Neighbor Selection and Weighting in User-Based Collaborative Filtering: A Performance Prediction Approach

ALEJANDRO BELLOGÍN, Universidad Autónoma de Madrid
PABLO CASTELLS, Universidad Autónoma de Madrid
IVÁN CANTADOR, Universidad Autónoma de Madrid

User-based collaborative filtering systems suggest interesting items to a user relying on similar-minded people called neighbors. The selection and weighting of these neighbors characterize the different recommendation approaches. While standard strategies perform a neighbor selection based on user similarities, trust-aware recommendation algorithms rely on other aspects indicative of user trust and reliability. In this paper we restate the trust-aware recommendation problem, generalizing it in terms of performance prediction techniques, whose goal is to predict the performance of an information retrieval system in response to a particular query. We investigate how to adopt the above generalization to define a unified framework where we conduct an objective analysis of the effectiveness (predictive power) of neighbor scoring functions. The proposed framework enables discriminating whether recommendation performance improvements are caused by the used neighbor scoring functions or by the ways these functions are used in the recommendation computation. We evaluated our approach with several state-of-the-art and novel neighbor scoring functions on three publicly available datasets. By empirically comparing four neighbor quality metrics and thirteen performance predictors, we found strong predictive power for some of the predictors with respect to certain metrics. This result was then validated by checking the final performance of recommendation strategies where predictors are used for selecting and/or weighting user neighbors. As a result, we have found that, by measuring the predictive power of neighbor performance predictors, we are able to anticipate which predictors are going to perform better in neighbor scoring powered versions of a user-based collaborative filtering algorithm.

Categories and Subject Descriptors: **H.3.3 [Information Search and Retrieval]**: information filtering

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Recommender Systems, User-Based Collaborative Filtering, Neighbor Selection, Neighbor Weighting, Trust, Performance Prediction

ACM Reference Format:

…

DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

Collaborative Filtering (CF) is a particularly powerful form of personalized recommendation that suggests interesting items to users based – in some way or other – on the preferences of similar-minded people [Herlocker et al. 2002; O'Donovan and Smyth 2005]. In CF, the simplest form of input data – evidence of user preference – consists of ratings, which are explicit relevance values given by users to items of interest. CF algorithms exploit the active user's ratings to make predictions, and thus have the interesting property that no item descriptions are

This research was supported by the Spanish Ministry of Science and Research (TIN2011-28538-C02-01). Part of this work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme, funded by European Comission FP7 grant agreement no.246016.

Author's addresses: A. Bellogín, P. Castells and I. Cantador, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Spain.

DOI10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

needed to provide recommendations, since they merely exploit information about past ratings between users and items. Compared to content-based information filtering approaches, CF has also the salient advantage that a user may benefit from other people's experience, thereby being exposed to potentially novel recommendations beyond her own experience [Adomavicius and Tuzhilin 2005].

Collaborative filtering approaches are commonly classified into two main categories: model-based approaches and memory-based approaches. Model-based approaches build statistical models of user/item rating patterns that provide automatic rating predictions. Memory-based approaches, in turn, can be user-based or item-based. In this paper we focus on the former, which explicitly seek people – commonly called neighbors – having tastes (and/or other characteristics) in common with the target user, and use preferences of the former to predict ratings for the latter. For additional information about collaborative filtering approaches in general, the reader is referred to [Su and Khoshgoftaar 2009; Ekstrand et al. 2011; Cacheda et al. 2011]. User-based algorithms are built on the principle that a particular user's rating records are not equally useful to all other users as input to provide them with item suggestions [Herlocker et al. 2002]. Central aspects to these algorithms are therefore a) how to identify which neighbors form the best basis to generate item recommendations for the active user, and b) how to properly make use of the information provided by them. Typically, neighborhood identification is based on selecting those users who are most similar to the active user according to a similarity metric [Desrosiers and Karypis 2011]. The similarity of two users is generally computed by a) finding a set of items that both users have interacted with, and b) examining to what degree the users displayed similar behaviors (e.g. rating, browsing and purchasing patterns) on these items. This basic approach can be complemented with alternative comparisons of virtually any user feature a system has access to, such as personal demographic and social network data. It is also a common practice to set a maximum number of neighbors (or a minimum similarity threshold) to restrict the neighborhood either for computational efficiency, or in order to avoid noisy neighbors who are not that similar. Once the active user's neighbors are selected, the more similar a neighbor is to the active user, the more her preferences are taken into account as input to produce recommendations. For instance, a common user-based approach consists of predicting the relevance of an item for the active user by a linear combination of her neighbors' ratings, which are weighted by the similarity between the target user and her neighbors.

User similarity has been the central criterion for neighbor selection in most of the user-based CF literature [Desrosiers and Karypis 2011]. Nonetheless, it has been recently suggested that additional factors could have a valuable role to play on this point. For instance, two users with a high global similarity value may no longer be reliable predictors for each other at some point, because of a divergence of tastes over time. Thus, in the context of user-based CF, more complex methods have been proposed in order to effectively select and weight useful neighbors [O'Donovan and Smyth 2005; Desrosiers and Karypis 2011]. In this way, a particularly relevant dimension considered in this context relates these additional factors with the general concept of trust (trustworthiness, reputation) on a user's contribution to the computation of recommendations. A number of trust-aware recommender systems have been proposed in the last decade [Hwang et al. 2007; O'Donovan and Smyth 2005; Golbeck 2009]. Most of these systems focus on the improvement of accuracy metrics, such as the Mean Average Error (MAE), by defining different heuristic trust functions, which, in most cases, are applied either as additional weighting factors in the neighbor-based CF formulation, or as a component of the neighbor selection criteria. The way trust is measured is considerably diverse in the literature. In fact, the notion of trust has embraced a wide scope of neighbor aspects, spanning from personal trust on the neighbor's faithfulness, to trust on her competence, confidence in the correctness of the input data, or the effectiveness of the recommendation resulting from the neighbor's data.

The research presented here seeks to provide an algorithmic generalization for a significant variety of notions, computational definitions, and roles of trust in neighbor selection. Specifically, we aim to provide a theoretical framework for neighbor selection and weighting, in which trust measures can be defined and evaluated in terms of improvements on the final recommender's performance. We cast the rating prediction task – typically based, as described above, on the aggregation of the neighbors' preferences – into a framework for dynamic combination of inputs, from a performance prediction perspective, borrowing from the methodology for this area in the Information Retrieval (IR) field. The application of this perspective is not trivial, and requires, in particular, a definition of what the performance of a neighbor means in this context. Hence, restating the problem in these terms, we propose to adapt and exploit techniques and methodologies developed in IR for predicting query performance; in our case, we equate the active user's neighbors to the queries, and our goal is to predict which of these neighbors will perform better for the active user (the retrieval system). Furthermore, by providing in our framework an objective measurement of the neighbor scoring function efficiency, we would be able to obtain a better understanding of the whole recommendation process.

The contributions of this paper can be summarized as follows:

— A framework that provides a formal setting for the evaluation of neighbor selection and weighting functions, while, at the same time, enables discriminating whether recommendation performance improvements are achieved by the neighbor scoring functions, or by the way these functions are used in the recommendation computation.

— A unification of state-of-the-art trust-based recommendation approaches, where trust measures are cast as neighbor performance predictors. As a result, we propose four neighbor quality metrics and thirteen performance predictors, defined upon a specific neighbor (user-based), a neighbor and the current user (user-user), and a neighbor and the current item (user-item).

— A generalization of the different strategies proposed in the literature to introduce trust into collaborative filtering. Moreover, thanks to the proposed formulation, new strategies have naturally appeared, and have also been evaluated. Empirical results show that the trust metrics, interpreted as neighbor scoring functions, that correlate with the notion of neighbor performance, produce better recommendations when they are introduced in a user-based collaborative filtering algorithm.

The remaining of the paper has the following structure. Section 2 describes the different strategies in which neighbor scoring functions can be incorporated into standard CF algorithms, along with a survey of the different scoring functions (such as trust measures) proposed in the literature. Section 3 presents a unified formulation and the proposed framework for neighbor selection and weighting in user-based recommendation, and Section 4 describes how the different neighbor scoring functions proposed in the literature fit into the framework. Finally, Section 5 presents experiments conducted with the framework, and Section 6 ends with some conclusions and potential future lines of work.

2. NEIGHBOR SELECTION AND WEIGHTING IN RECOMMENDER SYSTEMS

The first memory-based CF model can be attributed to Resnick and colleagues [Resnick et al. 1994], who modeled a target user u's preferences for items i as numeric ratings $r(u,i)$, whereby unseen items are recommended to u by predicting their ratings taking into account ratings observed by the users v who are most "similar" to u and have rated such items, as follows:

$$\tilde{r}(u,i) = \bar{r}(u) + C_0 \sum_{v \in N_k(u,i)} sim(u,v)\big(r(v,i) - \bar{r}(v)\big) \qquad (1)$$

where $\tilde{r}$ denotes a rating prediction (as opposed to observed ratings r), $N_k(u,i)$ is the set of k most similar users to u, usually called neighborhood, $sim(u,v)$ is a function that measures the similarity between two users, and the constant C_0 is a normalization factor. The preference of user u for item i is predicted based on the average rating $\bar{r}(u)$, and the sum of the deviations of neighbor v's ratings for i and average ratings $\bar{r}(v)$, weighted by the similarities with her neighbors. Other formulations consist of the weighted sum of the neighbors' ratings and similarities (ignoring the average ratings of the target user and neighbors), such as the one described in [Adomavicius and Tuzhilin 2005]. In the rest of the paper, we use the method proposed by Resnick as presented in Equation 1, although no significant modifications would be required if a different method is meant to be used. Thus, based on Resnick's scheme, the concept of neighbor scoring in CF has been developed in different recommendation approaches, which can be described as extensions and adaptations of the above user-based CF formula. Some of these extensions also involve an elaboration of the neighbor selection and weighting criteria beyond similarity. A well-known example is the addition of a confidence weight to the similarity measure, where the number of common items rated by two users is taken as an additional condition for selecting good neighbors [McLaughlin and Herlocker 2004; Ma et al. 2007]. In other works, the notion of trust is introduced to provide a measure of how neighbors should be weighted or when they should be selected. More specifically, in trust-aware recommender systems, a trust model is defined and, typically, introduced into the Resnick's equation either as an additional weight or as a filter for the potential user's neighbors. These models can be classified, depending on the nature of their input, into rating-based and social-based (using a trust network).

One of the first approaches that uses explicit ratings to define trust metrics between users was presented in [O'Donovan and Smyth 2005]. In that work, the authors propose to modify how the "recommendation partners" (neighbors) are weighted and selected in the user-based CF formula. They argue that the *trustworthiness* of a particular neighbor should be taken into account in the computed recommendation score by looking at how reliable her past recommendations were. Trust values are computed by measuring the number of correct recommendations in which a user has participated as a neighbor, and then they are used for weighting the influence (along with computing the similarity), and selecting the active user's neighbors. Weng et al. (2006) propose an asymmetric trust metric based on the expectation of other users' competence in providing recommendations to reduce the uncertainty in predicting new ratings. The metric is used in the standard CF formula instead of the similarity value. Two additional metrics are defined in [Kwon et al. 2009] based on the similarity between the ratings of a neighbor and the ratings from the community. Finally, Hwang et al. (2007) define two trust metrics (local and global) by averaging the prediction error of co-rated items between a user and a potential neighbor.

Social-based trust metrics make use of explicit trust networks of users, built upon friendship relations [Avesani et al. 2004; Massa and Bhattacharjee 2004] and explicit trust scores between individuals in a system [Ma et al. 2009; Walter et al. 2009]. These metrics and, to some extent, their inherent meanings, are different with respect to rating-based metrics; nonetheless, in [Ziegler and Lausen 2004], the authors conduct a thorough analysis that shows empirical correlations between trust and user similarity, suggesting that users tend to create social connections with people who have similar preferences. Once such a correlation is proved, techniques based on social-based trust can be applicable. In [Golbeck & Hendler 2006], the authors propose a metric called TidalTrust to infer trust relationships by using recursive search. Inferred trust values are used for every user who has rated a particular item in order to select only those users with high trust values. Then, a weighted average between ratings and trust provides the predicted ratings. In [Massa and Avesani 2007], the authors propose similar local (MoleTrust) and global (PageRank) trust metrics; they found that trust-based recommenders are very useful regarding cold start users.

