Confidence on Collaborative Filtering and Trust-Based Recommendations

Simon Meyffret[1,2], Lionel Médini[1,3], and Frédérique Laforest[1,4]

[1] Université de Lyon, CNRS
[2] INSA-Lyon, LIRIS, UMR5205, F-69621, France
simon.meyffret@liris.cnrs.fr
[3] Université Lyon 1, LIRIS, UMR5205, F-69622, France
lionel.medini@liris.cnrs.fr
[4] Telecom Saint-Etienne, LT2C, F-42000, France
frederique.laforest@telecom-st-etienne.fr

Abstract. Memory-based collaborative filtering systems predict items ratings for a particular user based on an aggregation of the ratings previously given by other users. Most systems focus on prediction accuracy, through MAE or RMSE metrics. However end users have seldom feedback on this accuracy. In this paper, we propose confidence on predictions in order to depict the belief from the system on the pertinence of those predictions. This confidence can be returned to the end user in order to ease his/her final choice or used by the system in order to make new predictions. It takes into account some characteristics on the aggregated ratings, such as number, homogeneity and freshness of ratings as well as users weight. We present an evaluation of such a confidence by applying it on different collaborative filtering systems of the literature using two datasets with different characteristics.

Keywords: recommendation, confidence, evaluation, dataset.

1 Introduction

Recommender systems are one solution classically proposed to help users select items among a lot of possibilities [13]. In this paper, we focus on memory-based collaborative filtering recommender systems that rely on relations between users to predict items that best fit their interests.

More and more e-commerce and collaborative websites include a recommendation system that proposes items or actions adapted to the user. Collaborative filtering is notably used on the Amazon website. The evaluation of such systems in the literature is mainly based on accuracy and coverage. These criteria are valuable for the comparison of systems and for the selection of the most efficient one. But when the system is deployed, end-users require other indications on the value of recommendations. Recommendations explanations can be provided using traces of the computation, but they are qualitative and difficult to interpret by naive end-users.

C. Huemer and P. Lops (Eds.): EC-Web 2013, LNBIP 152, pp. 162–173, 2013.

A quantitative confidence value, provided by the system as an indicator of the reliability of the recommendations, is easier to interpret. A study of the literature has shown that few systems propose a notion of confidence associated to their predictions. The few systems we have found just compute very simple confidence, for example with a standard deviation of the gathered ratings. In order to enrich the confidence notion, and to make it more valuable to the end-user, we propose a confidence formula dedicated to collaborative filtering recommender systems that takes into account five different confidence axes. Confidence should be provided with each prediction proposed to the end-user.

We also provide an evaluation of the proposed confidence so as to verify whether it is correlated with predictions accuracy. This evaluation is done using two different datasets extracted from two real websites with different characteristics. These datasets include data required for this evaluation as well as additional information gathered for wider purpose.

This paper is structured as follows. After a rapid tour of the literature, we define the five axes of confidence, as well as a synthetic confidence formula. We then describe our datasets and our evaluation protocol that measures the correlation between the confidence and accuracy of recommender systems predictions. Finally, we show the results of this evaluation on five different systems of the literature before concluding.

2 Related Work

Collaborative filtering systems predict item ratings for a particular user based on the items previously rated by other users [1]. To do so, they usually aggregate other users' ratings with the following function:

$$r_{a,i} = \frac{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'} \times r_{a',i}}{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'}} \tag{1}$$

where $r_{a,i}$ is the rating given by user a to item i, $\mathcal{A}_i$ is the set of users having rated item i (*aka.* "advisors") and $\omega_{a,a'}$ is a weight between a and a', typically a similarity coefficient. In this paper, we call **UserBasedCF** (respectively **ItemBasedCF**) the collaborative filtering algorithm defined in eq.1 where the ω coefficient is calculated using the Pearson's correlation coefficient between two users' ratings (resp. two items' ratings) [2].

