The Magic Barrier of Recommender Systems – No Magic, Just Ratings

Alejandro Bellogín[1], Alan Said[2], and Arjen P. de Vries[3]

[1] Universidad Autónoma de Madrid, Ciudad Universitaria de Cantoblanco, 28049 Madrid, Spain alejandro.bellogin@uam.es
[2] Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands
alansaid@acm.org
[3] Centrum Wiskunde & Informatica, Science Park 123, 1098XG, Amsterdam, The Netherlands
arjen.de.vries@cwi.nl

Abstract. Recommender Systems need to deal with different types of users who represent their preferences in various ways. This difference in user behaviour has a deep impact on the final performance of the recommender system, where some users may receive either better or worse recommendations depending, mostly, on the quantity and the quality of the information the system knows about the user. Specifically, the inconsistencies of the user impose a lower bound on the error the system may achieve when predicting ratings for that particular user.

In this work, we analyse how the consistency of user ratings (*coherence*) may predict the performance of recommendation methods. More specifically, our results show that our definition of coherence is correlated with the so-called *magic barrier* of recommender systems, and thus, it could be used to discriminate between easy users (those with a low magic barrier) and difficult ones (those with a high magic barrier). We report experiments where the rating prediction error for the more coherent users is lower than that of the less coherent ones. We further validate these results by using a public dataset, where the magic barrier is not available, in which we obtain similar performance improvements.

1 Introduction

Recommender systems aim to help people find items of interest from a large pool of potentially interesting items. However, when receiving these recommendations not all users are equally satisfied. One reason for this is, e.g. the choice of the recommendation algorithm. However, even when we account for this aspect, some users may receive better recommendations than others. Previous research has analysed this issue and characterised it as a matter of user inconsistency, that is, users have an inherent noise when interacting with the recommender system, which then affects the reliability of the recommendations produced. This concept is know as the magic barrier of recommender systems, a term coined by Herlocker et al. [9], referring to the upper bound on rating prediction accuracy: above it any further improvements on the evaluation metrics are meaningless [1, 18].

In this context, we aim to infer which users have a higher level of inconsistency *a priori*, that is, find the magic barrier without having the additional re-ratings as required in the approaches suggested until now. We are particulary interested in predicting which users will have a low/high magic barrier using readily available information. We propose to measure how coherent a user's ratings are within an item's feature space. In doing so, we associate highly coherent ratings to users with a lower magic barrier.

Once the magic barrier – or any other measure of user's inconsistency – is successfully predicted, several applications to improve the recommender system's performance become available. One possibility would be to create separate training models for a subset of the users according to their predicted consistency.

Our research aims to answer the following two research questions: **RQ1)** is the rating coherence of a user a good predictor of the magic barrier? and **RQ2)** is it possible to cluster the user community into easy and difficult users – according to their coherence – so that the performance of the system is improved? We address the first question by measuring the correlation between our definitions of coherence and the magic barrier of each user. For the second question, we study how the error of the recommender system changes when considering different subsets of users (selected according to the proposed coherence values) to train and test the models.

The rest of the paper is organised as follows. Section 2 presents other research considering the concept of user inconsistencies, it also defines the magic barrier that we will use throughout this paper. Section 3 describes our approaches to measure the coherence of a user; then, in Section 4 the datasets and other experimental settings are introduced. Finally, Section 5 shows the results obtained, Section 6 provides additional works dealing with the problem of predicting the user's difficulty or the performance of a system, and Section 7 concludes the paper and presents some lines of future work.

2 Measuring User Inconsistency in Recommendation

One of the first works mentioning user-induced noise in movie ratings was presented in [10] by Hill et al., where the authors created an email-based movie recommendation service. The service asked its users to rate movies from a list of 500 pre-selected movies before attempting to create recommendations. The authors mention *The Upper Limit* as a bound on performance prediction based on the idea that a person's ratings are noisy or inconsistent. Based on statistical theory, the authors claim that it will never be possible to perfectly predict the users' ratings, instead they cite *the square root of the observed test-retest reliability correlation* as the optimal level of prediction due to the levels of noise in user-generated data. No attempt at estimating the level of noise was however performed in the scope of that work.

