A Performance Prediction Approach to Enhance Collaborative Filtering Performance

Alejandro Bellogín and Pablo Castells

Universidad Autónoma de Madrid
Escuela Politécnica Superior
Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{alejandro.bellogin,pablo.castells}@uam.es

Abstract. Performance prediction has gained increasing attention in the IR field since the half of the past decade and has become an established research topic in the field. The present work restates the problem in the area of Collaborative Filtering (CF), where it has barely been researched so far. We investigate the adaptation of clarity-based query performance predictors to predict neighbor performance in CF. A predictor is proposed and introduced in a kNN CF algorithm to produce a dynamic variant where neighbor ratings are weighted based on their predicted performance. The properties of the predictor are empirically studied by, first, checking the correlation of the predictor output with a proposed measure of neighbor performance. Then, the performance of the dynamic kNN variant is examined on different sparsity and neighborhood size conditions, where the variant consistently outperforms the baseline algorithm, with increasing difference on small neighborhoods.

Keywords: recommender systems, collaborative filtering, neighbor selection, performance prediction, query clarity.

1 Introduction

Collaborative Filtering (CF) is a particularly successful form of personalized Information Retrieval, or personalized assistance over item choice problems in general [12,19]. CF has the interesting property that no item description is needed to recommend them, but only information about past interaction between users and items. Besides, it has the salient advantage that users benefit from other users' experience (opinions, votes, ratings, purchases, tastes, etc.), and not only their own, whereby opportunities for users' exposure to novel and unknown experiences with respect to previous instances are furthered, in contrast to other approaches that tend to reproduce the user's past, insofar as they examine the records of individual users in isolation.

CF is also based on the principle that the records of a user are not equally useful to all other users as input to produce recommendations [12]. A central aspect of CF algorithms is thus to determine which users form the best basis, and to what degree, to generate a recommendation for a particular user. Such users are usually referred to as *neighbors*, and their identification is commonly based on notions of similarity to the

target user. The similarity of two users is generally assessed by examining to what degree they displayed similar behaviors (selection, rating, purchase, etc.) in their interaction with items in the retrieval space. This basic approach can be complemented with alternative comparisons of virtually any user features the system may have access to, such as personal information, demographic data, or similar behaviors in external systems. Thus, the more similar a neighbor is to the active user, the more his tastes are taken into account as good advice to make up recommendations. For instance, a common CF approach consists of predicting the utility of an item for the target user by a weighted average of the ratings of all his neighbors, where the ratings are weighted by the similarity between each neighbor and the user. It is also common to set a maximum number of most similar users to restrict the set of neighbors to the *k* nearest, in order to avoid the noisy disruption of long tails of dissimilar users in the recommendation.

Similarity has indeed proved to be a key element for neighbor selection in order to provide accurate recommendations. Neighbor trustworthiness and expertise have also been researched as relevant complementary criteria to select the best possible collaborative advice [14, 19]. We believe however that further neighbor and data characteristics (individual or relative to the target user) can be exploited to enhance the selection and weighting of neighbors in recommendations. For instance, the size, heterogeneity, and other characteristics of the associated evidence (set of common known items, ratings, etc.), can be key to assess the significance of observations, the reliability of the evidence and the confidence of predictions, and the part of such elements in recommendations could be adjusted accordingly. Observations on users with little experience in common (where two or three coincidences of mismatches may lead to extreme similarity values) is far from being as significant as that on other users with a large subset of comparable history, and this difference should be accounted for in the CF algorithm. This type of issue is often mentioned and occasionally dealt with in the CF literature, but usually by hand-crafted solutions and manual tuning, rather than principled ways [4,12].

In this context, we research into notions of *neighbor goodness*, when seen as input for recommendation to a given user, where "goodness" should account for any aspect, besides similarity, that correlates with better results when the neighbor is introduced (or boosted) in computing a recommendation. Our proposed approach investigates the adaptation of performance prediction techniques developed in the IR field to assess neighbor goodness, where the latter is seen as an issue of *neighbor performance*. Specifically, we propose a neighbor goodness predictor inspired on query clarity. We analyze its correlation with an objective neighbor performance metric, and further measure the usefulness of the predictor by using it in a dynamic enhancement of a user-based *k* nearest neighbors (kNN) CF formulation, where neighbor ratings are weighted by their neighbor goodness. We show empiric evidence confirming that measurable improvements result from this approach.