Trust-based recommender systems build a subclass of collaborative filtering based on different links between users: users state that they trust the ratings expressed by other users [11,8,6]. For such systems, equation 1 is modified so that ω represents trust instead of similarity. Trust is implemented as a value in $[0, 1]$ that weights the links between users. Trust is the belief of one user in the usefulness of information provided by another user [5].

In the literature, very few collaborative filtering and trust-based systems use the notion of confidence. We present here the two best known trust-based systems as well as our previous work. We briefly explain the prediction principle and the associated confidence if any.

MoleTrust [8] predicts the trust value of a source user to a target user by gradually propagating trust in the user graph, up to a given depth k. In order to stop the propagation at some point, it defines a trust horizon, *i. e.* a maximum depth propagation k, being the maximum distance between u and v [9]. Beyond that distance, trust is not computed. MoleTrust does not provide confidence.

TrustWalker [3] is a random walk model combining trust-based and item-based recommendation. Each random walk returns a trusted user's rating on the item or on similar items, to depth k. Random walks are aggregated to produce the final prediction. TrustWalker associates a confidence value with each prediction, using the standard deviation of all T walks r_i (section 3.1 "variance confidence"):

$$confidence = 1 - \frac{\sigma^2}{\sigma^2_{max}} \quad \text{with} \quad \sigma^2 = \frac{\sum_{i=1}^{T}(r_i - \bar{r})^2}{T} \tag{2}$$

In **CoTCoDepth** [10], we use a trust or social network to propagate and aggregate ratings in a P2P manner up to a certain depth k. In [10], we have introduced a first version of our confidence coefficient, which takes into account previous confidence (recursively) and variance of rating predictions. This confidence is aggregated and transmitted at each hop.

As stated in the following, confidence is a composed notion that requires more attention. The next section presents a complete confidence formula.

3 Confidence

As shown in equation 1, collaborative filtering recommender systems usually aggregate ratings from trusted of similar users, *aka.* "advisors". This aggregation, or prediction, is returned as is to the final user, without justifying its accuracy. We think that all predictions should not be treated equally by the end user. For example, users cannot rely on a prediction computed from only one recommender as much as on a prediction computed from many advisors giving similar ratings.

In this section, we define a quantitative confidence coefficient associated with each prediction, in order to indicate to the final users which predictions are likely to be accurate. The higher the confidence, the higher the probability of the recommendation to be accurate, according to the system. Confidence is transmitted to the end user in order to justify the recommendation.

Definition 1 (Confidence). *The confidence $c_{a,i} \in [0, 1]$ of the system on the prediction provided to user a on item i depicts the belief from the system on the accuracy of this prediction.* 0 *means that the prediction is not likely to be accurate,* 1 *means that the system is confident on the accuracy. This coefficient is associated with each prediction.*

We extend this definition in order to attach confidence to any rating, not only prediction. That means that the system also deals with ratings differently during the final confidence computation. To better define confidence, we consider the following conditions to provide accurate predictions, therefore high confidence:

Size: many ratings are aggregated to provide the prediction,
Variance: aggregated ratings are homogeneous,
Advisors' confidence: ratings are associated with high confidence values,
Advisors' weight: ratings come from a well trusted advisors,
Freshness: ratings are recent.

In the following subsections, we provide a mathematical definition of confidence coefficients that take into account these conditions and aggregate those coefficients into a complete confidence formula.

3.1 Confidence Coefficients

Size Confidence (c^{size}) takes into account the number of advisors. The more advisors, the higher the confidence on the prediction.

We have chosen a logistic function (*c.f.* eq.3) to model that confidence: it is a monotonic increasing function. The initial growth (for positive values) is approximately exponential, followed by a slowing down until reaching value 1.

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3}$$

We define the following properties to adapt the logistic function to our case:

- 0.5 is the lowest size confidence, *i. e.* the confidence with only one advisor (flip-coin prediction),
- x is set so that a sufficient number of advisors leads to a high confidence[1].

