Predicting the Performance of Recommender Systems: An Information Theoretic Approach

Alejandro Bellogín, Pablo Castells, and Iván Cantador

Universidad Autónoma de Madrid
Escuela Politécnica Superior, Departamento de Ingeniería Informática
Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{alejandro.bellogín, pablo.castells, ivan.cantador}@uam.es

Abstract. Performance prediction is an appealing problem in Recommender Systems, as it enables an array of strategies for deciding when to deliver or hold back recommendations based on their foreseen accuracy. The problem, however, has been barely addressed explicitly in the area. In this paper, we propose adaptations of query clarity techniques from ad-hoc Information Retrieval to define performance predictors in the context of Recommender Systems, which we refer to as user clarity. Our experiments show positive results with different user clarity models in terms of the correlation with single recommender's performance. Empiric results show significant dependency between this correlation and the recommendation method at hand, as well as competitive results in terms of average correlation.

Keywords: performance prediction, recommender systems, language models

1 Introduction

Performance prediction has gained increasing attention in Information Retrieval (IR) since the late 90's, and has become an established research topic in the field [6]. It has been mostly addressed as a query performance issue, which refers to the performance of an IR system in response to a specific query. Particularly effective predictors have been defined based on language models by the so-called clarity score, which captures the ambiguity in a query with respect to the collection, or a specific result set [6].

Performance prediction finds a special motivation in Recommender Systems (RS). Contrary to query-based retrieval, as far as the initiative relies on the system, it may decide to produce recommendations or hold them back, depending on the expected level of performance on a per case basis, delivering only the sufficiently reliable ones. The problem of performance prediction, however, has barely been addressed in RS to date. The issue is in fact tackled in the RS literature by ad hoc heuristic tweaks –evidencing the relevance of the problem–, but has not been studied and addressed in a principled way. Examples of such heuristic approaches are significance weighting [12] and confidence [18], where additional computations (mainly normalizations) are introduced in order to better estimate the final prediction ratings.

Performance prediction finds further motivation in RS, as the performance of individual recommendation methods is highly sensitive to different conditions, such as

data sparsity, quality, and reliability, which in real settings are subject to an ample dynamic variability. Hence, being able to estimate in advance which recommenders are likely to provide the best output in a particular situation opens up an important window for performance enhancement. Alternatively, estimating which users in the system are likely to receive worse recommendations allows for modifications in the recommendation algorithms to predict this situation, and react in advance.

In the research presented here, we consider the adaptation –and area-specific elaborations thereupon– to RS of principles that have been proposed and developed in ad-hoc IR. In particular, the approaches based on Information Theory principles and measures, as developed in the query clarity models, have shown to be useful in many ways to deal effectively with poorly-performing queries [19]. We propose different vocabulary spaces where clarity definition may be applied to, in order to better capture the ambiguity in user preferences. Moreover, we define alternative statistical models and estimating approaches, under different independence assumptions. In conducted experiments, we have obtained similar correlation values to those of state-of-the-art predictors in terms of average correlation. We also find significant differences in correlation between different recommenders and the same predictor.

2 Performance Prediction in Information Retrieval

Query performance prediction in IR 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. In the literature, prediction methods have been classified into two groups depending on the available data used for prediction [9]: pre-retrieval approaches, which make the prediction before the retrieval stage, and post-retrieval approaches, which use the rankings produced by the retrieval engine.

Pre-retrieval approaches have the advantage that the prediction can be taken into account to improve the retrieval process itself. These predictors, however, have the potential handicap, with regards to their accuracy, that the extra retrieval effectiveness cues available after the system response are not exploited [19]. Query scope [11] is an example of this type of predictors. It is 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. Other examples such as statistic approaches based on Inverse Document Frequency (IDF), and variations thereof, have also been proposed [11, 16]. He & Ounis [11] propose a predictor based on the standard deviation of the IDF of the query terms. Plachouras et al. [16] represent the quality of a query term by a modification of IDF, where instead of the number of documents, the number of words in the whole collection is used, and the query length acts as a normalizing factor. These IDF-based predictors obtained moderate correlation with respect the query performance. Linguistic approaches have also been investigated [14].