The rest of the paper is organized as follows. Section 2 provides an overview of the state of the art in performance prediction in IR. In Section 3, the proposed approach is described, including the definition of the predictors and the formulation of rating prediction in CF as an aggregation operation with dynamic weights. Section 4 reports on the experimental work, where the proposed techniques are evaluated on a public dataset. Finally, Section 5 provides conclusions drawn from this work, along with potential lines for the continuation of the research.

2 Performance Prediction in Information Retrieval

Performance prediction in IR has been mostly addressed as a query performance issue, which refers to the performance of an IR system in response to a specific query. It also relates to the appropriateness of a query as an expression for a user information need. Dealing effectively with poorly-performing queries is a crucial issue in IR, and performance prediction provides tools that can be useful in many ways [22,23]. From the user perspective, it provides valuable feedback that can be used to direct a search, e.g. by rephrasing the query or providing relevance feedback. From the perspective of an IR system, performance prediction provides a means to address the problem of retrieval consistency: a retrieval system can invoke alternative retrieval strategies for different queries according to their expected performance (query expansion or different ranking functions based on the predicted difficulty). From the perspective of a system administrator, she can identify queries related to a specific subject that are difficult for the search engine, and e.g. expand the collection of documents to better answer insufficiently covered subjects. For distributed IR, performance estimations can be used to decide which search engine and/or database to use for each particular query, or how much weight to give it when its results are combined with those of other engines.

The prediction methods documented in the literature use a variety of available data as a basis for prediction, such as a query, its properties with respect to the retrieval space [7], the output of the retrieval system [5], or the output of other systems [3]. According to whether or not the retrieval results are used in the prediction, the methods can be classified into pre- and post-retrieval approaches [10]. The first type has the advantage that the prediction can be taken into account to improve the retrieval process itself. However, these predictors have the potential handicap, with regards to their accuracy, that the extra retrieval effectiveness cues available after the system response are not exploited [24]. In post-retrieval prediction, predictors make use of retrieved results [2,23,24]. Broadly speaking, techniques in this category provide better prediction accuracy. However, computational efficiency is usually a problem for many of these techniques, and furthermore, the predictions cannot be used to improve the retrieval strategies, unless some kind of iteration is applied, as the output from the retrieval system is needed to compute the predictions in the first place.

Pre-retrieval query performance has been studied mainly based on statistic methods, though linguistic approaches have also been researched [17]. Simple statistic approaches based on IDF, and variations thereof, have been proposed [11,13,18], showing moderate correlation with query performance though. He & Ounis propose the notion of query scope as a measure of the specificity of a query, which is quantified as the percentage of documents in the collection that contain at least one query term [11]. Query scope is effective in predicting the performance of short queries, though it seems very sensitive to query length [16].

More effective predictors have been defined on formal probabilistic grounds based on language models by the so-called *clarity score*, which captures the (lack of) ambiguity in a query with respect to the collection, or a specific result set [7,23,24] (the second case thus falling in the category of post-retrieval prediction). In this work, query ambiguity is meant to be "the degree to which the query retrieves documents in the given collection with similar word usage" [6]. Query clarity measures the degree of dissimilarity between the language associated with the query and the generic lan-

guage of the collection as a whole. This is measured as the relative entropy, or Kullback-Leibler divergence, between the query and collection language models (with unigram distributions).

Analyzing the entropy of the language model induced by the query is indeed a natural approach since entropy measures how strongly a distribution specifies certain values, in this case, terms. In its original formulation [7], query clarity is defined as follows:

$$\text{clarity}(q) = \sum_{w \in \mathcal{V}} \text{p}(w \mid q) \log_2 \frac{\text{p}(w \mid q)}{\text{p}_\text{c}(w)} \tag{1}$$

$$\text{p}(w \mid q) = \sum_{d \in R} \text{p}(w \mid d)\,\text{p}(d \mid q), \quad \text{p}(q \mid d) = \prod_{w_q \in q} \text{p}(w_q \mid d)$$

$$\text{p}(w \mid d) = \lambda\, \text{p}_\text{ml}(w \mid d) + (1-\lambda)\, \text{p}_\text{c}(w)$$

with w being any term, q the query, d a document or its model, R the set of documents in the collection that contain at least one query term (it is also possible to take the whole collection here), $\text{p}_\text{ml}(w|d)$ the relative frequency of term w in document d, $\text{p}_\text{c}(w)$ the relative frequency of the term in the collection as a whole, λ a free parameter (set to 0.6 in [7]), and $\mathcal{V}$ the entire vocabulary.