The size confidence of user a on i's rating is defined in eq.4. It goes from 0.5 with only one advisor to about 1 with 7 advisors or more.

$$c^{size}_{a,i} = sigmoid(|\mathcal{A}_i| - 1) \tag{4}$$

Variance Confidence (c^{σ}) takes into account the variance of advisors' ratings. The higher the variance, *i. e.* the more different the recommendations, the lower the confidence on the prediction. This coefficient is similar to the one defined in equation 2. However our approach refines it by using a weighted variance, taking into account users' weights:

$$c^{\sigma}_{a,i} = 1 - \frac{\sigma^2_{a,i}}{\sigma^2_{max}} \tag{5}$$

$$\sigma^2_{a,i} = \frac{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'} \times (r_{a',i} - \mu^*)^2}{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'}} \tag{6}$$

μ^* is the advisors' ratings weighted mean. σ^2_{max} is the maximum possible variance and is used to normalize the confidence. As stated by [3], $\sigma^2_{max} = \frac{Range^2}{4}$ for a dataset with a finite rating range denoted $Range^2$.

[1] Our experimentations show that five advisors are enough to provide good accuracy, therefore high confidence.

[2] In our datasets, ratings are in $[1, 5]$, so $Range = 4$ and $\sigma^2_{max} = 4$.

Advisors' Confidence ($c^{\mathcal{A}}$) is implemented as the mean of advisors' confidence on their ratings, weighted by their ω coefficients:

$$c^{\mathcal{A}}_{a,i} = \frac{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'} \times c_{a',i}}{\sum_{a' \in \mathcal{A}_i} \omega_{a,a'}}$$

The lower advisors are confident on their ratings, the lower $c^{\mathcal{A}}$. In the meantime, advisors with high coefficients are more likely to influence this confidence. However, if all advisors have a high confidence on their ratings but all ω between them and the user are low, advisors' confidence will still be high.

Weight Confidence (c^{ω}) copes with the advisors' confidence drawback described previously. We consider that if all advisors that provide a rating have a low ω coefficient, the confidence should remain low. Therefore we define the weight confidence as the maximum of advisors' weights ω:

$$c^{\omega}_{a,i} = \max_{a' \in \mathcal{A}_i} \omega_{a,a'} \tag{7}$$

If at least one weight is high, the associated confidence will impact more the advisors' confidence coefficient, which handles cases with mixed high and low weights. Otherwise it will remain low. This coefficient takes into account cases where a prediction comes from many advisors highly confident on their recommendations, but where the links between them and the user have low weights.

Freshness Confidence (c^t). This confidence aims at taking into account rating obsolescence. It is specific to timestamped explicit ratings and does not consider predicted ones, as explained in section 3.2.

Freshness is function of the age of the rating: the older the less confident on a rating. We bound freshness to $]0.5, 1]$ with the following assumptions:

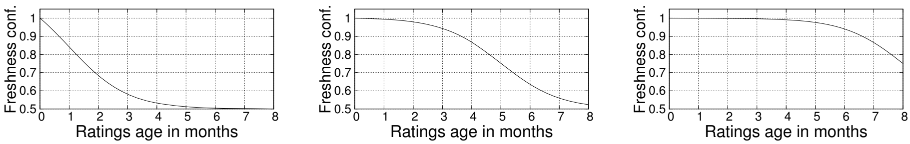
1. 1 is the highest confidence: when the rating has just been made,
2. it remains greater than 0.5: an old explicit rating is still an explicit rating made by the user.

These assumptions are generic but the freshness should be specific to items since some items ratings become obsolete faster than others. Therefore we define two parameters allowing us to tune the freshness according to the kind of recommended items:

– the half-life λ is the period of time after which the confidence lost about half its amplitude, *i. e.* equals 0.75 or so,
– the time unit $\mathcal{T}$, or scale, give the lifetime of a recommendation: minutes, days, months, etc.