To our knowledge, the first mention of the *magic barrier*, the term currently in use for stating the practical upper bound on rating prediction accuracy (or lower bound on rating prediction error), appeared in Herlocker et al. in their seminal paper on recommender system evaluation [9]. In that work, the authors speculate whether recommender systems are hitting such a potential magic barrier, e.g. a point where *natural variability may prevent us from obtaining much more accurate predictions*. Additionally, the authors speculate whether minuscule rating prediction errors actually translate to a perceived improvement from the users or whether the increasingly smaller accuracy improvements have no effect on the quality as perceived by the end users.

Cosley et al. in [5] conducted an early study on the Movielens[4] website where a selection of users were asked to provide re-ratings to previously rated movies. Similarly, Amatriain et al. attempted to characterise the noise in ratings based on the reliability in the re-rating process [1, 2]. More recently, Kluver et al. addressed this problem from a different direction [12], where instead of measuring the level of noise in the dataset, the authors measure how much preference information is contained in a rating. To do this, the authors based their approach on Shannon's information entropy, which indicates how much actual information is concealed in a rating. Preference bits were then found through repeated re-ratings by users on the same items. Each re-rating can then be used to estimate the amount of preference bits in a rating.

In this paper, we focus on the definition of magic barrier as defined by Said et al. in [18] and [19]. This concept is derived as the lower bound of the Root Mean Squared Error (RMSE) that can be attained by an optimal recommender system. It is defined as the standard deviation of the inconsistencies (noise) inherent in the user ratings, as follows:

$$\widehat{B}_{\mathcal{X}} = \sqrt{\frac{1}{|\mathcal{X}|} \sum_{(u,i) \in \mathcal{X}} (r(u,i) - o(u,i))^2} \tag{1}$$

where $\mathcal{X}$ is the set of ratings for which we have re-ratings (opinions, in that work) available, $r(u,i)$ denotes the actual rating for user u on item i, and $o(u,i)$ is the opinion given by the user at a different point in time than $r(u,i)$. Note that this definition of the magic barrier actually is an estimation as it is not possible to directly determine the magic barrier because it involves an optimal rating function which is not usually available [18].

3 A Measure of User Coherence for Recommendation

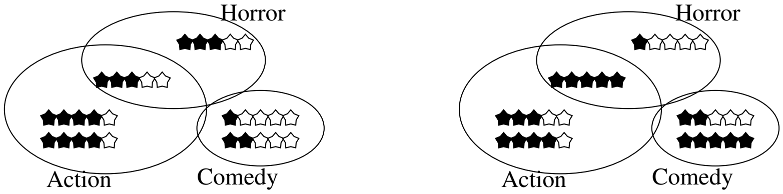
Given a user u, her rated items $I(u) \subseteq \mathcal{I}$, and the ratings $r(u,i)$ assigned to these items, i.e., $i \in I(u)$, we aim to provide a score $\gamma(u) = \gamma(I(u))$ that measures how coherent a user profile is in terms of her assigned ratings. To compute this score we propose to use an external information source with which we can measure the inconsistencies of the user's ratings, by describing items in terms of specific features, e.g. genres. Although other measures could be available where no external information is required, such as the entropy of the ratings [12] or the Kullback-Leibler divergence between the user's preferences and the overall preferences [4], we believe that our formulation provides a measure that is easily explainable and justifiable, allowing for further feedback from the recommender system to the user. Furthermore, as we show in the rest of the paper, this measure obtains very good results, despite its simplicity.

3.1 Example

Before presenting the actual definition of the rating coherence of a user (or *user coherence*, for simplicity), let us first consider the examples presented in Figure 1. Here

[4] http://www.movielens.org

we have two users that have rated the exact same set of items, although their ratings are slightly different. Specifically, the user in Figure 1a gives more consistent ratings to items sharing the same features, which in this case corresponds to the movie's genres. In the long term we argue that such a user will have a more consistent (less noisy) behaviour, since her taste for each item feature seem to be well defined.