It was observed that queries whose likely relevant documents are a mix of disparate topics receive a lower score than those with a topically-coherent result set. A strong correlation was also found between query clarity and the performance of the result set. Because of that, the clarity score method has been widely used for query performance prediction in the area. Some applications include query expansion (anticipating poorly performing queries as good candidates to be expanded), rank fusion, link extraction in topic detection and tracking [15], and document segmentation [8]. A prolific sequel of variants and enhancements on the notion of clarity followed the original works [8,11].

3 Neighbor Performance in Collaborative Filtering

Starting from the work on performance prediction in IR, our research addresses the enhancement of neighbor selection techniques in CF by introducing the notion of neighbor performance, as an additional factor (besides similarity) to automatically tune the neighbor's participation in the recommendations, according to the expected goodness of their advice.

Our approach investigates the adaptation of the query clarity technique from IR to CF, as a basis for finding suitable predictors. This involves finding a meaningful equivalence or translation of the retrieval spaces involved in ad-hoc IR (queries, words, documents) into the corresponding elements of a CF setting (users, items, ratings), in order to provide a specific formulation. Moreover, in order to validate any proposed predictor, we should consider a measurable definition of what neighbor performance means, in order to check the correlation between predicted outcomes and objective measurements. We further test the effectiveness of the defined predictors by introducing and testing a dynamic variant of memory-based, user-based CF, in which the weights of neighbors are dynamically adjusted based on their expected effectiveness.

3.1 Assessing Neighbor Performance

The purpose of predictors in the proposed approach is to assess how useful specific neighbors' ratings are as a basis for predicting ratings for the active user in the basic CF formula. A 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, there is not, to the best of our knowledge, an equivalent function for CF neighbors (let alone a standard one) in the literature. We therefore need to introduce some sound candidate metric.

The measure we propose, named *neighbor goodness* (NG, how "good a neighbor" a user is to her surroundings), is defined as the difference in performance of the recommender system when including vs. excluding the user (her ratings) from the dataset (the performance of an item could be analogously defined in item-based CF). For instance, based on the mean average error (MAE) standard metric, NG can be defined as follows:

$$\begin{aligned} \mathrm{NG}(u) &= \frac{1}{\left|\mathcal{R}_{\mathcal{U}-\{u\}}\right|} \sum_{v \in \mathcal{U}-\{u\}} \mathrm{CE}_{\mathcal{U}-\{u\}}(v) - \frac{1}{\left|\mathcal{R}_{\mathcal{U}-\{u\}}\right|} \sum_{v \in \mathcal{U}-\{u\}} \mathrm{CE}_{\mathcal{U}}(v) \\ &= \frac{1}{\left|\mathcal{R}_{\mathcal{U}-\{u\}}\right|} \sum_{v \in \mathcal{U}-\{u\}} \left[\mathrm{CE}_{\mathcal{U}-\{u\}}(v) - \mathrm{CE}_{\mathcal{U}}(v)\right] \end{aligned} \tag{2}$$

where $\mathcal{U}$ is the set of all users, $\mathcal{R}$ is the set of all user-item pairs in $\mathcal{U} \times \mathcal{I}$ with known ratings, $\mathcal{R}_{\mathcal{X}} = \{ (u,i) \in \mathcal{R} \mid u \in \mathcal{X} \}$ is the subset of $\mathcal{R}$ restricted to users in $\mathcal{X} \subset \mathcal{U}$, and $\mathrm{CE}_{\mathcal{X}}(v)$ is the cumulative error of the recommender system on user v considering only the ratings of users in $\mathcal{X} \subset \mathcal{U}$, that is: $\mathrm{CE}_{\mathcal{X}}(v) = \sum_{i \in \mathcal{I}, r(v,i) \neq \varnothing} \left|\tilde{r}_{\mathcal{X}}(v,i) - r(v,i)\right|$, $\tilde{r}_{\mathcal{X}}(v,i)$ denoting the rating predicted by the system when taking $\mathcal{X}$ as the CF user community.