In order to model the freshness function, we have also chosen a logistic function based on the *sigmoid* function defined in eq.3 page 165 (t is in $\mathcal{T}$ unit). The freshness is function of the age of the rating and monotonically decreasing. To satisfy the conditions 1 and 2, we define c^t as:

$$c^t_{a,i} = \frac{sigmoid(\lambda - t_{a,i})}{2 \times sigmoid(\lambda)} + 0.5 \tag{8}$$



(a) $\lambda = 1$ (b) $\lambda = 5$ (c) $\lambda = 8$

Fig. 1. Freshness confidence depending on ratings age with different values for λ

Figure 1 shows some examples with different λ and a unit time $\mathcal{T}$ in months.

Since λ and $\mathcal{T}$ are dependant on the kind of recommended items, they should be defined either by the users or the items category. We can assume that a tweet will have a low λ coefficient and a time unit $\mathcal{T}$ in hours or days, whereas a movie will have higher λ and $\mathcal{T}$.

3.2 Confidence Aggregation

Before returning confidence to the end user, the system aggregates the confidence coefficients. We consider two different cases: either it is an explicit rating made by the user, or it is a computed prediction. Depending on the situation, the confidence is not computed the same way.

Explicit Rating Confidence. If a user has rated an item, only the freshness confidence has a meaning, if the rating is timestamped. In that case, users' confidence on their own rating is the freshness confidence: $c_{a,i} = c^t_{a,i}$. Otherwise the confidence is 1, as we assume users to be confident on their own ratings.

Computed Prediction Confidence. When a user has not rated an item, the confidence on the computed prediction aggregates the other coefficients: size, advisors, weight and variance confidences.

If all coefficients are maximum (respectively minimum), then the aggregated confidence should be maximum (respectively minimum). But those coefficients are not independent from one another. The more advisors return ratings, the more advisors, weight and variance confidences are reliable.

The size confidence should influence the aggregation specifically: a high size confidence implies that the other coefficients are reliable, so we should use them; a low size confidence implies that the overall confidence should be low, since the other coefficients are not reliable enough.

Therefore we define the aggregated confidence c as follows:

$$c_{a,i} = c^{size}_{a,i} * \frac{c^{\mathcal{A}}_{a,i} + c^{\omega}_{a,i} + c^{\sigma}_{a,i}}{3} \tag{9}$$

With a high size confidence (near 1), the overall confidence is mainly computed using the advisors, weight and variance confidence. With a low size confidence (near 0.5), the overall confidence is low, no matter the other coefficients. Then the size confidence is always the maximum of the overall confidence.

Confidence Formula. The complete formula to compute confidence is then:

$$c_{a,i} = \begin{cases} c^{t}_{a,i} & \text{if } \exists r_{a,i} \\ c^{size}_{a,i} * \dfrac{c^{\mathcal{A}}_{a,i} + c^{\omega}_{a,i} + c^{\sigma}_{a,i}}{3} & \text{if } \nexists r_{a,i} \wedge \mathcal{A}_i \neq \emptyset \\ \bot & \text{otherwise} \end{cases} \tag{10}$$

4 Evaluation

In the previous section, we have defined confidence coefficients compatible with collaborative filtering recommender systems. This section evaluates the relevance of those coefficients in existing systems: UserBasedCF, ItemBasedCF, MoleTrust and CoTCoDepth, *c.f.* section 2. We also compare them with the confidence defined in TrustWalker [4].

Section 4.1 describes two datasets that we have built for the evaluation and comparison of collaborative filtering and trust-based systems. Section 4.2 depicts our evaluation metrics using those datasets. Section 4.3 provides a comparison between our coefficients on existing systems and the TrustWalker's one. It shows that confidence is correlated with accurate predictions.

4.1 Datasets

In this section, we introduce two datasets we have extracted from two different websites: Epinions and Appolicious.