Secondly, post-retrieval predictors make use of retrieved results. Broadly speaking, techniques in this category provide better prediction accuracy [2, 19]. 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 com-

pute the predictions in the first place. Most effective predictors have been defined based on language models by the so-called clarity score, which captures the (lack of) ambiguity in a query with respect to a specific result set, or the whole collection [6, 19] (the second case thus can be considered as a pre-retrieval predictor, since it does not make use of the result set). Besides query clarity, other post-retrieval predictors have been defined based on the differences in ranking between the original input and after query or document perturbation (see [9] for a summary of these methods).

In this work, we focus on the clarity score predictor, which is measured as the Kullback-Leibler divergence, and estimates the coherence of a collection with respect to a query q in the following way, given the vocabulary $\mathcal{V}$ and a subset of the document collection R:

$$\text{clarity}(q) = \sum_{w \in \mathcal{V}} p(w|q) \log_2 \frac{p(w|q)}{p_c(w)}$$

$$p(d|q) = p(q|d)p(d);\ p(q|d) = \prod_{w_q \in q} p(w_q|d)$$

$$p(w|q) = \sum_{d \in R} p(w|d)p(d|q)\,; p(w|d) = \lambda p_{\text{ml}}(w|d) + (1-\lambda)p_c(w)$$

The clarity value can be reduced, thus, to an estimation of the prior $p_c(w)$ and the posterior $p(w|q)$ of query terms w over documents d, based on term frequencies and smoothing. Cronen-Townsend et al [6] showed that clarity is correlated with performance, demonstrating that the result quality is largely influenced by the amount of uncertainty involved in the inputs the system takes. In this sense, queries whose likely documents are a mix of documents from disparate topics receive lower score than if they result in a topically-coherent retrieved set. Several works have exploited its functionality and predictive capabilities [5, 7, 8], supporting its effectiveness in terms of performance prediction and high degree of adaptation.

3 Predictive Models of Recommendation Performance

Predicting the performance of recommender systems requires the definition of the key element we want to predict the performance for. In this paper, we identify the user having the role of the query in an IR system, although an equivalent development could be made for items instead of users.

In the following, we define different user performance predictors, whose main goal is to infer how good or bad the system is expected to perform for a given user. We propose a fairly general adaptation of query clarity, which may be instantiated in different models, depending on the input spaces considered. Specifically, our adaptation of query clarity has the following formulation:

$$\text{clarity}(u) = \sum_{x \in X} p(x|u) \log_2 \frac{p(x|u)}{p(x)} \quad (1)$$

As we can observe, the clarity formulation strongly depends on a "vocabulary" space X, which further constrains the user-conditioned model (or user model for short) $p(x|u)$, and the background probability $p(x)$. In ad-hoc IR, this space is typically the space of words, and the query language model is a probability distribution over words [6]. In RS, however, we may have different interpretations, and thus, different formulations for such a probabilistic framework, as we shall show. In all cases, we will need to model and estimate two probability distributions: first, the probability that some event (depending on the current probability space X) is generated by the user language model (*user model*); and second, the probability of generating that event without any constraint (*background model*). In Table 1, we propose three different vocabulary spaces for X, along with the associated probabilistic models.

Table 1. Three possible user clarity formulations, depending on the interpretation of the vocabulary space.

User clarity	Vocabulary Space	User model	Background model
Rating-based	Ratings	$p(r\|u)$	$p_c(r)$
Item-based	Items	$p(i\|u)$	$p_c(i)$
Item-and-rating-based	Items rated by the user	$p(r\|i,u)$	$p_{ml}(r\|i)$

In all the above formulations, user clarity is in fact the difference (Kullback-Leibler divergence) between a user model and a background model. The use of user and background distributions as a basis to predict recommendation performance lies on the hypothesis that a user probability model being close to the background (or collection) model is a sign of ambiguity or vagueness in the evidence of user needs, since the generative probabilities for a particular user are difficult to singularize from the model of the collection as a whole. In IR, this fact is interpreted as a query whose ranked documents are a mix of articles about different topics [6].