Note that the first term $\frac{1}{\left|\mathcal{R}_{\mathcal{U}-\{u\}}\right|} \sum_{v \in \mathcal{U}-\{u\}} \mathrm{CE}_{\mathcal{U}-\{u\}}(v)$ in equation (2) is just the MAE of the system when leaving out user u. The second term $\frac{1}{\left|\mathcal{R}_{\mathcal{U}-\{u\}}\right|} \sum_{v \in \mathcal{U}-\{u\}} \mathrm{CE}_{\mathcal{U}}(v)$ includes u in the computation of the recommendations of which the errors are measured and summed, but excludes the error on u itself from the sum, since we mean to measure strictly the effect of u on its neighbors, and not the reverse.

This measure thus quantifies how much a user affects (contributes to or detracts from) the total amount of MAE of the system, since it is computed in the same way as MAE, 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 is the world, in the sense of how well recommendations work, with and without the user.

3.2 Predicting Good Neighbors

Now, inspired by the clarity score defined for query performance [7], we consider its adaptation to predict neighbor performance in collaborative recommendation. 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. Cronen-Townsend et al showed that clarity is correlated with performance, because the less ambiguous a query, the more chances are that the system will return a good result in response [7]. Cronen-Townsend's experiments thus seem to confirm the underlying hypothesis that the system performance is largely influenced by the amount of uncertainty involved in the inputs it takes to build the retrieval result. That is, the uncertainty should correlate negatively with the performance level one may a priori expect.

CF systems rank and recommend items without an explicit user query. However, the system uses other inputs that may also determine the resulting performance. In analogy to the work on query clarity, we may hypothesize that the amount of uncertainty involved in a user neighbor may be a good predictor of his performance. In this case, the uncertainty can be understood as the ambiguity of the user's tastes, and it can be approximated as an adaptation of equation (1) to compute the clarity of users.

There are many possible ways to map the terms in equation (1) to elements of CF in meaningful ways, many of which we have studied before reaching the formulation proposed herein, which goes as follows. First, whereas the clarity measure follows a language modeling approach where three probability spaces are involved: queries, documents, and words, we map and fold this triadic approach into a dyadic one, involving only a set of users and a set of items. We have tested alternative triadic approaches, such as considering sets of features as the equivalent of document words, but they yield lower empiric performance, likely because the relation between a query and its constituent words, which is structural (i.e. a query is modeled as equivalent to the conjunction of its terms), does not map well to the relation between users and features (or even items and features), which is considerably looser in our experimental domain (based on MovieLens[1] and IMDb[2]).

In the dyadic approach, we have investigated two possible models, one in which the clarity of a user is measured against the set of all items, and one in which it is defined in terms of the set of all users. We shall follow here the latter option, which has shown a similar but slightly more consistent behavior than the former in our experiments:

$$\text{clarity}(u) = \sum_{v \in \mathcal{U}} \text{p}(v \mid u) \log_2 \frac{\text{p}(v \mid u)}{\text{p}_\text{c}(v)}$$

The conditional probability between users in the above formula can be rewritten in terms of conditional probabilities involving users and items:

$$\text{p}(v \mid u) = \sum_{i \in \mathcal{I}, r(u,i) \neq \varnothing} \text{p}(v \mid i) \text{p}(i \mid u)$$