Rich Epinions Dataset (RED). The Epinions[3] website contains reviews made by users on items, where users build their web of trust within the community. A web of trust is a list of trusted or distrusted users.

The dataset contains 131 228 users, 317 755 items and 1 127 673 reviews, that is a 0.003 % density. 113 629 users have at least one rating. 47 522 users have at least one trust relation. 31 000 users have at least one similarity computed toward another user. 21 910 users have at least one review, one trust relation and one computed similarity. 4 287 users have neither reviews nor trust relation.

In average, a user has less than one trusted user with a computable similarity: intersection between trusted users and similar users is very small. The output and input trust are equally distributed and follow a power law. This is common to social network datasets.

[3] `http://www.epinions.com`

The ratings count distribution follows a power law, a few users made a lot of ratings whereas most users made few ratings. Similarly, few items have been reviewed many times whereas most items were reviewed a few times. The ratings distribution is as follows: 7.2 % of 1, 7.4 % of 2, 12 % of 3, 30 % of 4 and 43.4 % of 5. We can see the particular distribution of the dataset. It is similar to the Trustlet [7] and Alchemy [12] datasets, also extracted from Epinions, and corresponds to the real distribution of the Epinions website.

Appolicious Dataset (AD). The Appolicious[4] website contains reviews made by users on mobile applications. Users follow other users of the community. Here, "Follow" means the same thing as "trust" in the Epinions website.

The dataset contains 4 058 users, 8 935 items (applications), 28 963 ratings and 12 546 reviews, with 10 605 common ratings/reviews, that is a 0.08 % density. 1 007 users have at least one rating. All users follow at least one other user.

There are 20 815 following links, that is 5 following/follower per user in average. The output and input following links are equally distributed and follow a power law. This is common to social network datasets.

The ratings distribution is as follows: 2.5 % of 1, 5.1 % of 2, 20 % of 3, 37 % of 4 and 35.4 % of 5. The ratings count distribution follows a power law, a few users made a lot of ratings whereas most users made few ratings. Similarly, few items have been reviewed many times whereas most items were reviewed a few times.

4.2 Metrics

Confidence as defined in this paper has no impact on predictions, therefore evaluating using RMSE or coverage makes no sense. Traditional recommender systems evaluations usually try to detect which recommender systems provide with the best accuracy or coverage. In order to highlight the impact of our confidence on predictions, we measure ρ as the correlation between confidence and accuracy. The greater ρ, the more confidence is correlated with accuracy, the more relevant the confidence, *i. e.* high confidences are associated with accurate predictions.

We compute ρ as the opposite of the Pearson correlation coefficient between confidence and error:

$$\rho = -\frac{\sum_{n=1}^{N} (c_n - \bar{c})(e_n - \bar{e})}{\sqrt{\sum_{n=1}^{N} (c_n - \bar{c})^2} \times \sqrt{\sum_{n=1}^{N} (e_n - \bar{e})^2}} \tag{11}$$

Let N be the total number of predictions. e_n is the error of prediction p_n on rating r_n: $e_n = |r_n - p_n|$. c_n is the confidence of the n^{th} rating prediction.

4.3 Results and Discussion

In order to evaluate our confidence coefficient on existing systems, we have implemented UserBasedCF, ItemBasedCF, MoleTrust2, and CoTCoD2 using a

[4] `http://www.appolicious.com`

propagation at depth 2 for the latter and run them on our datasets[5]. We have implemented TrustWalker2 to compare its confidence, noted "only variance", with ours. We have also implemented our confidence without the size coefficient, in order to evaluate the impact of the number of advisors on the confidence.