3. A PERFORMANCE PREDICTION FRAMEWORK FOR NEIGHBOR SELECTION AND WEIGHTING

3.1 Unifying Neighbor Selection and Weighting in User-Based Collaborative Filtering

From the observation that most of the methods for neighbor selection and weighting are elaborated upon the standard Resnick's scheme, we propose a unified formulation as follows. Let us suppose, for the sake of generality, that we have a neighbor scoring function $s(u,v,i)$ that may depend on the active user u, a neighbor v, and a target item i. This function should output a higher value whenever the user, neighbor, item, or a combination of them, is more trustworthy (in the case of trust models) or expected to perform better as a neighbor according to the information available in the system, such as other ratings and external information, like a social network. Using this function, we generalize equation 1 to:

$$\tilde{r}(u,i) = \bar{r}(u) + C \sum_{v \in g(u,i;k;s)} f\big(s(u,v,i), sim(u,v)\big)\big(r(v,i) - \bar{r}(v)\big) \qquad (2)$$

where the function g denotes the selection of the set of neighbors, and f is an aggregation function combining the output of s and similarity into a weight value. In this way, we have the neighbor scoring function s integrated into the Resnick's formula in order to: a) select the neighbors to be considered in the formula, instead of or in addition to the most similar users (via function g), and b) provide a general weighting scheme (by introducing an aggregation function f) between the actual neighbor score and the similarity between the active user and her neighbors. Note that it is not required that s is bounded, since the constant C would normalize the output rating value. The function s is thus a core component in the generalization of the collaborative scheme. It might embody similarity in itself (in which case f might just return its first argument), but sim and f are left in the formulation to simplify the connection to the original similarity-only formulation, and to suit particular cases where s applies only other principles, separate from similarity.

The aggregation function f can take different definitions, some examples of which can be found in the literature. For instance, O'Donovan and Smyth (2005) initially propose to use the arithmetic mean of the neighbor score (x) and the similarity (y; henceforth denoted as f_1), and end up using the harmonic mean (f_2) because of its better robustness to large differences in the inputs. In [Bellogín et al. 2010], on the other hand, the product function (f_3) is used. Moreover, Hwang and colleagues [Hwang et al. 2007] propose to directly use the neighbor score as the weight given to neighbors, that is, they use the projection function $f_4(x,y) = x$. Obviously, the original Resnick's formulation can be expressed as the symmetric projection function $f_0(x,y) = y$.

The neighborhood selection embodied in function g also generalizes Resnick's approach – the latter corresponds to the particular case $g_0(u,i;k;s) = N_k(u,i)$, where the neighbor scoring function is ignored, and only similarity is used. The general form admits different instantiations. In [Golbeck and Hendler 2006] only the users with the highest trust values are selected as neighbors. In [O'Donovan and Smyth 2005], on the other hand, those users whose trust values exceed a certain threshold are taken into consideration. This threshold is empirically defined as the mean across all the obtained values for each pair of users. The latter strategy can be formulated in general as follows:

$$g_1(u,i;k;s) = \{v \in N_k(u,i) \colon s(u,v,i) > \tau\};\ \tau = \frac{1}{|\{(u,v,i)\}|} \sum_{(u,v,i)} s(u,v,i).$$

There are, nonetheless, some considerations to take into account when using specific combinations of neighbor weighting and neighbor selection functions. First, if f_4 is used together with g_0, since only the most similar users are considered in the neighborhood, then less reliable users (low f_4) who are very similar to the current user would be penalized, and more reliable neighbors but less similar to the current user are ignored, since they do not belong to the neighborhood. Second, when using f_0 together with g_1, neighbors are weighted by their similarities with the active user; however, these similarities could be very low, and thus, non-similar but reliable neighbors would be penalized. Finally, if f_4 is used with g_1, the similarity weight will not be considered at any point in the recommendation process. Nonetheless, some of these configurations may deserve further investigation, and will be considered in Section 5, along with other combinations not listed here.

3.2 Neighbor Selection and Weighting as a Performance Prediction Problem

Neighbor scoring and selection can be seen as an issue of predicting the effectiveness of neighbors as input for collaborative recommendations. This links to a considerable body of research on performance prediction in Information Retrieval, as we elaborate next. Performance prediction in IR has been mostly addressed in terms of query performance, which refers to the effectiveness of an IR system in response to a particular query. It also relates to the appropriateness of a query as an expression of the user's information needs. Dealing effectively with poorly-performing queries is a crucial issue in IR, and performance prediction helps systems cope with such problems in several ways [Cronen-Townsend et al. 2002; Yom-Tov et al. 2005; Zhou and Croft 2006]. From the user perspective, it provides valuable feedback that can be used to direct a search, e.g. by rephrasing the query or by providing relevance feedback. From the perspective of an IR system, performance prediction provides a means to address the problem of retrieval consistency: a system can invoke alternative information retrieval strategies for different queries according to their expected performance, such as query expansion and alternative ranking functions based on predicted complexity/difficulty of the query. From the perspective of a system administrator, performance prediction could help on identifying queries related to a specific subject that are difficult to capture by a search engine, or could help on expanding the collection of documents to better answer insufficiently covered subjects. Finally, for distributed IR, performance estimations can be used to decide which search engine and/or database to use for each particular query, or to decide how much weight give to different search engines when their results are combined.

The same as performance prediction in IR has been used to optimize rank aggregation [Yom-Tov et al. 2005], in our proposed framework each user's neighbor can be seen as a retrieval subsystem (or criterion) whose output is to be combined to form the final system output (the recommendations) to the user. In more general terms, for user-based CF algorithms, the estimation $\tilde{r}(u,i)$ of the preference of the active user u for a particular item i can be formulated as an aggregation function of the ratings of some other users $\hat{V}$:

$$\tilde{r}(u,i) \propto \mathrm{aggr}_{v\in\hat{V}}\big(sim(u,v);\ r(v,i);\bar{r}(u);\bar{r}(v)\big)$$

where $\hat{V}$ denotes the selected neighbors for a particular user u according to function g as defined in the previous section (see Equation 2). As observed in [Adomavicius and Tuzhilin 2005], different aggregation functions can be defined, but the most typical one is the weighted average function presented in the previous section.

In that function, the term $\tilde{r}(u,i)$ can be seen as a retrieval function that aggregates the outputs of several utility subfunctions $r(v,i) - \bar{r}(v)$, each corresponding to a recommendation obtained from a neighbor of the active user. The combination of utility values is defined as a linear combination (translated by $\bar{r}(u)$) of the neighbor's ratings, weighted by their similarity $sim(u,v)$ with the active user. The computation of utility values in user-based CF thus can be seen as a case of rank aggregation in IR, and as such, a case for the enhancement of aggregated results by predicting the performance of the recommendation outputs being combined. In fact, the similarity value can be seen as a prediction of how useful the neighbor's advice is expected to be for the active user, which has proved to be a quite effective approach. The question is whether other performance factors, beyond user similarity can be considered in a way that further enhancements can be drawn, as research on user trust awareness has attempted to prove in the last years.

The IR performance prediction view provides a methodological approach, which we propose to adapt to the neighbor selection problem. The approach provides a principled path to drive the formulation, development and evaluation of effective neighbor selection and weighting techniques, as we shall see. In the proposed view, the selection/weighting problem is expressed as an issue of neighbor performance, as an additional factor (besides user similarity) to automatically tune the neighbors' contribution to the recommendations, according to the expected goodness of their advice. There are three core concepts in the performance prediction problem as addressed in the IR literature: performance predictor, retrieval quality assessment, and predictor quality assessment. Since we are dealing with the prediction of which users may perform better as neighbors, these three concepts can be translated into: *neighbor performance predictor*, *neighbor quality*, and *neighbor predictor quality*. For the sake of simplicity, let us assume we can define a performance predictor as a function that receives as input a user profile u (in general, it could receive other users or items as well), the set of items I_u rated by that user, and the collection S of ratings and items (or any other user preference and item description information) available in the system. Then, following the notation given in (Carmel & Yom-Tov, 2010), we define a neighbor performance prediction function as:

$$\hat{\mu}(u) \leftarrow \gamma(u, I_u, S).$$

The function γ can be defined in many possible ways, for instance, by taking into account the rating distribution of each user, the number of ratings available in the system, and the (implicit or explicit) relations made by that user with the rest of the community. Essentially, the neighbor performance predictor is intended to estimate the true neighbor quality metric, denoted as $\mu(u)$, which is typically measured using groundtruth information about whether the neighbor's influence is positive. The application of this perspective is not trivial, and requires, in particular, a definition of what the performance of a neighbor means in this context, where no standard metric for neighbor performance is yet available in the literature.

Once the estimated neighbor performance prediction values $\hat{\mu}(u_n)$ are computed for all users, the quality of the prediction can be measured by the correlation between these estimations and the real values $\mu(u_n)$. In other words, the neighbor predictor quality metric is defined as the following correlation:

$$q(\gamma) = \mathrm{corr}([\hat{\mu}(u_1), \cdots, \hat{\mu}(u_n)], [\mu(u_1), \cdots, \mu(u_n)]).$$

This correlation provides an assessment of the prediction accuracy [Carmel and Yom-Tov 2010]; the higher its (absolute) value, the higher the predictive power of γ. Moreover, the sign of $q(\gamma)$ represents whether the two variables involved (neighbor prediction and neighbor quality) are directly or inversely correlated. Then, depending on the computed real correlation values, different dependencies between variables can be captured. Pearson's correlation, which captures linear dependencies between variables, is frequently used, and Spearman's and Kendall's correlation coefficients are used in order to uncover non-linear relationships between the variables.

Besides validating any proposed predictor by checking the correlation between predicted outcomes and objective measurements, we may also evaluate the effectiveness of the defined predictors by introducing and testing a dynamic variant of user-based CF, in which the weights of neighbors are dynamically adjusted based on their expected effectiveness, along with the decision of which users belong to each neighborhood, as in the general formulation presented in Equation 2, where the neighbor scoring function $s(u, v, i)$ is defined based on the values computed from each neighbor performance predictors.

Hence, the basic idea of the framework presented here is to formally treat the neighbor selection and weighting in memory-based recommendation as a performance prediction problem. In the next section we show different approaches to measure the true performance of a neighbor. We also present how we can integrate the different neighbor scoring functions defined in the literature into our framework; in particular, how the trust models and functions described in the previous section can be defined in terms of concepts related with performance prediction.

Additionally, an objective analysis of the trust model efficiency can be conducted by means of the proposed framework, which contrasts with the current development in memory-based recommendation, where the reason why different neighbor weighting strategies result in better or worse recommendation effectiveness is not clear. The performance prediction framework provides a principle basis to analyze whether the predictors are capturing some valuable, measurable characteristic known to be useful for prediction, independently from their latter use in a recommendation strategy. Furthermore, if a neighbor scoring function with strong predictive power is introduced into the recommendation process and the performance is not improved, then, new ways of introducing such predictor into the rating estimation should be tested (either for selection or weighting), since we have some confidence that this function captures interesting user's characteristics, valuable for recommendation.

4. NEIGHBOR QUALITY METRICS AND PERFORMANCE PREDICTORS

The performance prediction research methodology requires a means to compare the predicted performance with observed performance. This comparison is typically conducted in terms of some one-dimensional functional values, where the prediction can be translated to a certain numeric value (without any further interpretation than its relative magnitude, quantifying the expected degree of effectiveness), and performance is assessed by some specific metric. In the context of query performance in IR, metrics of system effectiveness in response to the query are used for this purpose, and the IR field is rich in well studied, widely understood metrics. In our case, predicting the performance of a neighbor for recommendation would thus require selecting some measure of how effective a neighbor is, for which there is no readily available metric in the literature. We thus address this need as part of our research. Along with that, we propose several performance predictors, which we shall later test empirically (in Section 5) against the proposed neighbor quality metrics.