As stated in [6], language models capture statistical aspects of the generation of language. Therefore, if we use different vocabularies, we may capture different aspects of the user. Specifically, for each of the vocabulary spaces defined in Table 1, we assume different user-specific interpretations. The rating-based clarity model captures how differently a user uses rating values (regardless of the items the values are assigned to) with respect to the rest of users in the community. The item-based clarity takes into account which items have been rated by a user, and therefore, whether she rates (regardless of the rating value) the most rated items in the system or not. Finally, the item-and-rating-based clarity computes how likely a user would rate each item with some particular rating value, and compares that likelihood with the probability that the item is rated with some particular rating value.

In this sense, the item-based user dependent model makes the assumption that some items are more likely to be generated for some users than for others depending on their previous preferences. The rating-based model, on the other hand, captures the likelihood of a particular rating value being assigned by a user, which is an event not as sparse as the previous one with a larger number of observations. Finally, the item-and-rating-based model is a combination of the previous models, by assuming unified models which incorporate items and ratings.

In the next section, we get into details on the formal definition of the u, i, and r random variables introduced in the above equations, along with the practical estimation of the involved distributions.

4 Ground Models

We ground the different clarity measures defined in the previous section upon a rating-oriented probabilistic model very similar to the approaches taken in [13] and [18]. The sample space for the model is the set $\mathcal{U}\times\mathcal{I}\times\mathcal{R}$, where $\mathcal{U}$ stands for the set of all users, $\mathcal{I}$ is the set of all items, and $\mathcal{R}$ is the set of all possible rating values. Hence an observation in this sample space consists of a user assigning a rating to an item. We consider three natural random variables in this space: the user, the item, and the rating value, involved in a rating assignment by a user to an item. This gives meaning to the distributions expressed in the different versions of clarity as defined in the previous section. For instance, $p(r|i)$ represents the probability that a specific item i is rated with a value r –by a random user–, $p(i)$ is the probability that an item is rated –with any value by any user–, and so on.

The probability distributions upon which the proposed clarity models are defined can use different estimation approaches, depending on the independence assumptions and the amount of involved information. Background models are estimated using relative frequency estimators, that is:

$$p_c(r) = \frac{|\{(u,i) \in \mathcal{U}\times\mathcal{I} \,|r(u,i) = r\}|}{|\{(u,i) \in \mathcal{U}\times\mathcal{I} \,|r(u,i) \neq \emptyset\}|};\; p_c(i) = \frac{|\{u \in \mathcal{U} \,|r(u,i) \neq \emptyset\}|}{|\{(u,j) \in \mathcal{U}\times\mathcal{I} \,|r(u,j) \neq \emptyset\}|}$$

$$p_{ml}(r|i) = \frac{|\{u \in \mathcal{U} \,|r(u,i) = r\}|}{|\{u \in \mathcal{U} \,|r(u,i) \neq \emptyset\}|};\; p_{ml}(r|u) = \frac{|\{i \in \mathcal{I} \,|r(u,i) = r\}|}{|\{i \in \mathcal{I} \,|r(u,i) \neq \emptyset\}|}$$

These are maximum likelihood estimations in agreement with the meaning of the random variables as defined above. Starting from these estimations, user models can be reduced to the above terms by means of different probabilistic expansions and reformulations, which we define next for each of the models introduced in the previous section.

Item based model. The $p(i|u)$ model can be simply expanded through ratings, but under two different assumptions: the item generated by the model only depends on the rating value, independently from the user or, in the contrary, depends on both the user and the rating). These alternatives lead to the following development, respectively:

$$p_R(i|u) = \sum_{r\in\mathcal{R}} p_{ml}(i|r)p_{ml}(r|u)$$

$$p_{UR}(i|u) = \sum_{r\in\mathcal{R}} p(i|u,r)p_{ml}(r|u)$$

Rating based model. This model assumes that the rating value generated by the probability model depends on both the user and the item at hand. For this model, we sum over all possible items in the following way:

$$p(r|u) = \sum_{r(u,i)=r} p(r|\mathrm{i},u)p(i|u)$$

where the $p(i|u)$ term can be developed as in the item-based model above. The term $p(r|\mathrm{i},u)$ requires further development, which we define in the next model.