(a) A coherent user ($c(u) = -1.28$)

(b) A not coherent user ($c(u) = -5.95$)

Fig. 1: Example of a coherent vs. not coherent user. Our definition of coherence takes into account the rating's deviation within each item feature, which in this example consists of three genres: action, comedy, and horror.

3.2 A Simple Definition for User Coherence

Following the rationale presented before, we define the user coherence based on a set of item features $\mathcal{F}$ as:

$$c(u) = -\sum_{f \in \mathcal{F}} \sigma_f(u) \tag{2}$$

where $\sigma_f(u) = \sqrt{\sum_{i \in I(u,f)} (r(u,i) - \overline{r}_f(u))^2}$ corresponds to the user's rating deviation within a specific feature f, having an associated mean rating for that feature $\overline{r}_f(u)$, which simply corresponds to the average rating within the set of items rated by user u that belong to feature f, denoted here as $I(u, f)$. We refer to this formulation as **basic coherence**.

With this formulation, the coherence $c(u)$ captures the variance of a user's ratings relative to the feature space in which the items are defined. Moreover, it also incorporates a negative sign to indicate that the larger the variance, the less coherent (or more incoherent) a user should be.

The key aspect of this function, hence, is that we are accounting for the rating deviation of the user with respect to a particular feature space. Besides, such definition allows for a more general case, where the space $\mathcal{F}$ could be – instead of (textual) item features – any embedding of the items into a space $\mathcal{F}$, such as an item clustering or the latent factors of the items, and also other functions apart from standard deviation to statistically summarise the user's ratings for a given feature, as will be described in the next section.

Table 1: Possible functions $g(u, f)$ to be used in Equation 3, where $u(f)$ denotes the user's ratings associated with items linked to feature f, $u(\mathcal{F})$ is the same but for any feature in the feature space $\mathcal{F}$, and the probabilities $p(f|u)$ and $p(f)$ are computed normalising the rating values of a user or of the whole community for a given feature.

Function $g(u, f)$	Definition	Function $g(u, f)$	Definition
Entropy	$p(f\|u) \log p(f\|u)$	KLD	$p(f\|u) \log \frac{p(f\|u)}{p(f)}$
Mean	$\mu(u(f))$	Weighted Mean	$\mu(u(f)) \frac{\\|u(f)\\|}{\\|u(\mathcal{F})\\|}$
Std. dev.	$\sigma(u(f))$	Weighted Std. dev.	$\sigma(u(f)) \frac{\\|u(f)\\|}{\\|u(\mathcal{F})\\|}$
Size	$\\|u(f)\\|$		

Going back to our previous example presented in Figure 1, we can observe that the values of $c(u)$ match our intuition about which user is more coherent, since user in Figure 1a receives a higher value from the proposed measure.

3.3 A General Definition for User Coherence

Based on the simple formulation presented in Equation 2, we now introduce a more general definition for the coherence of a user. We now allow (see Equation 3) any arbitrary function defined upon the information known for a user u and a specific feature f to be used. This information will generally be the ratings given by u to the items associated with f, that is $u(f) = \{r : (u, i, r)\ \forall i \in \mathcal{F}^{-1}(f)\}$, where $\mathcal{F}(i)$ denotes the subset of features $f \in \mathcal{F}$ for item i.

$$c_g(u) = -\sum_{f \in \mathcal{F}} g(u, f) \tag{3}$$

Table 1 shows some possibilities for these functions applied over the vector of ratings $u(f)$. Entropy, Kullback-Leibler divergence (KLD), standard deviation, mean, and size are presented in the table, along with two weighted versions of the standard deviation and mean to account for the actual number of items rated by the user in each feature. We have to note that the basic coherence presented in the previous section corresponds to the one where standard deviation is used as the function $g(u, f)$.

We have to emphasise that any of these variations of coherence can be calculated using the same data available for training the recommender system, and that no information from the test set is required.

4 Experimental Setup

We now describe the two datasets used to test the predictive power of the proposed measures for user coherence introduced in Section 3. We also present the specific training and test splits we generate to properly assess the performance improvement by exploiting a user clustering into easy and difficult users once the proposed coherence measure is used.