4.1 Neighbor Quality Metrics

The purpose of effectiveness predictors in our framework is to assess how useful specific neighbors' profiles are as a basis for predicting ratings for the active user. Each performance predictor for a neighbor needs thus to be contrasted to a measure of how "good" is the neighbor's contribution to the global community of users in the system. In contrast with query performance prediction, where a well-established array of metrics can be used to quantify query performance, to the best of our knowledge, there is not an equivalent function for CF neighbors in the literature. We therefore need to introduce and propose some sound candidate metrics.

Ideally, in the proposed framework, a quality metric should take the same arguments as the predictor, and thus, if we have, for instance, a user-item predictor, we should also be able to define a quality metric that depends on users and items. In general, we shall focus here on user-based predictors, but it would be possible to explore item-based alternatives. Furthermore, we shall consider metrics taking neighbors as their single input, independently from which neighborhood that user is being involved in (i.e., independently from the target user) and what item is being recommended. At the end of this section, nevertheless, we shall introduce a neighbor quality metric suitable for the user-user scenario, where both the neighbor and the target user are taken into account.

Now, we propose three different neighbor quality metrics. The first two metrics had a different original intended use by their authors, but we find they could be useful to evaluate how good a user is as a neighbor. The third one was proposed by us in [Bellogín et al. 2010], where the problem of neighbor performance was explicitly addressed. In [Rafter et al. 2009], the authors propose two metrics in order to examine whether the neighbors have any influence in the recommendation. Both metrics are based on the comparison between true ratings and the estimation of a neighbor's ratings, as a way to measure the direction of the neighbor estimation and the average absolute magnitude of the shift produced by this estimation. Thus, the larger the neighbor's influence, the better her performance, according to our definition of "good" neighbor. In this context, we use these metrics as neighbor quality metrics as follows:

$$\mu_1^a = \mu(v) = \frac{1}{|T_v|} \sum_{i \in T_v} \frac{1}{|N_k^{-1}(v;i)|} \sum_{w \in N_k^{-1}(v;i)} |r(w,i) - r(v,i)|$$

$$\mu_1^b = \mu(v) = \frac{1}{|T_v|} \sum_{i \in T_v} \frac{1}{|N_k^{-1}(v;i)|} \sum_{w \in N_k^{-1}(v;i)} \left| sim(v,w)\big(r(w,i) - r(v,i)\big) \right|$$

$$\mu_2 = \mu(v) = \frac{1}{|T_u|} \sum_{i \in T_v} \frac{1}{|N_k^{-1}(u;i)|} \sum_{w \in N_k^{-1}(v;i)} \delta\big(sgn\big(r(w,i) - \bar{r}(v)\big) = sgn\big(r(v,i) - \bar{r}(v)\big); 1\big)$$

δ being a binary function whose output is 1 if its arguments are true, and 0 otherwise. Metrics μ_1^a and μ_1^b represent the absolute error deviation of a particular user; in the second case the deviation is weighted by the similarity between a user and her neighbor, as originally proposed in [Rafter et al. 2009]. We found a negligible difference between these two metrics, and thus, for the sake of simplicity, in the following we will use $\mu_1 = \mu_1^a$ as the **absolute error deviation** metric. The metric μ_2 is the **sign of error deviation**. Now, $N_k^{-1}(v;i)$ is an inverse neighborhood, which represents those users for whom v is a neighbor, and T_v denotes the items rated by user v in the test set. We can observe how each of these metrics represents a different method to measure how accurate the user v is as a neighbor.

In [Bellogín et al. 2010], a measure named **neighbor goodness** is proposed, which is defined as the difference in performance of the recommender system when including vs. excluding the user (their ratings) from the dataset, similar to the influence measure proposed in [Rashid et al. 2005]. For instance, based on the mean average error standard metric, neighbor goodness can be instantiated as:

$$\mu_3 = \mu(v) = \frac{1}{|R_{U-\{v\}}|} \sum_{w \in U \setminus \{v\}} \left[CE_{U \setminus \{v\}}(w) - CE_U(w) \right]; CE_X(v) = \sum_{i \in I, r(v,i) \neq \emptyset} |\tilde{r}_X(v,i) - r(v,i)|,$$

where $\tilde{r}_X(v,i)$ represents the predicted rating computed using only the data in X. This formula quantifies how much a user affects (contributes to or detracts from) the total amount of mean average error of the system, since it is computed in the same way as that metric, but leaving out the user of interest – in the first term, it is completely omitted; in the second term, the user is only involved as a neighbor. In this way, we measure how a user contributes to the rest of users, or put informally, how better or worse the "world" is, in the sense of how well recommendations work, with and without the user. Hence, if the error increases when the user is removed from the dataset, she is considered as a good neighbor.

Based on the same idea of the last metric, we propose a user-user quality metric, which measures how one particular user affects to the error of another user when acting as her neighbor:

$$\mu_4 = \mu(u,v) = CE_{U \setminus \{v\}}(u) - CE_U(u)$$

We call this metric **user-neighbor goodness**. It quantifies the difference in user u's error when neighbor v is not in the system against the error when such neighbor is present, that is, it measures how much each neighbor contributes to reduce the error of a particular user.

4.2 Neighbor Performance Predictors

Having formulated neighbor selection in memory-based recommendation as an issue of neighbor effectiveness prediction, and having proposed effectiveness measures to compare against, the core of an approach to this problem is the definition of effectiveness predictors, which we address next. Similarity functions and trust models such as those discussed in Section 2 can be directly used already for this purpose, since in trust-aware recommendation, trust metrics aim to measure how reliable a neighbor is when introduced in the recommendation process [O'Donovan and Smyth 2005]. Interestingly, some of them depend only on one user (*global trust metrics*), and others depend on a user and an item or another user (*local trust metrics*). Furthermore, other authors have proposed different indicators for selecting good neighbors, mainly based on the overlap between the user and her neighbor, without considering the concept of trust.

We thus distinguish three types of neighbor performance predictors: *user predictors* – equivalent to the global trust metrics –, *user-item predictors*, and *user-user predictors* – equivalent to the local trust metrics. Note that, although trust metrics could now be interpreted as neighbor performance predictors, the proposed performance prediction framework let us provide an inherent value to these metrics (identified as performance predictors), independently of whether they improve the algorithm's performance when used for selecting or weighting in the specific CF algorithm. This is because it is possible to empirically check the quality of the prediction by analyzing their correlation with respect to the neighbor performance metric, prior to the integration in any CF method. Thus, each predictor would obtain an explicit score that represents its predictive power, related with our confidence *a priori* on whether such predictor is capturing the neighbor's reliability or trustworthiness. In the next subsections, we propose an array of neighbor effectiveness prediction methods, by adapting and integrating trust functions from the literature into our framework, and we further extend this set by proposing new additional predictor functions.

4.2.1. User Predictors

User predictors depend only on one user –the current neighbor. When that neighbor is predicted to perform well, her assigned weight in the CF formulation is high. In this context, one of the first trust metrics proposed in the literature is the **profile-level trust** [O'Donovan and Smyth 2005], which is basically defined as the percentage of correct recommendations in which a user has participated as a neighbor. If we denote the set of recommendations in which a user has been involved in as

$$RecSet(u) = \{(v,i) \colon u \in N_k(v;i)\},$$

then the predictor is defined as follows:

$$\gamma_1(u,v,i) = \gamma(v) = \frac{|CorrectSet(v)|}{|RecSet(v)|},$$

where the definition of correct recommendations depends on a threshold ϵ:

$$CorrectSet(u) = \{(c_k, i_k) \in RecSet(u) \colon Correct(i_k, u, c_k; 1)\}$$
$$Correct(i,u,v;\lambda) = \delta(|r(u,i) - r(v,i)| \leq \epsilon; \lambda),$$

As before, $\delta(a;b)$ represents a binary function, but in this case it outputs a value b if the predicate a is true, and 0 otherwise. That is, the recommendations considered as correct are those in which the user was involved as a neighbor, and her ratings were close (up to a distance of ϵ) to the actual ratings.

A similar trust metric, called **expertise trust**, is presented in [Kwon et al. 2009], where the concept of 'correct recommendations' is also used. In that work, the authors introduce a compensation value for situations in which few raters are available. Specifically, the correct recommendation function only outputs a value of 1 when there are enough raters for a particular item (more than 10 in the paper). Otherwise, an attenuation factor is introduced by dividing the number of raters by 10, in the same way as significance weighting is introduced in Pearson's correlation in [Herlocker et al. 2002]. More formally, the predictor is defined as:

$$\gamma_2(u,v,i) = \gamma(v) = \frac{1}{\sum_{j \in I_v} \sum_{w \in U_i} 1} \sum_{j \in I_v} \sum_{w \in U_i} Correct\big(j, v, w; \lambda(j)\big)$$

where $\lambda(j)$ is 1 when item j has more than 10 raters, and the users who rated item i are denoted as U_i. In the same paper the authors propose another trust metric called **trustworthiness**, which is equivalent to the absolute value of the similarity between the active user's ratings and the average ratings given by the community (denoted as $\bar{R}$). The authors also introduce the significance weighting factor β as in [Herlocker et al. 2002], in a way that $\beta(v)$ is 1 when user v has more than 50 ratings; otherwise, β is computed as the user's ratings divided by 50. Once the β factor is computed, the predictor is defined as follows:

$$\gamma_3(u,v,i) = \gamma(v) = \beta(v) \times \left| \frac{\sum_{j \in I_v} \big(r(v,j) - \bar{r}(v)\big)\big(\bar{r}(j) - \bar{R}\big)}{\sqrt{\sum_{j \in I_v} \big(r(v,j) - \bar{r}(v)\big)^2 \sum_{j \in I_v} (\bar{r}(j) - \bar{R})^2}} \right|$$

In [Hwang et al. 2007], the authors define a global trust metric, which we call **global trust deviation**, defined as an average of local (user-to-user) trust deviations. This metric makes use of the predicted rating for a user–item pair, by using only one user as neighbor:

$$\tilde{r}(u,i) \sim \tilde{r}(u,i;v) = \bar{r}(u) + \big(r(v,i) - \bar{r}(v)\big)$$

user v being the considered neighbor. Then, the predictor is computed by averaging the prediction error of co-rated items for each user, and normalizing the error according to the rating range R_r (e.g., in a typical 1 to 5 rating scale, R_r = 4):

$$\gamma_4(u,v,i) = \gamma(v) = \frac{1}{|N_k(v)|} \sum_{w \in N(v)} \left(\frac{1}{|I_v \cap I_w|} \sum_{j \in I_v \cap I_w} \left[1 - \frac{|\tilde{r}(v,j;w) - r(v,j)|}{R_r} \right] \right).$$

Finally, a performance predictor inspired by the clarity score defined for query performance [Cronen-Townsend et al. 2002], was proposed in [Bellogín et al. 2010; Bellogín et al. 2011], considering its adaptation to predict neighbor performance in CF. In essence, the clarity score captures the lack of ambiguity (uncertainty) in a query, by computing the distance between the language models induced by the query and the collection, via the Kullback-Leibler divergence. In the same way as query clarity captures the lack of ambiguity in a query, **user clarity** thus computed is expected to capture the lack of ambiguity in a user's preferences. Hence, the amount of uncertainty involved in a user's profile is hypothesized to be a good predictor of her performance. Thus, the larger the following value is, the lower the uncertainty and the higher the expected performance:

$$\gamma_5(u,v,i) = \gamma(v) = KLD(v \,\|U \setminus \{u\}) = \sum_{w \in U \setminus \{v\}} p(w|v) \log_2 \frac{p(w|v)}{p(w)}$$

In that work, the probabilistic models defined are based on smoothing estimations and conditional probabilities over users and items. Specifically, a uniform distribution is assumed for users and items, whereas the user-user probability is defined by an expansion through items as follows:

$$p(v|u) = \sum_{i \in I_u} p(v|i)p(i|u).$$

Conditional probabilities are linearly smoothed with the user probabilities and the maximum likelihood estimators, which finally depend on the rating given by a user towards an item; i.e., $p_{ml}(i|u) \propto r(u,i)$.