Item-and-rating based model. Three different models can be derived depending on how the Bayes' rule is applied. In the same way as proposed in [18], three relevance models can be defined, namely a user-based, an item-based, and a unified relevance model:

$$p_U(r|i,u) = \frac{p(u|r,i)p_{ml}(r|i)}{\sum_{r\in\mathcal{R}} p(u|r,i)p_{ml}(r|i)}$$

$$p_I(r|i,u) = \frac{p(i|r,u)p_{ml}(r|u)}{\sum_{r\in\mathcal{R}} p(i|u,r)p_{ml}(r|u)}$$

$$p_{UI}(r|i,u) = \frac{p(u,i|r)p_{\mathrm{c}}(r)}{\sum_{r\in\mathcal{R}} p(u,i|r)p_{\mathrm{c}}(r)}$$

The first derivation induces a user-based relevance model because it measures by $p(u|r,i)$ how probable it is that a user rates item i with a value r. The item-based relevance model is factorized proportional to an item-based probability, i.e., $p_I(r|i,u) \propto p(i|r,u)$. Finally, in the unified relevance model, we have $p_{UI}(r|i,u) \propto p(u,i|r)$.

Different combinations of distribution formulations and estimations result in a fair array of alternatives. Among them, we focus on a subset that is shown in Table 2, which provide the most interesting combinations, in terms of experimental efficiency, of user and background distributions for each clarity model. These alternatives are further analyzed in detail in the next sections.

Table 2. Different user clarity models implemented

User clarity name	User dependent model	Background model
RatUser	$p_U(r\|i,u); p_{UR}(i\|u)$	$p_{\mathrm{c}}(r)$
RatItem	$p_I(r\|i,u); p_{UR}(i\|u)$	$p_{\mathrm{c}}(r)$
ItemSimple	$p_R(i\|u)$	$p_{\mathrm{c}}(i)$
ItemUser	$p_{UR}(i\|u)$	$p_{\mathrm{c}}(i)$
IRUser	$p_U(r\|i,u)$	$p_{ml}(r\|i)$
IRItem	$p_I(r\|i,u)$	$p_{ml}(r\|i)$
IRUserItem	$p_{UI}(r\|i,u)$	$p_{ml}(r\|i)$

5 Qualitative observation

In order to illustrate the proposed prediction framework and give an intuitive idea of what the user characteristics predictors are capturing, we show the relevant aspects of specific users that result in clearly different predictor values, in a similar way to the examples provided in [6] for query clarity. We compare three user clarity models out of the seven models presented in Table 2: one for each formulation included in Table 1. In order to avoid distracting biases on the clarity scores that a too different number of ratings between users might cause, we have selected pairs of users with a similar

number of ratings. This effect would be equivalent to that found in IR between the query length and its clarity for some datasets [9].

Table 3. Two example users, showing the number of ratings they have entered, and their performance prediction values for three user clarity models.

User	Number of ratings	ItemUser clarity	RatItem clarity	IRUserItem clarity
u_1	51	216.015	28.605	6.853
u_2	52	243.325	43.629	13.551

Table 3 shows the details of two sample users on which we will illustrate the effect of the predictors. As we may see in the table, u_2 has a higher clarity value than u_1 for the three models analyzed. That is, according to our theory, u_2 is less "ambiguous" than u_1.

Figure 1 shows the clarity contribution in a term-by-term basis for one of the item-and-rating-based clarity models –where, in this case, terms are equivalent to a pair (rating, item)– as done in [6]. In the figure, we plot $p(r|u,i)\log_2(p(r|u,i)/p(r|i))$ for the different terms in the collection, sorted in descending order of contribution to the user model, i.e., $p(r|u,i)$, for each user. For the sake of clarity, only the top 20 contributions are plotted. We may see how the user with the smaller clarity value receives lower contribution values than the other user. This observation is somewhat straightforward since the clarity value, as presented in equation 1, is simply the sum of all these contributions, over the set of terms conforming the vocabulary. In fact, the figures are analogous for the rest of the models, since one user always obtains higher clarity value than the other.

