical values of the predictions. To do this we evaluate the accuracy using two standard metrics [10,7]: MAE, defined as the mean absolute error between the true and the predicted ratings and RMSE, that computes the root mean squared error, which gives greater weight to higher errors. Table 3 shows the obtained error when considering all the items in the test set and focusing on those items classified as confident. Thus, focusing on $\hat{c} = yes$, we can obtain improvements of about the 48% and 36% in terms of MAE and RMSE, respectively. Also in this case, the improvements has a positive relationship with the number of items rated by the user. In terms of MAE the improvements are 54%, 35% and 25% when having a large, medium and small training data, respectively. Similarly, in terms of RMSE we obtain improvements of 41%, 28% and 21%.

These results suggest that there exists a relation between accuracy and confidence, as well as lends support to the previous approaches which depended on the amount of information supporting the predictions. As such, our proposal might represent a valuable alternative to identify confidence in a recommendation framework. Low confidence predictions are not necessarily less relevant for the user and therefore should not automatically, be removed, but they are risky. Two alternatives are to give them lower weights (this situation will be explored in the next section) [14], or used to look for a different recommendation strategy/method in a hybrid system.

4.2 Finding Good Items Task

This second task aims to find the best items to be recommended to a given user. To start with we consider whether our proposal is useful for distinguishing between those items that are rated and those that are unrated by a user in the test set. A desirable outcome is that there are more unconfident predictions in the set of unobserved items, i.e. $Pr(\hat{c} = no|unrated) \geq Pr(\hat{c} = no|rated)$. Table 4 shows that there is a larger number of unconfident recommendations within the set of unobserved items, decreasing with the size of the training set. This suggests that using this measure of confidence can improve the precision for a user, i.e. he/she will receive a greater percentage of relevant (observed) items by filtering those where we can expect large errors, as it is shown in the last row of Table 4. These data allows us to infer that after analyzing the confidence of the recommendations the user can receive a greater percentage of relevant items.

Nevertheless, the final objective of a recommender system is to help the user to discover new items, so omitting those unconfident recommendations might

Table 4. Comparing error rates between rated and unrated items. The measures of precision is based on the whether an item was observed by a user or not, so for all the items we will achieve a precision of 0.5 since half of them were rated by the user.

	a) Global		b) LS		c) MS		d) SS	
	Rated	Unrated	Rated	Unrated	Rated	Unrated	Rated	Unrated
$Pr(\hat{c} = no \mid \bullet)$	0.584	0.774	0.590	0.799	0.568	0.722	0.558	0.645
	Original	Filtered	Original	Filtered	Original	Filtered	Original	Filtered
Precision	0.5	0.6481	0.5	0.6711	0.5	0.6085	0.5	0.5544

Table 5. Precision and recall for the top 10, 15 and 20, recommendations in a ranked list for four levels of support (Global, LS, MS, SS). These measures of precision and recall are based on the whether an item was observed by a user or not. A comparison is made between the baseline (BSL), hard ranking (HR) and soft ranking (SR). BSL - by ranking by predicted rating. HR - high confidence recommendations always before low confidence. Soft-reranking (SR) - high confidence recommendation before low confidence only if predicted rating is higher.

	a) Global			b) LS			c) MS			d) SS		
	BSL	HR	SR	BSL	HR	SR	BSL	HR	SR	BSL	HR	SR
Pr@10	0.725	0.721	0.740	0.810	0.859	0.877	0.734	0.696	0.740	0.539	0.522	0.540
Rc@10	0.528	0.516	0.533	0.238	0.252	0.246	0.648	0.610	0.651	0.937	0.908	0.938
Pr@15	0.727	0.732	0.744	0.780	0.824	0.810	0.659	0.613	0.658	–	–	–
Rc@15	0.568	0.552	0.574	0.343	0.358	0.356	0.859	0.803	0.857	–	–	–
Pr@20	0.700	0.711	0.744	0.752	0.787	0.783	0.609	0.574	0.607	–	–	–
Rc@20	0.617	0.607	0.626	0.440	0.452	0.456	0.933	0.883	0.930	–	–	–

not be the best option, particularly in those situations in which there are not many strong recommendations available. In this paper we will study a different alternative to filtering which consists of decreasing the importance of those unreliable recommendations in such a way that they score lower in a ranking. In order to evaluate this idea, we will consider as the baseline (BSL) the ranking obtained by sorting all the items in descending order of the predicted ratings. Thus, those items with greatest values will be placed in top positions.