In addition to the integration of the above methods in the role of neighbor effectiveness predictors in our framework, we propose two novel predictors based on well-known quantities measured over the probability models defined in [Bellogín et al. 2010]: the entropy and the mutual information. Entropy as an information-theoretic magnitude measures the uncertainty associated with a probability distribution [Cover and Thomas 1991]. We may therefore assess the uncertainty involved in the system's knowledge about a user's preferences by the entropy of the item distribution (the probability to choose an item) given the information in the user profile. Hypothesizing that this uncertainty may be a relevant signal in the effectiveness of a user as a potential neighbor, we define an **entropy**-based predictor as follows:

$$\gamma_6(u,v,i) = \gamma(v) = -H(I_v) = \sum_{j \in I_v} p(j|v) \log_2 p(j|v).$$

Note that the uncertainty, measured in this way, can be due to the system state of knowledge about the user's tastes, or may come from the user himself (e.g. some users may have strong preferences, while others are more undecided), and both causes may similarly affect the neighbor effectiveness. In both cases, the predictor can be interpreted as the lack of ambiguity in a user profile.

The second information-theoretic magnitude we propose to use over the probability models presented above is the mutual information. To be precise, the mutual information is a quantity computed between two random variables that measure the mutual dependence of the variables, or, in other terms, the amount of uncertainty that knowing either variable provides about the other [Cover & Thomas 1991]. Here, we propose to adapt this concept, and compute the **mutual information** between the neighbor and the rest of the community, in order to assess the uncertainty involved in the neighbor's preferences. For this, instead of computing the mutual information over all the events in the sample space for both variables (users), we fix one of them (for the current neighbor) and move along the other dimension:

$$\gamma_7(u,v,i) = \gamma(v) = I(v; U \setminus \{u\}) = \sum_{w \in U \setminus \{u\}} p(w|v) \log_2 \frac{p(w|v)}{p(v)p(w)}.$$

4.2.2. User-Item Predictors

User-item predictors are more difficult to apply because of their higher vulnerability to data sparsity. In a bi-dimensional user-item input space, less observations can be associated to each input data point, whereby the confidence on the predictor outcome is lower, as it can be biased to outliers or unusual users or items. Despite this difficulty, a local trust metric based on the target user and item is proposed in [O'Donovan and Smyth 2005]. This metric is called **item-level trust**, and aims to discriminate reliable neighbors depending on the current item, since the same user may be more trustworthy for predicting ratings for certain items than for others. The formulation of this predictor can be seen as a particularization of γ_1, but constraining the recommendation set only to the pairs in which the current item is involved:

$$\gamma_8(u,v,i) = \gamma(v,i) = \frac{|\{(c_k, i_k) \in CorrectSet(v) \colon i_k = i\}|}{|\{(c_k, i_k) \in RecSet(v) \colon i_k = i\}|}.$$

4.2.3. User-User Predictors

User-user predictors based on local trust measures have been studied further than user-item predictors in the literature in this area, since the former are able to represent how much a user can be trusted by another, and allow for different interpretations of the relation between users. These measures have been often researched in the scope of social networks, and the users' explicit links in this context [Ziegler and Lausen 2004; Massa and Avesani 2007], along with several trust measures based on ratings, as we shall show below. In this way, although social-based metrics could be smoothly integrated in our framework, we focus here on a complementary view on trust though, where predictors are defined based on ratings, and leave that other type of predictors as future work.

A first very simple neighbor reliability criterion one may consider is the amount of common experience with the target user, that is, the amount of information upon which the two users can be compared. If we define "user experience" in this context as the set of items the user has interacted with, we can define a predictor embodying this principle as:

$$\gamma_9(u,v,i) = \gamma(u,v) = |I_u \cap I_v|.$$

We shall refer to this predictor as **user overlap**. This predictor will serve as a basis for subsequent predictors, since most of them will depend on the items rated by both users. For instance, it has a clear use in assessing the reliability of the inter-user similarity assessments, which has been applied in the literature under a more practical, ad-hoc manner [Bellogín et al. 2013]. Specifically, Herlocker et al. (2002) proposed the introduction of a weight on the similarity function, where the latter is devalued when it has been based on a small number of co-rated items. We can formulate **Herlocker's significance weighting** predictor as follows:

$$\gamma_{10}(u,v,i) = \gamma(u,v) = \frac{|I_u \cap I_v|}{n_H} \text{ if } |I_u \cap I_v| < n_H; 1 \text{ otherwise},$$

where n_H is the minimum number of co-rated items that two users should have in common in order to avoid similarity penalization. A value of $n_H = 50$ has proved empirically to work effectively. A variation of this scheme was proposed in [McLaughlin and Herlocker 2004], to which we shall refer as **McLaughlin's significance weighting**:

$$\gamma_{11}(u,v,i) = \gamma(u,v) = \frac{\max(|I_u \cap I_v|, n_{Mc})}{n_{Mc}}.$$

This predictor is aimed to be equivalent to the Herlocker's significance weighting (γ_{10}) formulation when $n_{Mc} = n_H$. However, we note that γ_{10} and γ_{11} represent different concepts, and are not fully equivalent. For instance, as noted in [Ma et al. 2007], γ_{11} may return values larger than 1 when $|I_u \cap I_v| > n_{Mc}$, while γ_{10}, by definition, always returns a value in the (0,1] interval. Alternatively, the following variant can be drawn from [Ma et al. 2007], which is just a more compact reformulation of γ_{10}:

$$\gamma_{12}(u,v,i) = \gamma(u,v) = \frac{\min(|I_u \cap I_v|, n_M)}{n_M}.$$

A more elaborated predictor can be found in [Weng et al. 2006]. The rationale behind it is to consider two situations: whether user u takes into account the recommendation made by neighbor v or not; in this sense, trustworthiness is defined as the reduction in the proportion of incorrect predictions of going from the latter situation to the former. The definition of this predictor, denoted as **user's trustworthiness**, is the following:

$$\gamma_{13}(u,v,i) = \gamma(u,v) = \frac{1}{|R|^2 - \sum_x n(u,v;x,\cdot)^2}\left[|R| \sum_x \sum_y \frac{n(u,v;x,y)^2}{n(u,v;\cdot,y)} - \sum_x n(u,v;x,\cdot)^2\right]$$

In this formulation, $|R|$ represents the number of allowed rating values in the system (e.g. in a 1 to 5 rating scale, $|R| = 5$), the function $n(u,v;x,y)$ represents the number of co-rated items on which v's ratings have the value y while u's ratings are x, that is, $n(u,v;x,y) = |\{(u,\cdot,x)\} \cap \{(v,\cdot,y)\}|$ when each rating tuple is represented as (a,b,c), given a user a, an item b, and a rating value c. In the same way, $n(u,v;x,\cdot) = \sum_y n(u,v;x,y)$ represents all the co-rated items between u and v rated with any rating value by user v, and, analogously, $n(u,v;\cdot,y) = \sum_x n(u,v;x,y)$. In this case, the assumed hypothesis is that trust is one's expectation of other's competence in reducing its uncertainty in predicting new ratings.

Finally, a user-user predictor can be defined based on the global trust deviation predictor defined above (γ_4). In fact, if we ignore the average along users, we can define **trust deviation** [Hwang et al. 2007] as follows:

$$\gamma_{14}(u,v,i) = \gamma(u,v) = \frac{1}{|I_u \cap I_v|} \sum_{j \in I_u \cap I_v} \left[1 - \frac{|\tilde{r}(u,j;v) - r(u,j)|}{R_r}\right]$$

This predictor identifies effective neighbors mainly based on how many trustworthy (understood as "accurate") recommendations a user has received from another.

Table I. Overview of the studied neighbor quality metrics.

Name	Description	Reference
μ_1: absolute error deviation	Average difference (deviation) between the user's true ratings and the neighbors' estimated ratings	[Rafter et al., 2009]
μ_2: sign of error deviation	Similar to μ_1 but only considering the sign of the rating prediction deviation	[Rafter et al., 2009]
μ_3: neighbor goodness	Difference of the system's performance when including and excluding the user's ratings	[Bellogín et al., 2010]
μ_4: user-neighbor goodness	Difference of the system's performance for the user when including and exclusing her neighbors' ratings	This paper

Table II. Overview of the studied neighbor performance preditors.

Type	Name	Rationale	Reference
User	γ_1: profile-level trust	Percentage of correct recommendations in which the user has participated as neighbor	[O'Donovan and Smyth 2005]
	γ_2: expertise trust	Similar to γ_1 but incorporating a compensation value for cases of small neighborhoods	[Kwon et al., 2009]
	γ_3: trustworthiness	Similarity between the user's ratings and the average ratings given by the community	[Herlocker et al., 2002]
	γ_4: global trust deviation	Average of the prediction error on co-rated items for each user	[Hwang et al., 2007]
	γ_5: user clarity	Lack of ambiguity in the user's ratings as a signal for predicting the user's performance	[Bellogín et al., 2010]
	γ_6: entropy	Uncertainty about the user's ratings based on her rated item distribution	This paper
	γ_7: mutual information	Mutual dependence between the user's ratings and the ratings of the rest of the community	This paper
User-item	γ_8: item-level trust	Similar to γ_1 but constraining the recommendation set to the pairs where a particular item appears	[O'Donovan and Smyth 2005]
User-user	γ_9: user overlap	Number of items shared by two users	
	γ_{10}: Herlocker's significance weighting	Similar to γ_9 but incorporating a compensation value for cases of small overlap	[Herlocker et al., 2002]
	γ_{11}, γ_{12}: McLaughlin's significance weighting	Similar to γ_9 but incorporating a minimum value for cases of small overlap	[McLaughlin and Herlocker, 2004]
	γ_{13}: user's trustworthiness	Reduction in the proportion of incorrect predictions when taking or not the neighbor's ratings into account	[Weng et al., 2006]
	γ_{14}: trust deviation	Similar to γ_4 but without averaging the deviations for every user	[Hwang et al., 2007]

5. EXPERIMENTS

We now report experiments in which the proposed neighbor effectiveness prediction framework is tested as follows. First, we check the correlation between the user-based predictors defined in Sections 4.2.1 and 4.2.3 (summarized in Table II), and the neighbor performance metrics proposed in section 4.1 (summarized in Table I), as a direct test of their predictive power; for the predictors defined in Section 4.2.2 (user-item predictors, also in Table II) we cannot analyze their correlation because we have no neighbor performance metric depending on both the target user and an item available. After that, we test the usefulness of the predictors to enhance the final performance of memory-based algorithms, by using the predictors' values in the selection and weighting of neighbors, that is, taking the predictors as the scoring function in equation 2.

Our experiments have been carried out on two versions of the MovieLens dataset, namely the 100K and 1M versions, and on a dataset from Yahoo! Music. One version of MovieLens (100K) has 943 users, 1,682 items, and 100,000 ratings, whereas the 1M version has 6,040 users, 3,900 items, and one million ratings. For these datasets, we performed a 5-fold cross validation using five random 80-20% disjoint splits of the rating set (in MovieLens 100K we used the partition included in the dataset distribution). The Yahoo! dataset contains ratings for songs collected from two different sources: ratings supplied by users during normal interaction with Yahoo! Music services, and ratings for randomly selected songs collected during an online survey conducted by Yahoo! Research. The rating data has been divided into a training set, and a test set. The test set consists of the 54,000 ratings for randomly selected songs, while the training set consists of approximately 300,000 user-supplied ratings.

Additionally, for the user-based CF method, we use Pearson's correlation as the similarity measure between users, and a varying neighborhood size (k), which is a parameter with respect to which the results are examined.

5.1 Correlation Analysis

We analyze the correlation between neighbor quality metrics and neighbor performance predictors in terms of the Pearson's and Spearman's correlation metrics. The correlation provides a measure of the predictive power of the neighbor effectiveness prediction approaches: the higher the (absolute) correlation value, the better the predictor estimates the positive neighbor effect on the recommendation accuracy. The sign of the correlation coefficient represents whether the two variables involved –neighbor quality metric and neighbor performance predictor– are directly or inversely correlated.

Tables III and IV show the correlation values obtained on the MovieLens 100K dataset for the user-based predictors. We may observe how Spearman's correlation values are consistent but slightly higher than Pearson's, thus evidencing a non-linear relationship between the quality metrics and the performance predictors. We associate a sign to each quality metric, indicating whether the metric is direct (denoted as '+') or inverse (denoted with '-'), according to the expected sign of the correlation with the predictor, i.e., a metric is direct if the higher its value, the better the true neighbor performance.