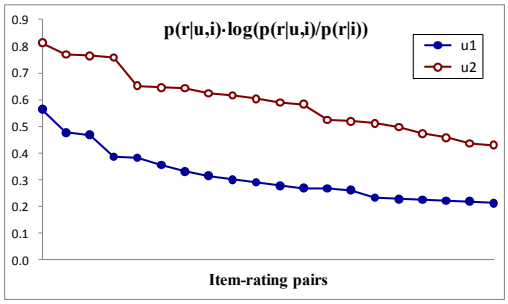

Fig. 1. Term contributions for each user, ordered by their corresponding contribution to the user language model. IRUserItem clarity model.

Let us now analyze more detailed aspects in the statistical behavior of the users that explain their difference in clarity. The IRUserItem clarity model captures how differently a user rates an item with respect to the community. Take for instance the top item-rating pairs for users 1 and 2 in the above graphic. The top pair for u_2 is (4, "McHale's Navy"). This means that the probability of u_2 rating this movie with 3 is much higher than the background probability (considering the whole user community) of this rating for this movie. Indeed, we may see that u_2 rated this movie with a 3, whereas the community mode rating is 1 –quite farther away from 4. This is the trend in a clear user. On the other extreme of the displayed values, the bottom term in the figure for user 1 is (2, "Donnie Brasco"), which is rated by this user with a 5, and the

community mode rating for this item is 4, thus showing a very similar trend between both. This is the characteristic trend of a non-clear user.

Furthermore, if we compare the background model with the user model, we obtain more insights about how our models are discriminating distinctive from mainstream behavior. This is depicted in Fig. 2. In this situation, we select those terms which maximize the difference between the user and background models. Then, for this subset of the terms, we sort the vocabulary with respect to its collection probability, and then we plot the user probability model for each of the terms in the vocabulary.