In this paper we will explore two different alternatives to rerank those items classified as unconfident. The first one, that will be named hard-reranking (HR), promote any confident prediction to the top and unconfident recommendations will follow them, in both cases ordered by the predicted ratings. The second approach, named soft-reranking (SR), considers the value of the predicted rating, $\hat{r}$. Particularly, any unconfident item is located after those reliable predictions having the same round value of the predicted rating, $\hat{r}$. Thus, the ranking begin with all the items with a round prediction of 5$\star$, but confident predictions come before unconfident ones (in both cases ordered by the value of the predictions), followed by the items with round predictions of 4$\star$, 3$\star$, 2$\star$ and 1$\star$.

Table 6. Performance at Finding good items task for three filtering approaches: baseline (BSL), hard ranking (HR) and soft ranking (SR). Results are measured as-NDCG@10 and @20 for four levels of support (Global, LS, MS, SS).

	NDCG original ranking			NDCG Hard-Rerank			NDCG Soft-Rerank		
	ndcg	ndcg@10	ndcg@20	ndcg	ndcg@10	ndcg@20	ndcg	ndcg@10	ndcg@20
Global	0,5761	0,7602	0,7459	0,5704	0,7617	0,7538	0,5744	0,7667	0,7591
LS	0,5741	0,8308	0,7828	0,5778	0,8701	0,8164	0,5772	0,8499	0,8076
MS	0,5927	0,7721	0,6801	0,5801	0,7364	0,6420	0,5874	0,7668	0,6724
SS	0,5629	0,5961	–	0,5523	0,5791	–	0,5585	0,5954	–

Three metrics were used to evaluate the rankings: the precision and recall of the items located in the top k position of the ranking, with $k = 10, 15$ and 20. Note that the value k reflects that users normally only see a small set of k recommendations (see the results in Table 5), and the third metric is Normalized Discounted Cumulative Gain, NDCG [11], which takes into account if the system places items which have been previously rated by the user in higher positions. We present the results of this metric in Table 6, considering all the items in the test set (*ndcg* column) and also focusing on the top 10 and 20 recommendations (columns *ndcg*@10 and *ndcg*@20). A Friedman test was run to see if there are differences between the different ranking strategies (BSL, HR and SR). The tests allows us to conclude that there were a highly statistically significance ($p < 0.01$) when considering recall/precision metrics. With respect to the NDCG, the differences are highly significant for most of the cases, except for ndcg@20 with medium support (MS) where they are statistically significant ($p < 0.05$) and in the case of considering small support (SS) there is no statistical significance.

With respect to recall/precision values we can conclude that promoting confident predictions increases the number of relevant items in the top positions of the ranking, particularly for those users with large support (experienced users). In general, soft-reranking seems to perform more stably than hard-reranking in most cases. Hard-reranking performs worse when we do not have enough data for training (low support/novice users), but it has a good performance for large training sets. In this case, it seems that other relevant items are promoted to the top positions of the ranking, given the users the chance to inspect them.

With respect to NDCG similar conclusions can be obtained: our approach only successfully promotes observed items for users with a large number of rated items. In this case, HR performs better at finding observed items than SR. If we focus on users with low support (MS and LS), the original ranking is slightly better than SR, nevertheless SR does not seem to damage the results, so using SR could be a good overall approach.

5 Conclusions

The paper presents a simple and efficient method to determine the reliability of a prediction based on a classification approach, starting from the real and

predicted ratings, the entropy and the rating distribution as features, and the confidence of the prediction as class variable. The approach has been evaluated in terms of two RS tasks: prediction rating and finding good items.

Taking into account the results for the two tasks considered in the paper, we can conclude that the confidence criterion has the potential to improve user trust in the system, particularly for experienced users. There are two arguments to supports this: On the one hand, the user will receive better recommendations (in terms of accuracy) in top positions and, on the other hand, our approach increases the capability of the system to distinguish between observed and non-observed recommendations. For novice users, using confidence does not decrease the system performance. Finally, we want to note that our approach does not need large processing capacity, and therefore has the potential to be used in large scale systems.

With respect to future work, one of the direct research lines is to correct the predicted rating for those which were classified as not reliable, supporting that correction based on the way in which the user corrects the predicted rating when explanation facilities are offered. Also, we will explore our approach using other RSs as those using matrix factorization techniques, where the vector of factors could play a similar role to the rating histogram.