Table III. Pearson's correlation between the proposed neighbor quality metrics and neighbor performance predictors in the MovieLens 100K dataset. Next to the metric name, an indication about the sign of the metric – direct(+) or inverse(-) – is included. Not significant values for a $\mathbf{p}$-value of $\mathbf{0.05}$ are denoted with an asterisk (*).

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	-0.21	+0.14	+0.17
Entropy	-0.18	+0.12	+0.18
Expertise	-0.62	+0.25	+0.03
Global Trust Deviation	-0.35	+0.08	-0.01
Mutual Information	-0.20	+0.12	+0.17
Profile Level Trust	+0.62	-0.24	-0.04*
Trustworthiness	-0.21	+0.20	+0.03

Table IV. Spearman's correlation between quality metrics and performance predictors in the MovieLens 100K dataset.

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	-0.30	+0.21	+0.16
Entropy	-0.22	+0.15	+0.17
Expertise	-0.65	+0.30	+0.02
Global trust deviation	-0.38	+0.11	-0.03
Mutual Information	-0.25	+0.17	+0.16
Profile Level Trust	+0.65	-0.30	-0.02
Trustworthiness	-0.24	+0.25	+0.03

The absolute error deviation (μ_1) metric presents higher values when the neighbor's prediction is less accurate, being thus an inverse neighbor metric. The other two metrics, sign of error (μ_2) and neighbor goodness (μ_3), are, by definition, direct neighbor metrics, since the former indicates how many times a recommendation from the neighbor has been made in the right direction, whereas the later represents the change in error between excluding a particular user in the neighborhood or including her, and thus, the larger this error, the "better" neighbor this user is.

We may observe in Table III that, except for some of the predictors which obtain very low absolute values (< 0.10), the four quality metrics are consistent with each other. This consistency is evidenced by the way the predictors correlate with the different metrics: some of the predictors obtain the correct correlations in every situation, that is, positive correlation with direct metrics and negative correlation with the inverse metric (like the clarity predictor), while other predictors obtain opposite values for all the metrics, that is, positive correlations with the inverse metric and negative correlations with direct metrics (such as the profile level trust predictor).

Tables V and VI show the correlation values obtained on the MovieLens 1M dataset. We may observe how the trend in correlation is very similar to the behavior observed on the 100K dataset, and thus, similar conclusions can be drawn from it. There are, however, some changes in the absolute values of the correlation scores for some combinations of performance predictor and metric. For instance, the clarity predictor and the neighbor goodness metric obtain larger values in this dataset, while the correlation between entropy and absolute error deviation is smaller.

Table V. Pearson's correlation between quality metrics and performance predictors in the MovieLens 1M dataset. All the values are significant for a $\mathbf{p}$-value of $\mathbf{0.05}$.

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	-0.14	+0.02	+0.40
Entropy	-0.07	-0.08	+0.39
Expertise	-0.95	+0.70	-0.06
Global Trust Deviation	-0.55	+0.36	-0.24
Mutual Information	-0.17	+0.13	+0.30
Profile Level Trust	+0.83	-0.55	+0.04
Trustworthiness	-0.27	+0.36	+0.03

Table VI. Spearman’s correlation between quality metrics and predictors in the MovieLens 1M dataset.

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	-0.16	+0.04	+0.35
Entropy	-0.03	-0.10	+0.37
Expertise	-0.94	+0.69	-0.09
Global trust deviation	-0.54	+0.39	-0.25
Mutual information	-0.16	+0.04	+0.31
Profile level trust	+0.94	-0.69	+0.09
Trustworthiness	-0.25	+0.37	+0.02

It is important to note that the number of points used to compute the correlation values is different in the two datasets; there are less than 1,000 points in MovieLens 100K (with 943 users), and more than 6,000 points in MovieLens 1M dataset. This difference affects the significance of the correlation results. The confidence test for a Pearson’s correlation, modeled as the t-value of a t-distribution (assuming normality) with $N-2$ degrees of freedom (being N the size of the sample), is defined by the following equation:

$$t = r\sqrt{\frac{N-2}{1-r^2}}$$

The t-value depends on the size of the sample, and thus, the significance of a Pearson’s correlation value r may change for different sample sizes [Snedecor and Cochran 1980]. In particular, for small samples, we may eventually obtain strong but non-significant correlations; whereas for large samples, on the other hand, we may obtain significant differences, even though the strength of the correlation values may be lower. In our experiments, for MovieLens 100K, the correlations are significant for a p-value of 0.05 when $r > 0.05$, and in the 1M dataset, when $r > 0.02$. Hence, in Table III, there is only one non-significant correlation value (denoted with an asterisk), whereas in Table V, all the results are statistically significant. The above also applies to the Spearman’s correlations reported in Tables IV and VI [Snedecor and Cochran 1980; Zar 1972].

Analyzing in more detail the reported results for both datasets, we observe that the profile level trust predictor consistently obtains direct correlation values with the inverse metrics, and inverse correlation values with the direct metrics. This predictor gives higher scores to neighbors with larger deviations in their prediction error, which would result on a bad performance prediction, since its predictions are not in the same direction than the performance metrics. The expertise and global trust deviation predictor obtain strong correlations with the absolute error deviation metric, although their correlations with respect to the neighbor goodness metric are negligible, especially for the first predictor, in both datasets. At the other end of the spectrum, the clarity, entropy, and mutual information predictors obtain strong correlation values with the neighbor goodness and moderate correlations with the rest of metrics, which make these predictors good candidates for successful neighbor performance predictors. Finally, the trustworthiness predictor obtains a significant amount of correlation with respect to the absolute error deviation and sign of error metrics, although its correlation with respect to the neighbor goodness is very low. This predictor thus seems to be useful on estimating how accurate the neighbor may be in terms of the error in a user basis, but probably not as a global metric.

To provide a more general view of the predictive power of the neighbor effectiveness prediction approaches, we show in Tables VII and VIII the correlation results for the Yahoo! dataset. We can observe that the results are similar to those found for the MovieLens datasets, although we can see more differences for some predictors with respect to the previous datasets. For instance, the clarity predictor is not consistent in the three neighbor quality metrics, since it shows a positive correlation with the inverse metric (absolute error deviation) and a negative value with the sign of error metric (direct). Furthermore, the sign of the error metric does not agree with the neighbor goodness metric, like in Tables III-VI. Aside from these differences, the trend in predictive power is comparable to the ones presented before, where clarity, entropy, and trustworthiness have strong predictive capabilities, whereas the mutual information predictor shows worse results.

Table VII. Pearson's correlation between quality metrics and performance predictors in the Yahoo! dataset. Not significant values for a $\mathbf{p}$-value of $\mathbf{0.05}$ are denoted with an asterisk (*).

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	+0.28	-0.28	+0.33
Entropy	+0.00*	-0.02*	+0.26
Expertise	-0.11	-0.06	+0.09
Global Trust Deviation	-0.29	+0.16	+0.02
Mutual Information	+0.04	-0.06	-0.27
Profile Level Trust	+0.11	+0.06	-0.10
Trustworthiness	-0.16	+0.16	+0.03*

Table VIII. Spearman's correlation between quality metrics and predictors in the Yahoo! dataset.

Neighbor performance predictor	Neighbor quality metric		
	Absolute error deviation μ_1 (-)	Sign of error μ_2 (+)	Neighbor goodness μ_3 (+)
Clarity	+0.31	-0.33	+0.34
Entropy	-0.01	-0.03	+0.23
Expertise	-0.07	-0.09	+0.08
Global trust deviation	-0.19	+0.12	-0.01
Mutual information	+0.09	-0.07	-0.26
Profile level trust	+0.07	+0.09	-0.08
Trustworthiness	-0.17	+0.18	+0.03

Table IX shows the correlations obtained for user-user neighbor predictors and the proposed user-neighbor goodness metric. Due to the high dimensionality of the vectors involved in this computation, we have only considered those users which have at least one item in common, since both the predictors and the metric would return the same score –a zero– in any other case. Despite this fact, Pearson's correlation is almost neglibible for all the datasets, and thus we show here only Spearman's correlation coefficient. We can observe that in MovieLens datasets the correlation for the McLaughlin's significance weighting predictor is stronger than for the rest of the predictors, which evidences some non-linear relation between this predictor and the metric. In the next section, we shall show that this function is one of the best performing predictors among the evaluated neighbor scoring functions. The situation in the Yahoo! dataset is slightly different, since all the predictors have some correlation, except for trust deviation; again, in the next section we shall see that this predictor is among the worst performing neighbor scoring functions. These results, thus, confirm the usefulness of the proposed neighbor performance metrics, since they are able to discriminate which neighbor performance predictors are able to capture interesting properties between the user and her neighbors.

Table IX. Spearman's correlation between the user-neighbor goodness and user-user predictors.

User-neighbor performance predictor	Dataset		
	Movielens 100K	Movielens 1M	Yahoo!
Herlocker	0.03	0.02	0.31
McLaughlin	0.12	0.11	0.32
Trust Deviation	0.01	0.01	0.11
User Overlap	0.03	0.02	0.31
User's Trustworthiness	-0.02	-0.01	0.31

In summary, we have observed that most of the performance predictors agree with respect to the different performance metrics, and in general, the correlations computed between neighbor quality metrics and neighbor performance predictors are statistically significant.

5.2 Performance Analysis

The results reported in the previous section show that some of the studied predictors have the ability to capture neighbor performance, and because of that we hypothesize that they could be used to improve the accuracy of a recommendation model. But this hypothesis has to be checked, since the metric against which we measure the neighbor goodness is not the same as the final recommendation performance metric we aim to optimize. With the experiments we report next, we aim to confirm the usefulness of the proposed predictors, the validity of the proposed metrics as useful references to assess the power of the predictive methods, and the usefulness of the overall framework as a unified approach to enhance neighbor-based collaborative filtering.

In order to achieve this, we test the integration of the neighbor predictors into a neighbor selection and weighting scheme for user-based CF, as described in Section 3.1. Besides testing the effectiveness of the predictors, this experiment provides for observing to what extent the correlations obtained in the previous section correspond with improvements in the final performance of those predictors.

We provide recommendation accuracy results on the MovieLens 1M and Yahoo! datasets. Those obtained on the MovieLens 100K dataset are not reported here since they had similar trends to those of MovieLens 1M. Figures 1 and 2 show the Root Mean Square Error (RMSE) of the Resnick's CF adaptation proposed in Equation 2 when used for different neighbor selection and weighting approaches for MovieLens 1M, the equivalent figures for the Yahoo! dataset are presented in Figures 3 and 4. The curves at the top of the figures represent the values obtained when neighbor performance predictors are used for neighbor weighting, that is, the standard neighbor selection strategy is used ($g = g_0$ in Equation 2). Furthermore, in each approach a different aggregation function is used: whether the harmonic mean between the predictor score and the similarity value (function $f = f_2$, on the right) or the projection function ($f = f_4$, on the left) in order to ignore the similarity. The curves at the bottom of the figures show the neighbor selection approach ($g = g_1$ in Equation 2) along with the same neighbor weighting functions described above (i.e., f_2 on the right and f_4 on the left). The rest of the aggregation functions, such as average (f_1) and product (f_3), were also evaluated for neighbor selection and weighting, but they provided results equivalent to those of the harmonic mean. For this reason, they have been omitted in the figures to avoid cluttering them. We believe this equivalence may be due to the normalization factor included in the CF formulation, since it would cancel out the weights obtained by the harmonic, average, and product functions in the same way.