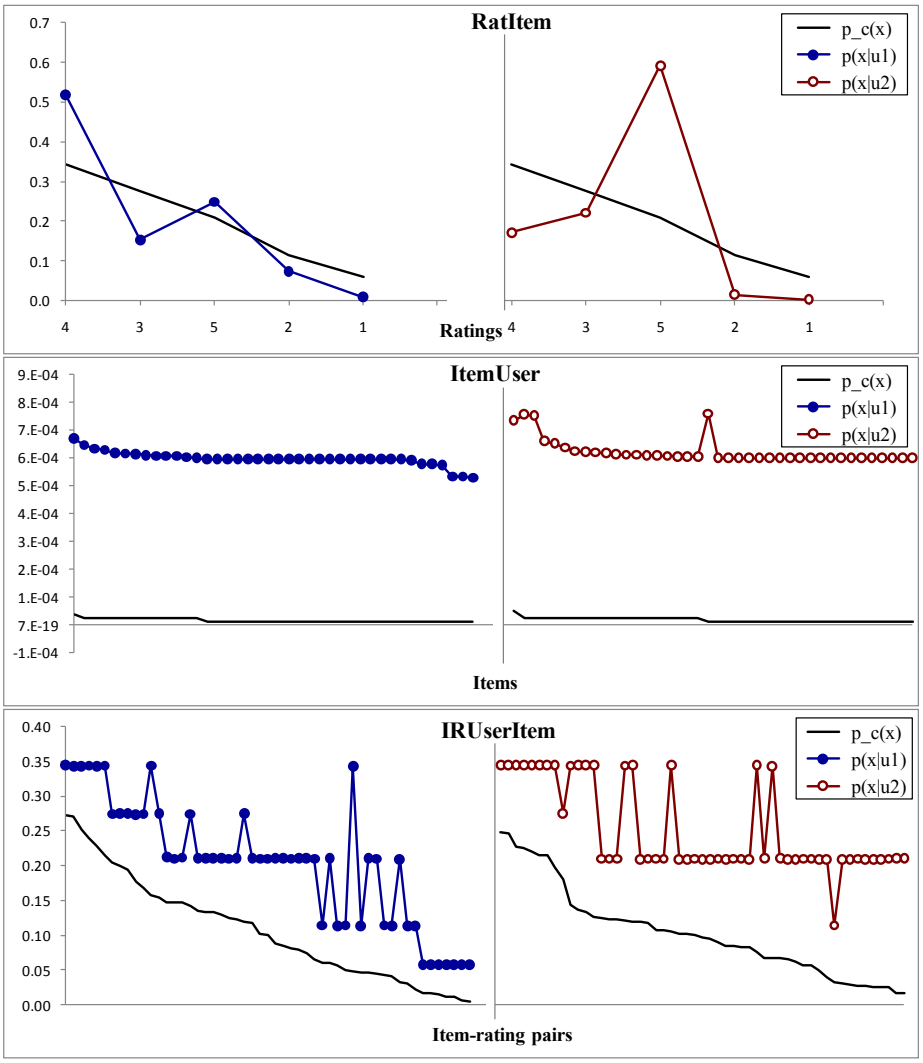

Fig. 2. User language model sorted by collection probability.

These figures show how the most ambiguous user obtains a similar distribution to that of the background model, while the distribution of the less ambiguous user is more different. In the rating-based model this effect is clear, since the likelihood of

not so popular rating values (i.e., a '5') is larger for user 2 than for user 1, and at the same time, the most popular rating value (a '4') is much more likely for user 1. The figure about the ItemUser model is less clear in this aspect, although two big spikes are observed for user 1 with respect to the collection distribution, which correspond with two strange movies: 'Waiting for Guffman' and 'Cry, the beloved country', both with a very low collection probability. Finally, the figure about the IRUserItem model successfully shows how user 2 has more spikes than user 1, indicating a clear divergence from the backgrund model; in fact, user 1's distribution partially mimics that of the collection. In summary, the different models proposed are able to successfully separate information concerning the user and that from the collection, in order to infer whether a user is different or similar from the collection as a whole.

Finally, it is worth noticing the relation between the clarity value and the performance metric. For instance, the value of nDCG@50 for user 1 is 0.288, and for user 2 is 0.371. In this situation, thus, the relation is linear, since performance values increase with clarity values. As we shall show in the next sections, this is coherent with the empirical correlation, which is, in median, between 0.25 and 0.50. This seems to indicate that users who follow mainstream trends are more difficult to be suggested successful items by a recommender system. In IR, one can observe a similar trend: more ambiguous (mixture of topics) queries perform worse than higher-coherence queries [6]. Note that this result might seem contradictory with the popular intuition of the *gray sheep* user who is difficult to get accurate recommendations because he lacks enough similarity with the rest of users. This trend may suggest a revision or perhaps just a more precise definition of what a gray sheep –as a performance challenging situation– really is.

6 Experiments

In this section, we study the correlation of the user performance predictors defined in previous sections and the performance of different recommenders. We use for this purpose the Movielens 100K[1] dataset, with a 5-fold cross validation of all tests. We test two state-of-the-art CF algorithms [1] (user-based with 50 neighbors, denoted as UB, and item-based, as IB) as implemented in the Mahout library[2]. We used two additional algorithms, recently developed, which obtain very good performance in terms of precision metrics, which we denote as TF-L1 and TF-L2 [4]. They implement an item-based CF approach with different normalization and weighting functions for the similarity or rating values. Finally, we implemented a content-based recommender (denoted as CBF) using movie genre, director, and country, from IMDb[3], as item attributes.

Table 4 shows the Pearson's correlation values between the predictors presented in previous sections, and the nDCG when only the top 50 items are considered (nDCG@50). We can observe fairly high correlation values for recommenders TF-L1 and TF-L2, comparable to results in the query performance literature. A slightly lower

[1] Available at http://www.grouplens.org/node/73
[2] Available at http://mahout.apache.org
[3] Internet Movie Database, http://www.imdb.com

correlation is found for UB, whereas an insignificant value is observed for CBF and IB. These results are consistent when other performance metrics are used such as MAP, and at different cutoff lengths. Spearman's correlation yields similar values.