[1] http://www.grouplens.org/node/73
[2] http://www.imdb.com

Now p($v|i$) and p($i|u$) can be computed by linearly smoothing probability estimates from observed evidence, as follows:

$$\mathrm{p}(v \mid i) = \lambda_1 \, \mathrm{p}_{\mathrm{ml}}(v \mid i) + (1-\lambda_1)\,\mathrm{p}_{\mathrm{c}}(v)$$
$$\mathrm{p}(i \mid u) = \lambda_2 \, \mathrm{p}_{\mathrm{ml}}(i \mid u) + (1-\lambda_2)\,\mathrm{p}_{\mathrm{c}}(i)$$

where we assume a uniform distribution for p_{c}, and we estimate p_{ml} based on rating data:

$$\mathrm{p}_{\mathrm{ml}}(v \mid i) = \frac{r(v,i)}{\sum_{u \in \mathcal{U}} r(u,i)}, \quad \mathrm{p}_{\mathrm{c}}(v) = \frac{1}{|\mathcal{U}|}$$
$$\mathrm{p}_{\mathrm{ml}}(i \mid u) = \frac{r(u,i)}{\sum_{j \in \mathcal{I}} r(u,j)}, \quad \mathrm{p}_{\mathrm{c}}(i) = \frac{1}{|\mathcal{I}|}$$

The same 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 tastes. Analogously, item clarities could be defined with respect to the space of users or the space of items in item-oriented CF, but we shall focus here only on the user-oriented approach. Having thus defined the notion of user clarity, the question is whether it can serve as a good neighbor performance predictor, and as such, whether its predictive power can be leveraged to dynamically weight the contribution of neighbors in CF in a way that improves the quality of recommendations. We address this in the next section.

3.2 Rating Prediction as a Dynamic Aggregation of Utilities

The same as performance prediction in IR has been used to optimize rank aggregation, in our proposed view each user's neighbor is seen as a retrieval subsystem (or criteria) whose output is to be combined to form the final system output (the recommendations) to the user.

A common utility-based formulation for rating prediction in memory-based CF, in a user-based, mean-centered variant [1], can be expressed as:

$$\tilde{r}(u,i) = \bar{r}(u) + C \sum_{v \in \mathrm{N}[u]} \mathrm{sim}(u,v) \cdot \left(r(v,i) - \bar{r}(v)\right), \qquad C = \frac{1}{\sum_{v \in \mathrm{N}[u]} \left|\mathrm{sim}(u,v)\right|} \tag{3}$$

where N[u] is the set of neighbors of the active user, $\bar{r}(u)$ is the average of all ratings by user u, and C is a normalizing constant to keep the rating values within scale. Note that this particular formulation of memory-based CF is chosen here without loss of generality, as our approach can be developed in equivalent terms for alternative CF variants (not mean-centered, item-based, etc. [1]).

The term $\tilde{r}(u,i)$ in equation (3) can be seen as a retrieval function that aggregates the output of several utility subfunctions $r(v,i) - \bar{r}(v)$, each corresponding to a recommendation given by a neighbor of the target 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) (scaled by C) to the target user. The computation of utility values in CF can thus be viewed as a case of rank aggregation in IR, and as such, a case for the enhancement of the aggregated result 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 quite an effective approach. The question is whether other performance factors, beyond similarity can be considered in a way that further enhancements can be drawn.

We thus aim to investigate whether CF results can be further enhanced by introducing, in addition to a similarity function, further effectiveness predictors, such as the user clarity value defined in the previous section, into the weights of the linear combination of neighbor ratings. The idea can be expressed as rewriting equation (3) as:

$$\tilde{r}(u,i) = \bar{r}(u) + C \sum_{v \in N[u]} \gamma(v,u,i) \cdot \mathrm{sim}(u,v) \cdot \left(r(v,i) - \bar{r}(v)\right)$$

where $\gamma(v,u,i)$ is a predictor of the performance of neighbor v.

In the general case, γ can be sensitive to the specific target user u, the item i, and in general it could even take further inputs from the recommendation space and context. As a first step, we explore the simple case when the predictor only examines the data related to the neighbor user v, and in particular, we consider $\gamma(v,u,i)$ = clarity(v). In the next section we show the experiments we have set up in order to observe the effect of the introduction of this predictor in the computation of collaborative recommendations.

4 Experimental Work

The experiments reported here have been carried out using the MovieLens dataset, and more specifically the so-called "100K" set. The main variable with respect to which the behavior of the proposed method is tested is the amount of sparsity, which we relate to the number of available ratings in the dataset based on which recommendations are computed. To this purpose, we split the dataset into different training / test cuts (10% to 90% in increments of 10%), with ten random splits per sparsity level. The neighborhood size is another parameter with respect to which the results are examined.