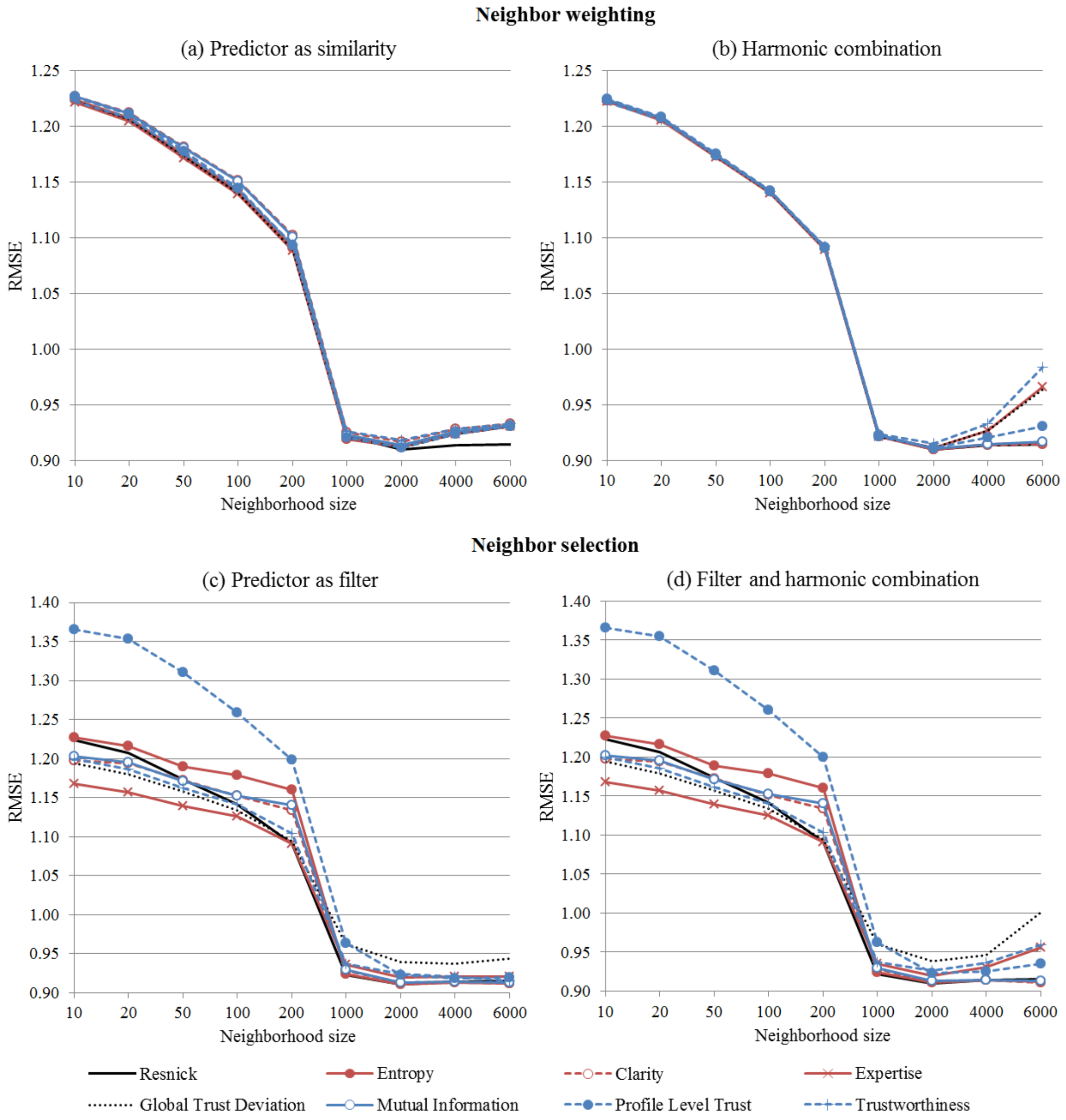

Fig. 1. Performance comparison for user-based predictors and different neighborhood sizes in MovieLens 1M.

Figure 1 shows the accuracy results when only user-based neighbor predictors are evaluated in the MovieLens 1M dataset. We observe that, independently from the neighborhood size, using performance predictors as similarity scores does not lead to large differences with respect to the baseline. These results are compatible with those presented in [Weng et al. 2006], where the improvement in RMSE is not very high (ΔMAE < 0.05 in that work). Note that the accuracy trend on neighborhood size in our experiments does not fully match the ones reported in [Herlocker et al. 2002], where the optimal neighborhood for MovieLens 100K is much smaller (around 50) than in our experiments. This is due to a difference in how the neighborhoods are computed: whereas Herlocker and colleagues take a different neighborhood for each user–item pair, we take a fixed neighborhood of size *k* (independent of target items) for every user. As a consequence, the optimal neighborhood size to reach a certain accuracy level in our implementation is needly larger, since the number of users among *k* preselected neighbors who have rated a target item is usually quite lower than *k* (thus the equivalent effective neighborhood size is much smaller). This version of the memory-based CF implementation is common in the field [Ren et al. 2012; Guo et al. 2012; Schclar et al. 2009; Clements et al. 2007] and has advantages in terms of computational cost, as neighborhoods can be built offline. In exchange, larger neighborhood sizes are needed to obtain the best performance values.

For the sake of clarity, in Tables X and XI, we present the error values for a horizontal cut of the left curves; specifically, when the neighborhood size is 50. We can observe that some predictors do improve Resnick's accuracy. Regarding the use of the harmonic mean as aggregation function (curves on the right), similar results are obtained except for very large neighborhood sizes, for which some of the performance predictors produce worse results than the baseline, probably due to the amount of noise created by considering too many neighbors.

The curves at the bottom of the figures represent the accuracy results for neighbor selection strategies. In this case, some of the predictors lead to worse performance than the baseline, particularly the profile level trust (γ_1). This situation is consistent with the correlations observed for MovieLens 1M in the previous section, since this predictor obtained inverse correlations with the different metrics, i.e., direct correlation values with inverse metrics, and inverse values with direct metrics. Moreover, as predicted by the correlation analysis, trustworthiness (γ_3), mutual information (γ_7), and clarity (γ_5) result in some of the best performing recommenders (with strong correlations), as shown in the figures and in Table XI, along with expertise (γ_2) and global trust deviation (γ_4), which obtained more moderated correlation values.

Table X. Detail of the accuracy in MovieLens 1M of baseline vs. recommendation using neighbor weighting; here, performance predictors are used as similarity scores (50 neighbors).

	RMSE		RMSE
Resnick	1.174	Resnick	1.174
Clarity	1.181	Herlocker	1.175
Entropy	1.175	Item-level Trust	1.264
Expertise	1.171	McLaughlin	1.174
Global Trust Deviation	1.173	Trust Deviation	1.173
Mutual Information	1.180	User Overlap	1.175
Profile Level Trust	1.177	User's Trustworthiness	1.175
Trustworthiness	1.175		

Table XI. Detail of the accuracy in MovieLens 1M of baseline vs recommendation using neighbor selection; here, performance predictors are used for filtering (50 neighbors).

	RMSE		RMSE
Resnick	1.174	Resnick	1.174
Clarity	1.172	Herlocker	1.156
Entropy	1.189	Item-level Trust	1.843
Expertise	1.139	McLaughlin	0.581
Global Trust Deviation	1.158	Trust Deviation	1.168
Mutual Information	1.171	User Overlap	1.146
Profile Level Trust	1.310	User's Trustworthiness	1.174
Trustworthiness	1.162		

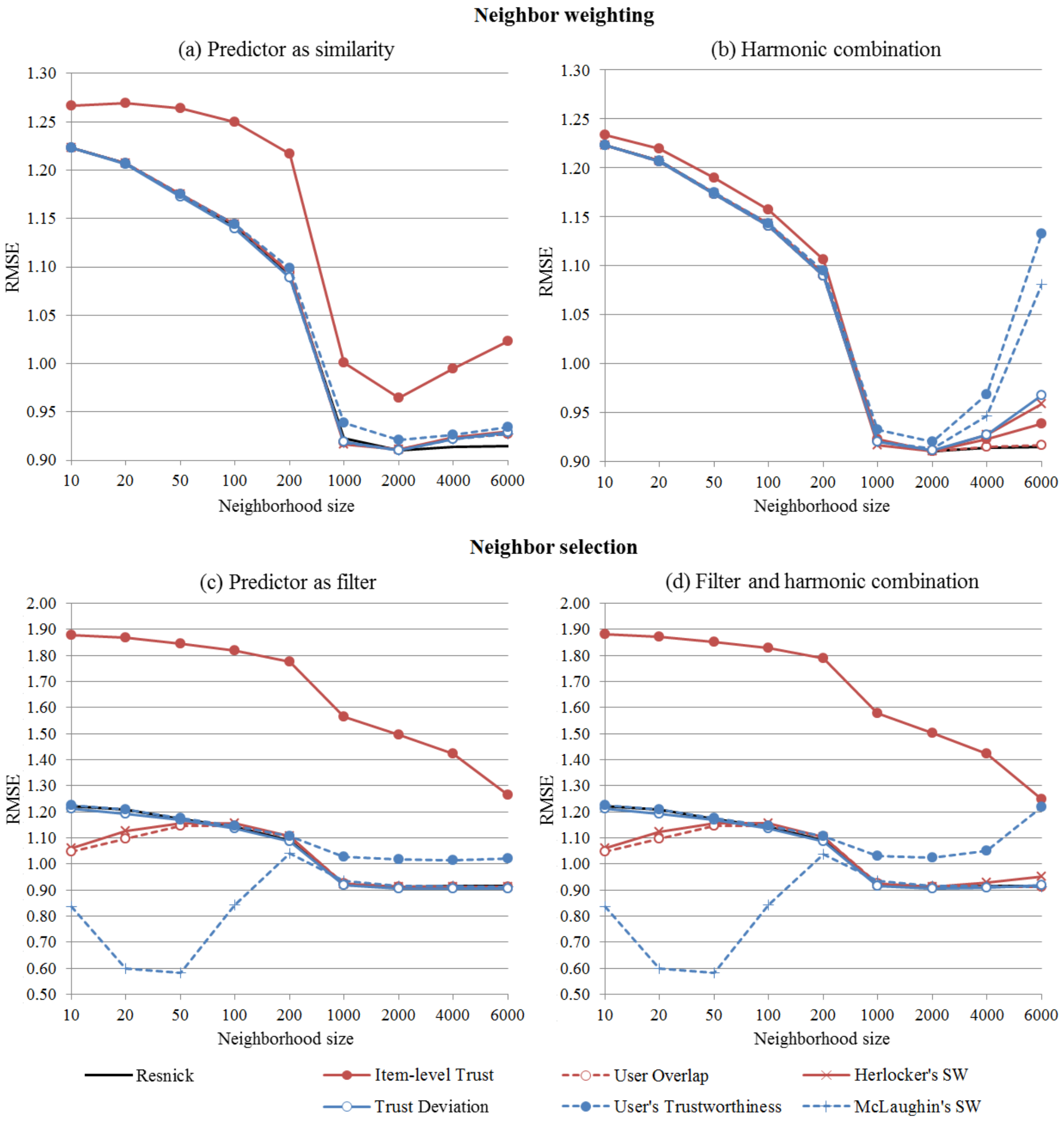

Fig. 2. Performance comparison using user-item and user-user predictors for different neighborhood sizes in MovieLens 1M.

In Figure 2, we can see how user-item and user-user neighbor predictors affect the performance of CF recommenders, again for MovieLens 1M. The curves in the top show that most of the predictors obtain a similar performance to that of the baseline, except for the item-level trust (γ_8), the performance of which is much worse than Resnick's. Table X shows the specific error values for these recommenders. It is interesting to note that the performance of this predictor is drastically improved when using the harmonic mean as the aggregation function (shown on the right side of the figure). Similarly to user-based neighbor predictors (Figure 1), some of the user-item and user-user predictors decrease their accuracy with large neighborhoods; in this case, user's trustworthiness (γ_{13}) and McLaughlin's significance weighting (γ_{12}) are the most representative examples.

A different conclusion results when neighbor selection is analyzed (curves at the bottom). Two of the predictors are characterized by a much better (McLaughlin's significance weighting, γ_{12}) or worse (item-level trust, γ_8) final performance, independently from the weighting aggregation function. Table XI shows the specific error values obtained for each of these predictors. It is interesting how the McLaughlin's predictor, despite its inability to boost good neighbors (see top figures), seems to be very useful for neighbor selection. This effect, nonetheless, is attenuated when the neighborhood increases, since in that situation, selection methods have to deal with too many users in each neighborhood. We believe the reason why this predictor is very good for neighbor selection is because it gives higher scores to those neighbors that have more items in common with the active user, and thus the confidence in the computation of the similarity values between the neighbor and the active user will be higher. It is worth noting that, to the best of our knowledge, this function has never been used for neighbor selection, since its original motivation was to penalize the similarity value whenever it has been based on a small number of co-rated items. However, by plugging this function into the framework, and measuring its predictive power for user-neighbor performance, a novel application naturally emerges and provides very good results.