Table 4. Pearson's correlation between predictors and nDCG@50 for different recommenders.

Predictor	CBF	IB	TF-L1	TF-L2	UB	**Median**	**Mean**
ItemSimple	0.257	0.146	0.521	0.564	0.491	0.491	0.396
ItemUser	0.252	0.188	0.534	0.531	0.483	0.483	0.398
RatUser	0.234	0.182	0.507	0.516	0.469	0.469	0.382
RatItem	0.191	0.184	0.442	0.426	0.395	0.395	0.328
IRUser	0.171	-0.092	0.253	0.399	0.257	0.253	0.198
IRItem	0.218	0.152	0.453	0.416	0.372	0.372	0.322
IRUserItem	0.265	0.105	0.523	0.545	0.444	0.444	0.376

The standard procedure in IR for this kind of evaluation is to compute correlations between the predictor(s) and one retrieval model (like in [6, 10]) or an average of several methods [14]. This approach may hide the correlation effect for some recommenders, as we may observe from the median and mean correlation values, which are still very large despite the fact that two of the recommenders analyzed have much lower correlations. These aggregated values, i.e., the mean and the median, provide competitive correlation values when compared with those in the literature.

We believe the difference in correlation for CBF and IB recommenders may be explained considering two factors: the actual recommender performance, and the input sources used by the recommender. With regards to the first factor, the IB algorithm performs poorly (in terms of the considered ranking quality metrics, such as nDCG and MAP) in comparison to the rest of recommenders. It seems natural that a good predictor for a well performing algorithm (specifically, TF-L2 is the best performing recommender in this context) would hardly correlate at the same time with a poorly performing one.

This does not explain however the somewhat lower correlation with the content-based recommender, which has better performance than UB. The input information that this recommender and the predictors take are very different: the latter compute probability distributions based on ratings given by users to items, while the former uses content features from items, such as directors and genres. Furthermore, the CBF recommender is not coherent with the inherent probabilistic models described by the predictors, since the events modeled by each of them are different: CBF would be related with the likelihood an item is described by the same features as those items preferred by the user, whereas predictors are related with the probability that an item is rated by a user. Moreover, the predictors' ground models coherently fit in the standard CF framework [18], which reinforces the suitability of the user performance predictors presented herein, at least for CF recommenders.

It is worth noting to this respect that most clarity-based query performance prediction methods in IR study their predictive power on language modeling retrieval systems [6, 10, 20] or similar approaches [11]. This suggests that a well performing predictor should be defined upon common spaces, models, and estimation techniques as the retrieval system the performance of which is meant to be predicted.

7 Conclusion

We have proposed adaptations of query clarity techniques from ad-hoc Information Retrieval to define performance predictors in Recommender Systems. Taking inspiration in the query performance predictor known as query clarity, we have defined and elaborated in the Recommender Systems domain several predictive models according to different formulations and assumptions.

We obtain strong correlation values confirming that our approach results in a high predictive power for recommender systems performance. As a side-effect, our study introduces an interesting revision of the gray sheep user concept. A simplistic interpretation of the gray sheep intuition would suggest that users with a too unusual behavior are a difficult target for recommendations. It appears however in our study that, on the contrary, users who somewhat distinguish themselves from the main trends in the community are easier to give well-performing recommendations. This suggests that perhaps the right characterization of a gray sheep user might be one who has scarce overlap with other users. On the other hand, the fact that a clear user distinguishes herself from the aggregate trends does not mean that she does not have a sufficiently strong neighborhood of similar users.