We first check the direct correlation between the user clarity predictor proposed in section 3.2 and the NG performance metric defined in 3.1, computed with a standard CF algorithm using the Taste library[3]. NG quantifies how a user affects the total amount of MAE, so that a well performing user should relate to high values of this measure (and vice-versa), reflecting to what degree the whole community gets better (or worse) results when the user is included as a potential neighbor. In the computation of clarity values, the λ_1 and λ_2 parameters were set to 0.6, as in [7].

The values shown in Table 1 show a positive direct correlation, meaning that the higher the value of the measure (well performing user), the higher the value of the predictor (clear user), thus confirming the predictive power of clarity for neighbor perfor-

[3] http://lucene.apache.org/mahout/taste.html

mance. An exception to this is when only 10% of ratings are used, where the correlation appears as negative. This lower end value results from data splits in which users have about ten ratings each on average in the training set, which seems insufficient to draw reliable predictions from. The correlation by the Spearman and Kendall functions yields similar results. While being indicative of a positive trend, and not far from previous results in query performance [5], observed correlation values still leave room for further elaboration and refinements of the proposed predictor and alternative ones, as well as the NG metric itself, in order to match the best findings in query performance [7].

Table 1. Pearson correlation values between user clarity and the NG performance metric at nine training/test split levels on the MovieLens rating dataset. The percentages indicate the ratio of rating data used for training in the CF algorithm.

% training	10%	20%	30%	40%	50%	60%	70%	80%	90%
correlation	-0.10	0.10	0.18	0.18	0.18	0.17	0.17	0.15	0.15

The second experiment consists of measuring final performance improvements when dynamic weights are introduced in a user-based CF. That is, the dynamic aggregation of neighbor ratings based on a prediction of their performance, when seen as individual recommenders (as defined in section 3.3), is tested against the basic CF algorithm without dynamic weights. Again, we have used the Taste implementation of user-based heuristic CF, both as a baseline, and as the algorithm into which the clarity-based enhancement is introduced.

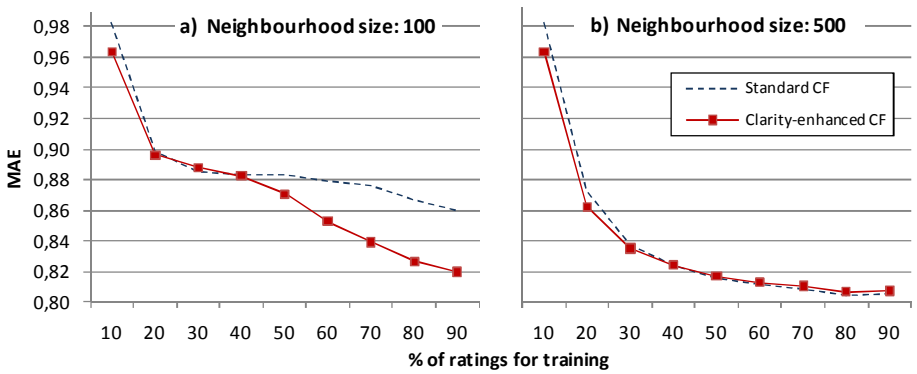

Fig. 1. Performance comparison of CF with clarity-based neighbor weighting, and standard CF, using neighborhoods of a) 100 users and b) 500 users.

Figure 1 shows the results for the clarity predictor, when taking neighborhood sizes of 100 and 500 users respectively. Each graphic shows performance values (MAE) for the nine splits described above. Our method clearly improves the baseline (by up to 5% for 60-80% cuts) when smaller neighborhoods (100 users) are used, and gets almost equal performance with neighborhoods of size 500 users. This shows that our method works particularly well when limited neighborhoods are used, and the improvement fades down to the baseline as they are enlarged. This means that our me-

thod is more efficient than the static option with respect to this variable, i.e. that it is able to get better results out of more economic neighborhood sizes.

Enlarging neighborhoods comes at an important computational cost in a CF system. Computational cost being one of the well-known problems in the field [9], achieving equal (or improved) performance at a lower cost is a relevant result. Let us recall that the total number of users in this dataset is 943, which means that 100 users is about a 10% of the total user community. CF systems described in the literature commonly take neighborhood sizes of 5 to 500 users for this dataset, 50 to 200 being the most common range [19,21].