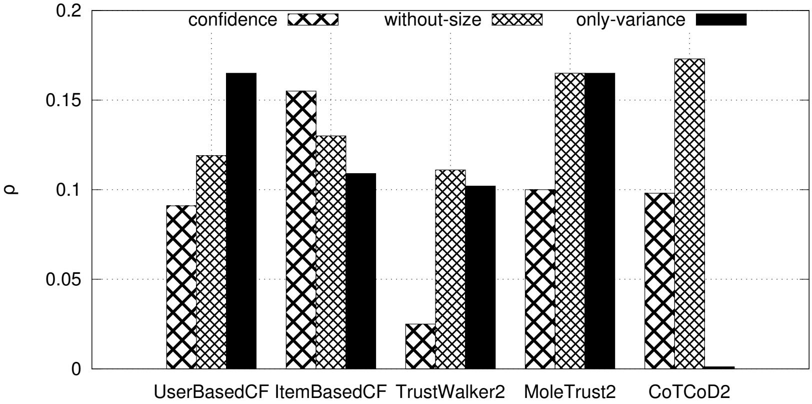


Fig. 2. Confidence correlation with prediction accuracy on RED

Figure 2 indicates ρ, the Pearson correlation coefficient between confidence and accuracy on RED. First of all, the *only variance* version is not correlated at all with the accuracy on CoTCoD2. CoTCoDepth is a trust-based recommender system propagating and aggregating ratings in a social network. Since ratings are aggregated, the number of advisors is usually low and produces a quite small variance, this latter being therefore not relevant. However this confidence is quite good with UserBasedCF. This approach aggregates ratings from similar users, *i. e.* users with homogeneous ratings. Moreover RED contains a lot of users, enhancing the chances to compute similarity.

We have evaluated our confidence with and without size confidence. We expected some improvements when taking into account the size but it seems that this is not always the case. Using RED, the size coefficient improves confidence correlation only with ItemBasedCF. Since RED is sparse, similarity between items is seldom computable, less than with users. Predictions using only few items are, as expected, less likely to be accurate.

Moreover, trust-based approaches provide the highest correlation between confidence and accuracy, especially with CoTCoD2. Sparse networks make similarity difficult to compute, prevailed by trust.

Figure 3 indicates the Pearson correlation coefficient between confidence and accuracy on AD. Clearly, size confidence is not compatible with TrustWalker. The latter aggregates ratings until the variance is low enough. With a dense

[5] Using a 99 % "training set" campaign on RED with a 4-cross validation and a "leave one out" campaign on Appolicious dataset.

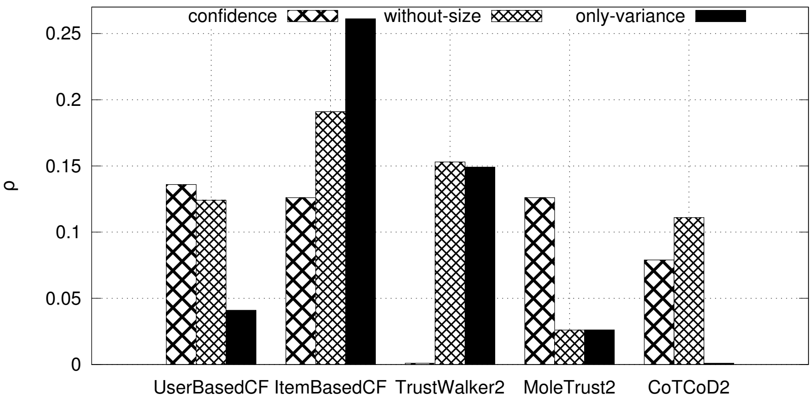


Fig. 3. Confidence correlation with prediction accuracy on AD

dataset such as AD, it does not need a lot of walks, therefore making predictions with few ratings. This implies a size confidence always low and adding noise in the confidence computation.

Using this dataset, our confidence is more relevant with UserBasedCF, but not with ItemBasedCF. AD contains much more items than users, letting ItemBasedCF use more items to compute predictions. Our coefficient produces higher correlation with UserBasedCF and MoleTrust with size confidence and TrustWalker / CoTCoDepth without.