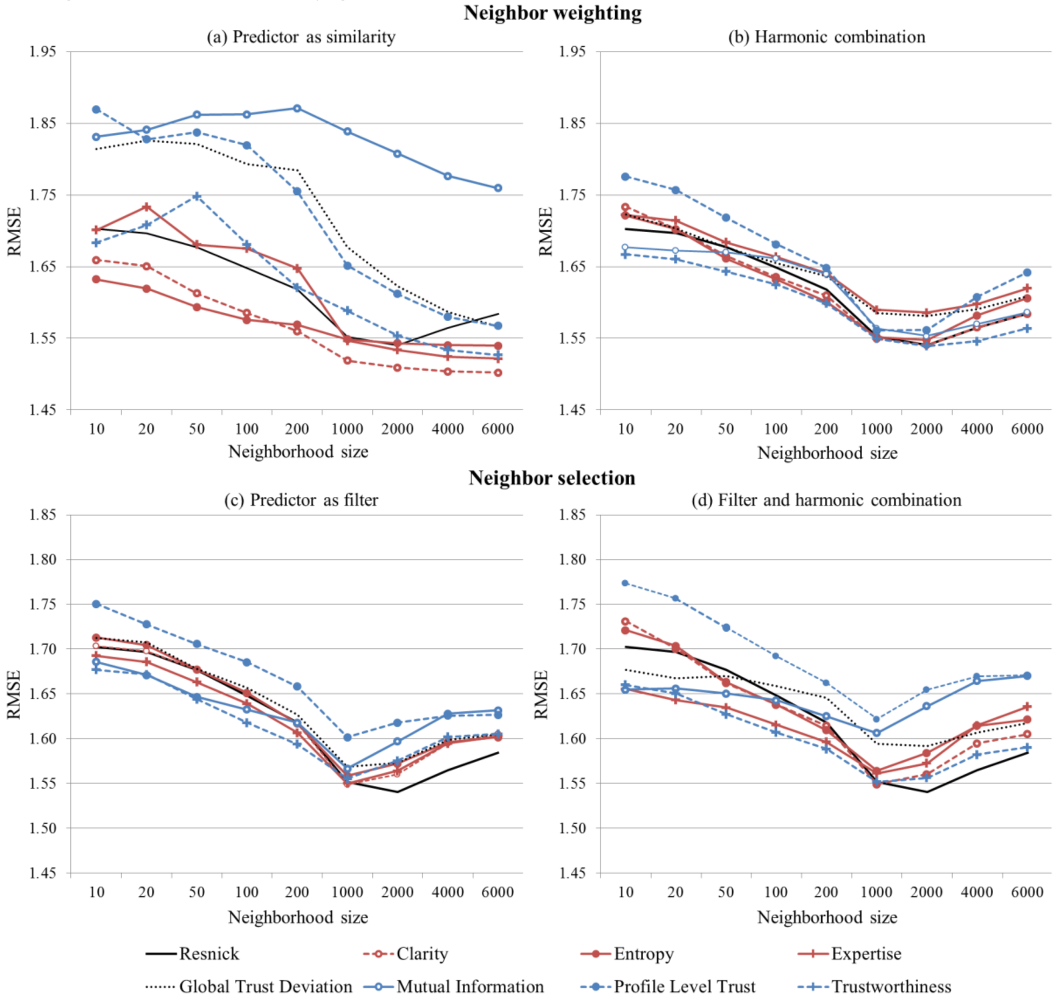

Fig. 3. Performance comparison for user-based predictors and different neighborhood sizes in the Yahoo! dataset.

Regarding the results in the Yahoo! dataset, we can observe in Figures 3 and 4 that the performance values are also consistent with the correlations presented in Tables VII-IX. More specifically, the profile level trust predictor always performs worse than the baseline, just as we observed in the MovieLens 1M dataset, and in agreement with its low correlation values presented before. Furthermore, the better performing user-based predictors are, again, clarity, trustworthiness, and expertise.

The performance of the entropy predictor seems to depend on the way it is introduced in the recommendation technique, which may be related to its negligible correlations for two out of the three neighbor quality metrics. Moreover, in contrast with what was observed in the MovieLens 1M dataset, the performance of the mutual information predictor is worse now, something already anticipated by its low correlations with the absolute error deviation and the sign of error metrics, and its negative correlation with neighbor goodness.

Additionally, in Figure 4 we can observe that the item-level trust, like in the MovieLens 1M dataset, is among the worst performing predictors. Another bad-performing predictor is the trust deviation, whose performance, in contrast to what was found in MovieLens, decreases drastically, especially when used as similarity or filter; this result, however, is consistent with the correlations presented in Table IX, that anticipated that this predictor would not be as good as the rest. Apart from this, the user's overlap and McLaughlin's significance weighting yield very positive results, as in the previous experiment with the MovieLens datasets.

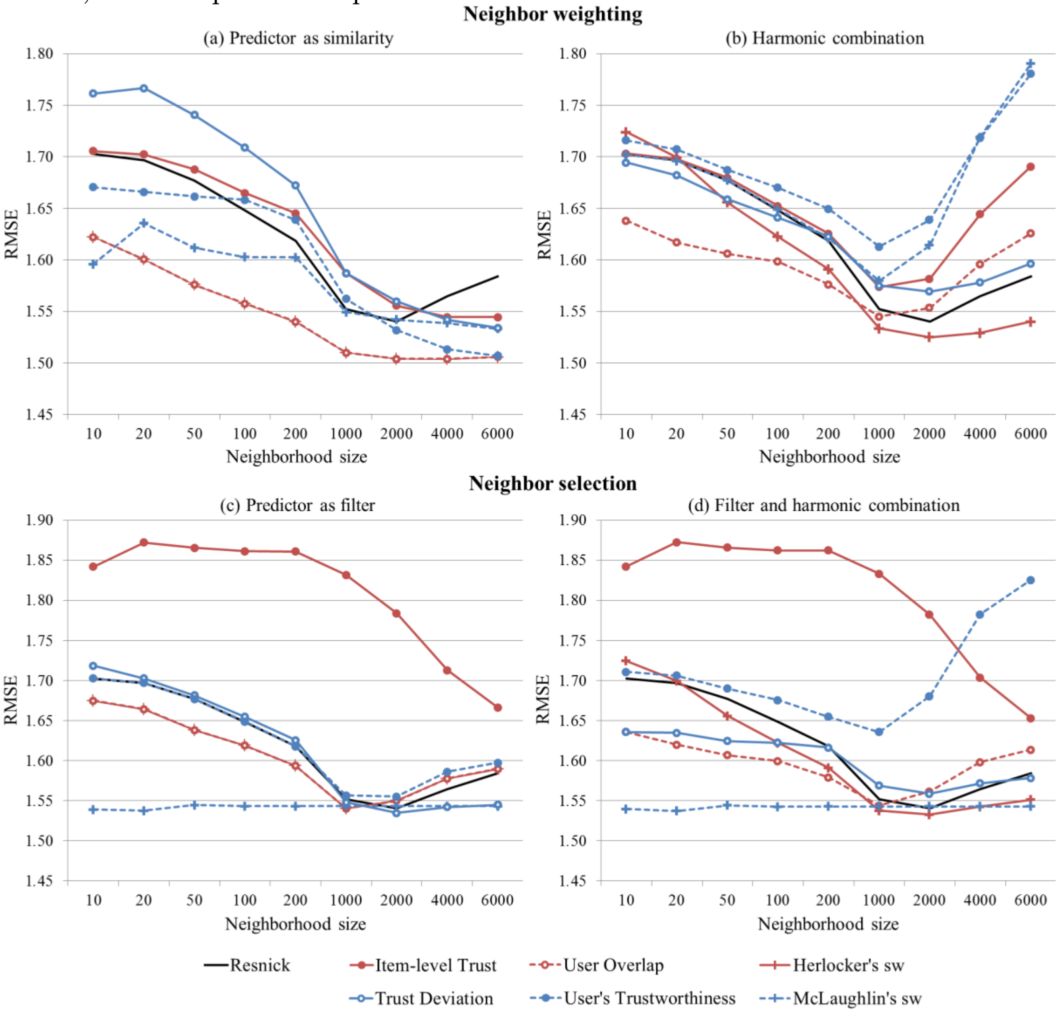

Fig. 4. Performance comparison using user-item and user-user predictors for different neighborhood sizes in Yahoo!.

In summary, we have been able to validate the proposed user-user neighbor performance metric, and the different evaluated user-user neighbor performance predictors. We have obtained positive results when this type of predictors has been introduced and compared against the baseline in the different aggregation strategies and configurations; and these results are consistent with the correlations obtained between the predictors and the performance metrics. In particular, McLaughlin's significance weighting obtains an improvement up to 55% in accuracy (i.e., error decrease) when this predictor is used to select the neighbors which will further contribute to the rating prediction. Besides, the (Spearman's) correlation for this predictor is positive and strong in the MovieLens datasets, in contrast to the values obtained for the rest of user-user predictors, which did not improve the accuracy of the baseline. At the same time, in the Yahoo! dataset, these correlations are also positive, and allow to discriminate between a bad user-user predictor (trust deviation) and the rest of the predictors.

In this situation, a possible drawback of the conducted analysis is that we have not been able to define neighbor performance metrics based on user-item pairs, and thus the user-item neighbor performance predictors are out of the scope of the developed correlation analysis. Nevertheless, results show that the only user-item neighbor performance predictor defined here, the item-level trust, is not able to outperform the baseline recommender. We believe this fact, which is in contradiction with what was reported in [O'Donovan and Smyth 2005], may be caused by the different variables taking place in our evaluation, such as the datasets (MovieLens 1M instead of MovieLens 100K, and Yahoo!), the neighborhood size (not specified in the original paper), and the several aggregation functions and combinations used across our experiments.

5.3 Discussion

The reported experiments provide empirical evidence of the usefulness of the proposed framework, and the specific proposed predictors, as an effective approach to enhance the accuracy of memory-based CF. As described in the preceding sections, the methodology comprises two steps, one in which the predictive power of neighbor predictors is assessed, and one in which the predictors are introduced in the CF scheme to enhance the effectiveness of the latter. Our experiments confirm a strong correlation for some of the predictors – both user predictors and user-user predictors – and this has been found to correspond with final accuracy enhancements in the recommendation strategy: the predictors that obtained strong direct correlations with the performance metrics were the best performing dynamic strategies; the profile level trust predictor, which obtained inverse correlation values with respect to the neighbor performance metrics, was the worst performing dynamic strategy. In light of these results, it could be further investigated whether the actual correlation values between neighbor performance predictors and neighbor performance metrics could be used to infer how each predictor should be incorporated into the memory-based CF as a neighbor scoring function, since there is no obvious link between the ranking of best performing scoring functions and the strength of their corresponding correlations. As a starting point, only the sign of the correlation could be considered, using either the raw neighbor predictor score (for positive correlations) or its inverse (for negative values). Then, this rationale could be further elaborated and evaluated in order to check whether the performance improvements are consistent.

Research on finding functions with strong correlation power with respect to neighbor performance metrics could be a very interesting area by itself, since it could have many different final applications; here, we have experimented with variations in neighbor selection and weighting for memory-based CF but those predictors (functions) could also be used for active learning [Elahi 2011] or for providing more meaningful explanations [Marx et al. 2010], depending or based on the predicted performance of a particular user's neighbors. In the same way the performance prediction research in IR has been mainly focused on defining predictors with strong correlation values, not usually investigating the different uses which could be drawn from such knowledge, in CF it may be possible to investigate the effect of neighbor selection separate from the accuracy of the algorithm, since we have some certainty that the ability of performance predictors to capture the neighbor's performance is related with the correlation values obtained between the predictor and the neighbor performance metric. Then, different applications to integrate the output from neighbor performance predictors into the recommendation process could be derived, such as those mentioned above.