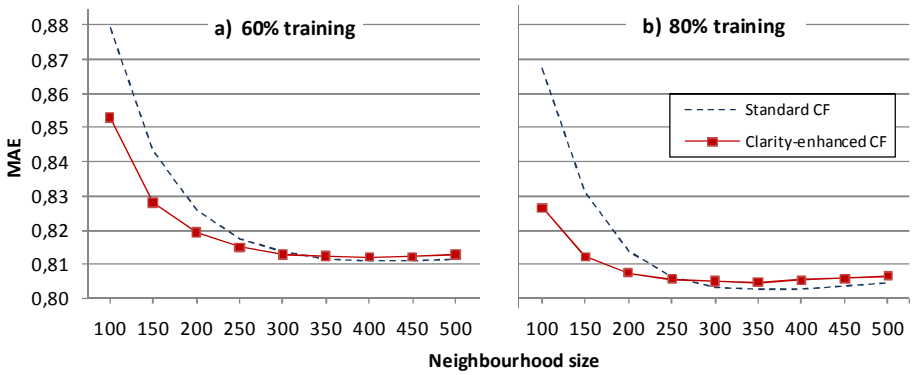

Fig. 2. Comparison of standard CF with dynamic, clarity-based neighbor weighting, and standard CF, using neighborhoods varying from 100 to 500 users, at a) 60% cut, b) 80% cut.

The trend in the evolution of performance with neighborhood size is clear in Figure 2, showing the effect of the clarity-based enhancement at different sizes, setting the sparsity cut at a) 60% and b) as a double check, 80% (which are standard ranges in the CF literature). It can be seen that the shape of the curves in both figures is very similar, evidencing the consistent superiority of clarity-enhanced CF with small to medium (i.e. usual) neighborhood sizes (e.g. over 5% improvement at size = 100 users with 80% training data).

5 Conclusions

Our work explores the use of performance prediction techniques to enhance the selection and weighting of neighbors in CF. The proposed approach consists of the adaptation of performance predictors originally defined for ad-hoc retrieval, into the CF domain, where users and items (and ratings), instead of documents and queries, make up the problem space. A predictor is proposed and used to introduce dynamic weights in the combination of neighbor ratings in the computation of collaborative recommendations, in an approach where the better the expected performance of a neighbor is, the higher weight is assigned to her ratings in the combination. The reported experimental results show performance improvements as a result of this dynamic weights

adjustment approach, which supports the predictive power of clarity-based techniques in CF as a basis for this kind of adjustment. The results are particularly positive in small neighborhood situations.

Future work includes the exploration of alternative variants of the clarity-based predictor, as well as new predictors based on other techniques which have achieved good results in IR. We also aim to research neighbor selection methods based on external information sources, such as social network data. Our research so far has focused on the user-based kNN approach to CF, as it is particularly intuitive for the formulation of the researched problem, and lends itself well to exploit user qualities implicit in the data, or obtainable from external sources, linking to interesting problems in adjacent areas (e.g. social dynamics). We plan nonetheless to study the proposed approach under alternative baseline CF formulations, such as item-based kNN and factor models.