This evaluation shows that confidence coefficients should be selected regarding the system and the dataset. Systems aggregating few ratings, such as CoTCoDepth, are not compatible with a simple variance, they require more sophisticated coefficients such as the ones we propose in this paper. On the other hand, TrustWalker performs better without size confidence, since their random walks aggregation is very heterogeneous, varying from few ratings up to 10 000 ones.

Regarding dataset density, size confidence is more relevant with dense datasets (AD) than with sparse ones (RED). It is more effective to distinguish predictions using lots of ratings.

5 Conclusion

In this paper, we introduce a confidence coefficient that aims to foresee predictions accuracy regarding some characteristics of the predictions, such as number, homogeneity and freshness of ratings as well as weights between users. Unlike traditional works on recommendation, we are not focusing on enhancing accuracy but on anticipating it. Our confidence is compatible with main classical collaborative filtering systems (UserBasedCF and ItemBasedCF) as well as trust-based systems. By definition, it is compatible with any approach aggregating ratings using weights or not.

End users may take into account this confidence as a second indicator, besides ratings. Existing systems already provide the number of ratings for each item, letting the user decide if an item with one excellent rating is more relevant than an item with many fairly good ratings. Confidence allows users to consider ratings number as well as other dimensions when selecting items.

The evaluation shows that our confidence is correlated with accuracy. Even if this correlation could be improved by further researches, it is most of the time higher than the state of the art's one. We show that some coefficients are more adapted than others to some system and/or dataset characteristics.

Confidence is composed of several coefficients defined in section 3, some of which are specific to ratings (freshness) and can be used during the aggregation in order to promote ratings that are likely to be accurate. In [10], we use this confidence during ratings propagation. We can extend existing recommender systems to propose a new function aggregating ratings, similarly to eq.1, considering weights between users and confidence on ratings:

$$r_{a,i} = \frac{\sum_{a' \in \mathcal{A}_i} f(\omega_{a,a'}, c_{a',i}) \times r_{a',i}}{\sum_{a' \in \mathcal{A}_i} f(\omega_{a,a'}, c_{a',i})} \tag{12}$$

With f a function of ω and c. In [10], we have used $\omega \times c$. This function should promote ratings that are likely to be accurate for the final prediction.

Finally we can imagine a meta-recommender system selecting the right prediction from several recommender systems using their confidence on predictions. The prediction with the higher confidence being returned to the end-user.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Madison, Wisconsin, pp. 43–52. Morgan Kaufmann Publishers Inc. (1998)
3. Jamali, M., Ester, M.: TrustWalker: a random walk model for combining trust-based and item-based recommendation. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 397–406. ACM (2009)
4. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 135–142. ACM (2010)
5. Lee, D.H., Brusilovsky, P.: Does Trust Influence Information Similarity? In: Proceedings of Workshop on Recommender Systems & the Social Web, the 3rd ACM International Conference on Recommender Systems, pp. 3–6. Citeseer (2009)
6. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–210. ACM, New York (2009)

7. Massa, P., Avesani, P.: Trust-aware bootstrapping of recommender systems. In: Proceedings of the ECAI 2006 Workshop on Recommender Systems, pp. 29–33 (2006)
8. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 17–24. ACM, New York (2007)
9. Massa, P., Avesani, P.: Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers. International Journal on Semantic Web and Information Systems 3(1), 39–64 (2007)
10. Meyffret, S., Médini, L., Laforest, F.: Trust-based local and social recommendation. In: Proceedings of the 4th ACM RecSys Workshop on Recommender Systems and the Social Web - RSWeb 2012, Dublin, Ireland, pp. 53. ACM Press (2012)
11. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: Proceedings of the 10th International Conference on Intelligent User Interfaces, pp. 167–174. ACM, New York (2005)
12. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 61–70. ACM, New York (2002)
13. Schafer, J.B., Konstan, J., Riedi, J.: Recommender systems in e-commerce. In: Proceedings of the 1st ACM Conference on Electronic Commerce - EC 1999, pp. 158–166. ACM Press, New York (1999)