6. CONCLUSIONS

We have proposed a theoretical framework for neighbor selection and weighting in user-based CF systems, based on a performance prediction approach, drawing from query performance methodology in the IR field. By viewing the neighbor-based CF rating prediction task as a case of dynamic output aggregation, our approach places user-based CF in a more general frame, linking to the principles underlying the formation of ensemble recommenders, or rank aggregation in IR. By doing so, it is possible to draw concepts and techniques from these areas, and vice versa. Our study thus provides a comparison of different state-of-the-art rating-based trust metrics and other neighbor scoring techniques, interpreted as neighbor performance predictors, and evaluated under this new angle. The framework provides for an objective analysis of the predictive power of several neighbor scoring functions, integrating different notions of neighbor performance into a unified view. The proposed methodology discriminates which neighbor scoring functions are more effective in predicting the goodness of a neighbor, and thus identifies which weighting functions are more effective in a user-based CF algorithm.

Drawing from different state-of-the-art neighbor scoring functions – cast as user, user-user, and user-item neighbor performance predictors – we have conducted several experiments in order to, first, check the predictive power of these functions, and second, validate them by comparing the final performance of neighbor-scoring powered memory-based strategies with that of the standard CF algorithm. We have also evaluated different ways to introduce these functions in the rating prediction formulation, namely for neighbor weighting, neighbor selection, and combinations thereof. In this context, methods where neighbor scoring functions were integrated outperform the baseline for different values of neighborhood size and predictor type.

We have proposed several neighbor performance metrics that capture different notions of neighbor quality. The evaluated performance predictors show consistent correlations with respect to these metrics, and some of them present particularly strong correlations. Interestingly, a correspondence is confirmed between the correlation analysis and the final performance results, in the sense that the correlation values obtained between neighbor performance predictors and neighbor performance metrics anticipate which predictors will perform better when introduced in a memory-based CF algorithm.

Apart from the performance predictors defined here, other predictors such as item or item-item could be defined [Weng et al. 2006; Ma et al. 2007] and easily incorporated in item-based algorithms. Additionally, performance predictors defined upon social data, such as the user's trust network, could be smoothly integrated into our framework and analyzed in the future. Furthermore, alternative neighbor performance metrics may be defined to check the predictive power of user-user and user-item predictors. These metrics, although based on a smaller amount of information, may help on better understanding which notions of the neighbor performance such predictors are capturing. In particular, our framework would allow for different interpretations of the user's performance, by modeling different neighbor performance metrics, whether they be more oriented to accuracy (using error metrics as in this paper) or ranking precision, or even towards alternative metrics such as diversity, coverage, or serendipity [Shani and Gunawardana 2011].

REFERENCES

ADOMAVICIUS, G. AND TUZHILIN, A. 2005. 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.

BELLOGÍN, A. AND CASTELLS, P. 2010. A performance prediction approach to enhance collaborative filtering performance. In *Proceedings of the 32nd European Conference on Information Retrieval (ECIR '10)*, Milton Keynes, UK, March 2010, volume 5993 of Lecture Notes in Computer Science, C. GURRIN ET AL., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 382-393.

BELLOGÍN, A., CASTELLS, P., AND CANTADOR, I. 2011. Predicting the performance of recommender systems: an information theoretic approach. In *Proceedings of the 3rd International Conference on the Theory of Information Retrieval (ICTIR '11)*, Bertinoro, Italy, September 2011, volume 6931 of Lecture Notes in Computer Science, AMATI, G., AND CRESTANI, F., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 27-39.

BELLOGÍN, A., CASTELLS, P., AND CANTADOR, I. 2013. Improving memory-based collaborative filtering by neighbour selection based on user preference overlap. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval (OAIR '13)*, Lisbon, Portugal, March 2013, Le Centre de Hautes Etudes Internationales D'Informatique Documentaire, Paris, France, 145-148.

CACHEDA, F., CARNEIRO, V., FERNÁNDEZ, V., AND FORMOSO, V. 2011. Comparison of collaborative filtering algorithms: limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Transactions on the Web 5, 1, 1-33.

CARMEL, D. AND YOM-TOV, E. 2010. Estimating the query difficulty for Information Retrieval. In *Synthesis lectures of information concepts, retrieval, and services*. Morgan & Claypool Publishers.

CLEMENTS, M., DE VRIES, A.P., POUWELSE, J.A., WANG, J., AND REINDERS, M.J.T. 2007. Evaluation of neighbourhood selection in decentralized recommendation systems. In *Proceedings of the Workshop on Large Scale Distributed Systems for Information Retrieval*.

COVER, T.M. AND THOMAS, J.A. 1991. *Elements of Information Theory*. Wiley-Interscience.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, B.W. 2002. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02)*, Tampere, Finland, August 2002, ACM, New York, NY, USA, 299-306.

DESROSIERS, C. AND KARYPIS, G. 2011. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, chapter 4, F., RICCI, L., ROKACH, B., SHAPIRA, P.B., KANTOR, Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 107-144.

EKSTRAND, M.D., RIEDL, J., AND KONSTAN, J.A. 2011. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction* 4, 2, 175-243.

GOLBECK, J. 2009. Trust and nuanced profile similarity in online social networks. *ACM Transactions of the Web* 3, 4, 12-32.

GOLBECK, J., AND HENDLER, J. 2006. FilmTrust: movie recommendations using trust in web-based social networks. In *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference, (CCNC '06)*, Las Vegas, Nevada, USA, January 2006, IEEE, 282-286.

GUO, G., ZHANG, J., AND THALMANN, D. 2012. A simple but effective method to incorporate trusted neighbors in recommender systems. In *Proceedings of the 20th User Modeling, Adaptation, and Personalization (UMAP '12)*, Montreal, Canada, July 2012, volume 7379 of Lecture Notes in Computer Science, MASTHOFF, K., MOBASHER, B., DESMARAIS, M.C., AND NKAMBOU, R., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 114-125.

HERLOCKER, J., KONSTAN, J.A., AND RIEDL, J. 2002. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval* 5, 4, 287-310.

HWANG, C.-S. AND CHEN, Y.-P. 2007. Using trust in collaborative filtering recommendation. In *Proceedings of the 20th International Conference on Industrial Engineering and other applications of Applied Intelligent Systems (IEA/AIE '07)*, Kyoto, Japan, June 2007, volume 4570 of Lecture Notes in Computer Science, chapter 105, OKUNO, H., AND ALI, M., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 1052-1060.

KWON, K., CHO, J., AND PARK, Y. 2009. Multidimensional credibility model for neighbor selection in collaborative recommendation. *Expert Systems with Applications* 36, 3, 7114-7122.

MA, H., KING, I., AND LYU, M. R. 2007. Effective missing data prediction for collaborative filtering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '07)*, Amsterdam, The Netherlands, July 2007, ACM, New York, NY, USA, 39-46.

MA, H., KING, I., AND LYU, M. R. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '09)*, Boston, MA, USA, July 2009, ACM, New York, NY, USA, 203-210.

ELAHI, M. 2011. Adaptive active learning in recommender systems. In *Proceedings of the 19th User Modeling, Adaptation, and Personalization (UMAP '11)*, Girona, Spain, July 2011, volume 6787 of Lecture Notes in Computer Science, KONSTAN, J.A., CONEJO, R., MARZO, J.L., AND OLIVER, N., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 414-417.

MARX, P., THURAU, T.H., AND MARCHAND, A. 2010. Increasing consumers' understanding of recommender results: a preference-based hybrid algorithm with strong explanatory power. In *Proceedings of the 4th ACM conference on Recommender systems (RecSys '10)*, Barcelona, Spain, September 2010, ACM, New York, NY, USA, 297-300.

MASSA, P. AND AVESANI, P. 2007. Trust-aware recommender systems. In *Proceedings of the 1st ACM conference on Recommender systems (RecSys '07)*, Minneapolis, MN, USA, October 2007, ACM, New York, NY, USA, 17-24.

MASSA, P. AND BHATTACHARJEE, B. 2004. Using trust in Recommender Systems: an experimental analysis. In *Proceedings of the 2nd International Conference on Trust Management (iTrust '04)*, Oxford, UK, March 2004, volume 2995 of Lecture Notes in Computer Science, JENSEN, C., POSLAD, S., AND DIMITRAKOS, T., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 221-235.

MCLAUGHLIN, M.R. AND HERLOCKER, J.L. 2004. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04)*, Sheffield, UK, July 2004, ACM, New York, NY, USA, 329-336.

O'DONOVAN, J. AND SMYTH, B. 2005. Trust in recommender systems. In *Proceedings of the 10th international conference on Intelligent user interfaces (IUI '05)*, San Diego, CA, USA, January 2005, ACM, New York, NY, USA, 167-174.

RAFTER, R., O'MAHONY, M., HURLEY, N., AND SMYTH, B. 2009. What have the neighbours ever done for us? A collaborative filtering perspective. In *Proceedings of the 17th User Modeling, Adaptation, and Personalization (UMAP '09)*, Trento, Italy, June 2009, volume 5535 of Lecture Notes in Computer Science, chapter 36, HOUBEN, G.-J., MCCALLA, G., PIANESI, F., AND ZANCANARO, M., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 355-360.

RASHID, A.M., KARYPIS, G., AND RIEDL, J. 2005. Influence in ratings-based recommender systems: an algorithm-independent approach. In *Proceedings of the 5th SIAM International Conference on Data Mining*, Newport Beach, CA, USA, April 2005, 556-560.

REN, Y., LI, G., ZHANG, J., AND ZHOU, W. 2012. The efficient imputation method for neighborhood-based collaborative filtering. In *Proceedings of the 21st ACM international conference on Information and knowledge management (CIKM '12)*, Maui, HI, USA, October 2012, ACM, New York, NY, USA, 684-693.

RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P. AND RIEDL, J. 1994. GroupLens: an open architecture for collaborative filtering of Netnews. In *Proceedings of the ACM conference on Computer supported cooperative work (CSCW '94)*, Chapel Hill, NC, USA, October 1994, ACM, New York, NY, USA, 175-186.

SCHCLAR, A., TSIKINOVSKY, A., ROKACH, L., MEISELS, A., AND ANTWARG, L. 2009. Ensemble methods for improving the performance of neighborhood-based collaborative filtering. In *Proceedings of the 3rd ACM conference on Recommender Systems (RecSys '09)*, New York, NY, USA, October 2009, ACM, New York, NY, USA, 261-264.

SHANI, G. AND GUNAWARDANA, A. 2011. Evaluating recommender systems. In *Recommender Systems Handbook*, chapter 8, F., RICCI, L., ROKACH, B., SHAPIRA, P.B., KANTOR, Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 257-298.

SNEDECOR, G.W. AND COCHRAN, W.G. 1980. *Statistical Methods*, 7th edition. Iowa State Press.

SU, X. AND KHOSHGOFTAAR, T.M. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence* 2009, 1-19.

WALTER, F.E., BATTISTON, S., AND SCHWEITZER, F. 2009. Personalised and dynamic trust in social networks. In *Proceedings of the 3rd ACM conference on Recommender systems (RecSys '09)*, New York, NY, USA, October 2009, ACM, New York, NY, USA, 197-204.

WENG, J., MIAO, C., AND GOH, A. 2006. Improving collaborative filtering with trust-based metrics. In *Proceedings of the 21st ACM symposium on Applied computing (SAC '06)*, Dijon, France, April 2006, ACM, New York, NY, USA, 1860-1864.

YOM-TOV, E., FINE, S., CARMEL, D., AND DARLOW, A. 2005. Learning to estimate query difficulty: including application to missing content detection and sitributed information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '05)*, Salvador, Brazil, August 2005, ACM, New York, NY, USA, 512-519.

ZAR, J.H. 1972. Significance Testing of the Spearman's Rank Correlation Coefficient. *Journal of the American Statistical Association* 67, 339, 578-580.

ZHOU, Y., AND CROFT, B.W. 2006. Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06)*, Arlington, VA, USA, November 2006, ACM, New York, NY, USA, 567-574.

ZIEGLER, C.N. AND LAUSEN, G. 2004. Analyzing correlation between trust and user similarity in online communities. In *Proceedings of the 2nd International Conference on Trust Management (iTrust '04)*, Oxford, UK, March 2004, volume 2995 of Lecture Notes in Computer Science, JENSEN, C., POSLAD, S., AND DIMITRAKOS, T., Eds. Springer Berlin / Heidelberg, Berlin, Heidelberg, 251-265.

Received March 2012; revised May 2013; revised November 2013; …