Beyond the current research presented here, recommender systems, and personalized IR at large, are particularly propitious areas for the introduction of performance prediction techniques, because of the naturally arising need for combination of multiple diverse evidence and strategies, and the uncertainty (and thus the variable accuracy) involved in the exploitation of implicit evidence of user interests. For instance hybrid recommender systems combine a content-based and a collaborative approach. Performance predictors could be researched to weight the influence of each component in the final recommendations (e.g. CF is sensitive to gray sheep or new item situations, while content-based filtering is not). Personalized ah-hoc retrieval is another interesting problem for this approach, where the weight of a query vs. implicit evidence from user history can be dynamically adjusted depending on the predicted effectiveness of each side. To the best of our knowledge, the introduction of performance predictors in these areas has been barely addressed, if at all, as a formal problem.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02) and the Ministry of Industry, Tourism and Commerce (CENIT-2007-1012).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734--749 (2005)
2. Amati, G., Carpineto, C., Romano, G.: Query difficulty, robustness, and selective application of query expansion. In: McDonald, S., Tait, J. (eds.) Advances in Information Retrieval, LNCS, vol. 2997, pp. 127--137. Springer, Heidelberg (2004)
3. Aslam, J.A., Pavlu, V.: Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In: Amati, G., Carpineto, C., Romano, G. (eds.) Advances in Information Retrieval, LNCS, vol. 4425, pp. 198--209. Springer, Heidelberg (2007)
4. Baltrunas, L., Ricci, F.: Locally adaptive neighborhood selection for collaborative filtering recommendations. In: Nejdl, W., Kay, J., Pu, P., Herder, E. (eds.) Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS, vol. 5149, pp. 22--31. Springer, Heidelberg (2008)
5. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006), pp. 390--397. ACM Press, New York, (2006)

6. Cronen-Townsend, S., Zhou, Y., Croft, W.: Precision prediction based on ranked list coherence. Information Retrieval 9(6), 723--755 (2006)
7. Cronen-Townsend, S., Zhou, Y., Croft, B.W.: Predicting query performance. In: 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2002), pp. 299--306. ACM Press, New York (2002)
8. Diaz, F., Jones, R.: Using temporal profiles of queries for precision prediction. In: 27th annual international conference on Research and development in information retrieval (SIGIR 2004), pp. 18--24. ACM Press, New York (2004)
9. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: A constant time collaborative filtering algorithm. Information Retrieval 4(2), 133--151 (2001)
10. Hauff, C., Azzopardi, L., and Hiemstra, D.: The combination and evaluation of query performance prediction methods. In: Advances in Information Retrieval, LNCS, vol. 5478, pp. 301--312. Springer, Heidelberg (2009)
11. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In: Apostolico, A., Melucci, M. (eds.) String Processing and Information Retrieval, LNCS, vol. 2346, pp. 43--54. Springer, Heidelberg (2004)
12. Herlocker, J., Konstan, J.A., Riedl, J.: An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. Information Retrieval 5(4), 287--310 (2002)
13. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11--20 (1972)
14. Kwon, K., Cho, J., Park, Y.: Multidimensional credibility model for neighbor selection in collaborative recommendation. Expert Systems with Applications 36(3), 7114--7122 (2009)
15. Lavrenko, V., Allan, J., Deguzman, E., Laflamme, D., Pollard, V., and Thomas, S. Relevance models for topic detection and tracking. In: 2nd int. conference on Human Language Technology Research, pp. 115--121. Morgan Kaufmann Publishers, San Francisco (2002)
16. Macdonald, C., He, B., Ounis, I.: Predicting query performance in intranet search. In: ACM SIGIR Workshop on Predicting Query Difficulty – Methods and Applications (2005)
17. Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty. In: ACM SIGIR Workshop on Predicting Query Difficulty – Methods and Applications. Salvador, Brazil (2005)
18. Plachouras, V., He, B., Ounis, I.: University of Glasgow at TREC2004: Experiments in Web, Robust and Terabyte tracks with Terrier. In: 13th Text Retrieval Conference (TREC 2004). Gaithesburg, Maryland (2004)
19. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: 2005 International Conference on Intelligent User Interfaces (IUI), pp. 167--174. ACM Press, New York (2005)
20. Wang, J., de Vries, A.P., Reinders, M.J.T.: Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In: 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2006), pp. 501--508. ACM Press, New York (2006)
21. Xue, G.R., Lin, C., Yang, Q., Xi, W., Zeng, H.J., Yu, Y., Chen, Z.: Scalable collaborative filtering using cluster-based smoothing. In: 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005), pp. 114--121. ACM Press, New York (2005)
22. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005), pp. 512--519. ACM Press, New York, (2005)
23. Zhou, Y., Croft, B.W.: Ranking robustness: a novel framework to predict query performance. In: 15th ACM conference on Information and knowledge management (CIKM 2006), pp. 567--574. ACM Press, New York (2006)
24. Zhou, Y., Croft, B.W.: Query performance prediction in web search environments. In: 30th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2007), pp. 543--550, ACM Press, New York (2007)