Acknowledgments. This paper has been supported by the Spanish "Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía", the "Ministerio de Ciencia e Innovación" and the research programme "Consolider Ingenio 2010" under the projects P09-TIC-4526, TIN2011-28538-C02-02 and MIPRCV:CSD2007-00018, respectively.

References

1. Bilgic, M., Mooney, R.J.: Explaining recommendations: Satisfaction vs. promotion. In: Proc. of the Wokshop Beyond Personalization, in Conjunction with the International Conference on Intelligent User Interfaces, pp. 13–18 (2005)
2. Chen, L., Pu, P.: Trust building in recommender agents. In: WPRSIU 2002 (2002)
3. Cleger-Tamayo, S., Fernandez-Luna, J., Huete, J.F.: Explaining neighborhood-based recommendations. In: SIGIR 2012, pp. 1063–1064 (2012)
4. Cleger-Tamayo, S., Fernández-Luna, J.M., Huete, J.F.: A New Criteria for Selecting Neighborhood in Memory-Based Recommender Systems. In: Lozano, J.A., Gámez, J.A., Moreno, J.A. (eds.) CAEPIA 2011. LNCS, vol. 7023, pp. 423–432. Springer, Heidelberg (2011)
5. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J.: Is seeing believing?: how recommender system interfaces affect users' opinions. In: CHI. Recommender systems and social computing, vol. 1, pp. 585–592 (2003)
6. Cramer, H., Evers, V., Ramlal, S., van Someren, M., Rutledge, Y., Stash, N., Aroyo, L., Wielinga, B.J.: The effects of transparency on trust in and acceptance of a content-based art recommender. User Model User-Adapt. Interaact 18(5), 455–496 (2008)
7. Gunawardana, A., Shani, G.: A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. J. of Machine Learning Research 10, 2935–2962 (2009)

8. Herlocker, J.L., Konstan, J.A., Riedl, J.: Explaining collaborative filtering recommendations. In: Proc. of the ACM Conference on Computer Supported Cooperative Work, CSCW 2000, pp. 241–250. ACM, New York (2000)
9. Herlocker, J.L., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 5(4), 287–310 (2002)
10. Herlocker, J.L., Konstan, J.A., Terveen, L., Riedl, J.T.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
11. Jarvelin, K., Kekalainen, J.: Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems 20(4), 422–446 (2002)
12. Konstan, J.A., Riedl, J.: 'Recommender systems: from algorithms to user experience'. User Model. User-Adapt. Interact. 22(1-2), 101–123 (2012)
13. Knijnenburg, B.P., Willemsen, M.C., Gantner, Z., Soncu, H., Newell, C.: Explaining the user experience of recommender systems. User Modeling and User-Adapted Interaction (UMUAI) 22, 441–504 (2012)
14. Lathia, N., Hailes, S., Capra, L., Amatriain, X.: Temporal diversity in recommender systems. In: SIGIR (2010)
15. Massa, P., Avesani, P.: Trust-aware recommender systems. In: RecSys, pp. 17–24 (2007)
16. McNee, S.M., Riedl, J., Konstan, J.A.: Being accurate is not enough: How accuracy metrics have hurt recommender systems. In: Extended Abstracts of ACM Conf. on Human Factors in Computing Systems (CHI 2006), pp. 1097–1101 (2006)
17. McNee, S.M., Lam, C., Guetzlaff, S.K., Konstan, J.A., Riedl, J.: Confidence displays and training in recommender systems. In: INTERACT IFIP TC13 International Conference on Human-Computer Interaction, pp. 176–183 (2003)
18. O'Sullivan, D., Smyth, B., Wilson, D.C., McDonald, K., Smeaton, A.: Improving the quality of the personalized electronic program guide. User Modeling and User-Adapted Interaction 14, 5–36 (2004)
19. Pu, P., Chen, L.: Trust-inspiring explanation interfaces for recommender systems. Knowledge-based Systems 20, 542–556 (2007)
20. Pu, P., Chen, L., Hu, R.: A user-centric evaluation framework for recommender systems. In: Recsys (2011)
21. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo (1993)
22. Ricci, F., Rokach, F., Shapira, B., Kantor, P. (eds.): Recommender System Handbook. Springer (2011)
23. Tintarev, N.: Explanations of recommendations. In: RecSys, pp. 203–206 (2007)
24. Tintarev, N., Masthoff, J.: Evaluating the effectiveness of explanations for recommender systems. In: User Modeling and User-Adapted Interaction (2012)