Besides the theoretic interest per se, we envision two potential applications for the proposed prediction techniques: dynamic neighbor weighting in collaborative filtering, and the dynamic adjustment of recommender ensembles. The first problem was already researched in [3], where a dynamic collaborative filtering algorithm outperformed the standard formulation by promoting neighbors that are expected to perform better in a nearest-neighbor recommendation algorithm. We are currently working on the second problem, namely, how to dynamically choose the best weights in a recommender ensemble. An additional application –somewhat obvious, albeit not less useful– is to use performance prediction to trigger recommendations only when the predicted performance is above some threshold, thus saving the user potential misses, plus the computational cost. We also plan to continue exploring further performance predictors. Specifically, we are interested in incorporating explicit recommender dependence into the predictors, so as to better exploit the information managed by the recommender, in order to achieve an even higher final correlation between them.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), University Autónoma de Madrid and the Community of Madrid (CCG10-UAM/TIC-5877).

References

1. Adomavicius, G., Tuzhilin, T.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. on Knowl. and Data Eng. 17 (6), 734–749 (2005)
2. Amati, G., Carpineto, C., Romano, G.: Query difficulty, robustness, and selective application of query expansion. In ECIR 2004, LNCS, vol. 2997, pp. 127–137. Springer, Heidelberg (2004)

3. Bellogín, A., Castells, P.: A performance prediction approach to enhance collaborative filtering. In ECIR 2010, LNCS, vol. 5993, pp. 382–393. Springer, Heidelberg (2010)
4. Bellogín, A., Wang, J., Castells, P.: Text retrieval methods applied to ranking items in collaborative filtering. In ECIR 2011, LNCS, vol. 6611, pp. 301–306. Springer, Heidelberg (2011)
5. Buckley, C.: Topic prediction based on comparative retrieval rankings. In 27th ACM conference on Research and Development in Information Retrieval, pp. 506–507. ACM Press, New York (2004)
6. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: Predicting query performance. In 25th ACM conference on Research and Development in Information Retrieval, pp. 299–306. ACM Press, New York (2002)
7. Cronen-Townsend, S., Zhou, Y., Croft, W.B.: A framework for selective query expansion. In 13th ACM conference on Information and Knowledge Management, pp. 236–237. ACM Press, New York (2004)
8. Dang, V., Bendersky, M., Croft, W.B.: Learning to rank query reformulations. In 33rd ACM conference on Research and Development in Information Retrieval, pp. 807–808. ACM Press, New York (2010)
9. Hauff, C.: Predicting the Effectiveness of Queries and Retrieval Systems. PhD thesis, University of Twente, Enschede (2010)
10. Hauff, C., Hiemstra, D., de Jong, F.: A Survey of Pre-Retrieval Query Performance Predictors. In 17th ACM conference on Information and Knowledge Management, pp. 439–448. ACM Press, New York (2008)
11. He, B., Ounis, I.: Inferring query performance using pre-retrieval predictors. In SPIRE 2004, LNCS, vol. 2346, pp. 43–54. Springer, Heidelberg (2004)
12. Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In 22nd ACM conference on Research and Development in Information Retrieval, pp. 230–237. ACM press, New York (1999)
13. Hofmann, T.: Latent semantic models for Collaborative filtering. ACM Trans. Inf. Syst. 22(1), 89–115 (2004)
14. Mothe, J., Tanguy, L.: Linguistic features to predict query difficulty. In ACM SIGIR Workshop on Predicting Query Difficulty – Methods and Applications (2005)
15. Pérez-Iglesias, J., Araújo, L.: Ranking List Dispersion as a Query Performance Predictor. In ICTIR 2009, LNCS, vol. 5766, pp. 371–374. Springer, Heidelberg (2009)
16. 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. Gaithesburg, Maryland (2004)
17. Shtok, A., Kurland, O., Carmel, D.: Predicting Query Performance by Query-Drift Estimation. In ICTIR 2009, LNCS, vol. 5766, pp. 305–312. Springer, Heidelberg (2009)
18. Wang, J., de Vries, A. P., Reinders, M. J. T.: Unified relevance models for rating prediction in collaborative filtering. ACM Trans. Inf. Syst. 26(3), 1–42 (2008)
19. 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 ACM conference on Research and Development in Information Retrieval, pp. 512–519. ACM Press, New York (2005)
20. Zhou, Y., Croft, B.W.: Query performance prediction in web search environments. In 30th ACM conference on Research and Development in Information Retrieval, pp. 543–550, ACM Press, New York (2007)