Table 2: Statistics of the datasets used for the experiments, where *opinions* refers to those used to estimate the magic barrier as in [18].

Dataset	Users	Items	Ratings	Density	Range of ratings
Movielens	6,040	3,900	1,000,209	4.24%	[1-5]
Moviepilot	318,418	31,948	12,825,203	0.13%	[0-100]
Moviepilot opinions	306	2,309	6,299	0.89%	[0-100]

4.1 Datasets

We have used two datasets, whose statistics are presented in Table 2: Moviepilot and Movielens. The former is a snapshot of the commercial movie recommender system Moviepilot[5], having more than one million users, 55,000 movies, and over 10 million ratings. Movies are rated on a 0 to 100 scale with a step size of 5. To estimate the magic barrier, in [18] a user study was performed to collect users *opinions* on movies that had been previously rated on Moviepilot. We refer the reader to the detailed description contained in that paper, the relevant part for the present work is that every user taking part in this study gave their opinion on at least 20 movies, which were collected and aggregated to calculate the magic barrier of the system as presented in Section 2.

The second dataset used in our experiments is one of the datasets provided by Movielens[6], containing one million ratings, more than 6,000 users and almost 4,000 items, as we can see in Table 2. Since the re-ratings or other opinions are not available for this dataset, the magic barrier cannot be estimated, but we can still use it (in Section 5.2) as a proof of concept that the proposed coherence functions are able to discriminate between difficult and easy users.

Table 3: Notation for the different training and test models considered.

Name	Training	Test
All	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	$\mathrm{Te}_e \cup \mathrm{Te}_d$
All-Easy	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	Te_e
All-Diff	$\mathrm{Tr}_e \cup \mathrm{Tr}_d$	Te_d
Easy-Easy	Tr_e	Te_e
Diff-Diff	Tr_d	Te_d

4.2 Training and Test Splits

As stated in the research question **RQ2**, we aim to check if it is possible to cluster the users (into easy and difficult ones) such that the performance of the system is improved

[5] http://www.moviepilot.de/

[6] Available at http://www.grouplens.org/node/73

as a result of this user partition. To properly evaluate this, we propose the splits for the training and test sets summarised in Table 3. The training and test splits for the easy users are denoted as Tr_e and Te_e, whereas the splits corresponding to the difficult users are referred to as Tr_d and Te_d. Assuming that we have already classified the users as easy or difficult ones (e.g., using a percentage p of all the users labelled as easy users), we perform a 5-fold cross validation within the whole set of ratings relative to each of the easy and difficult users, in order to obtain a training and test split for each subset of the data. Other splitting conditions may be used (based on percentage or time conditions) instead of 5-fold, and will be considered in the future.

Once these splits are generated, we build the combinations for training and test models presented in Table 3. The rationale of these models goes as follows: the *All* model is a simple evaluation split where all the users are used to train and test a specific recommender system; the *Easy-Easy* and *Diff-Diff* evaluation models focus only on one type of user, by training and testing on the ratings associated with that corresponding type of user. The other two combinations use the same training information available as in *All*, but only evaluate one type of user, either those classified as easy users (*All-Easy*) or as difficult ones (*All-Diff*).

5 Results

In this section we present two experiments that aim to answer the research questions stated earlier, i.e. **RQ1)** is the rating coherence of a user a good predictor of the magic barrier?, and **RQ2)** is it possible to cluster the users into easy and difficult users so that the overall performance of the system is improved?

For these experiments, we have used the datasets described in Section 4.1. For the second experiment, we performed a standard 5-fold cross-validation evaluation, splitting the data according to the different strategies described in Section 4.2, where we assume that half of the users are easy and the other half are difficult (i.e., $p = 0.5$ from previous section), once they are sorted according to a particular coherence function. In each fold, the training and test splits contained 80% and 20% of the data respectively.

The recommendation algorithm tested is a standard user-based collaborative filtering method [21], using Pearson's correlation as the user similarity function and 50 neighbours.

Furthermore, for the features used to compute the coherence functions, we exploit, for the Movielens dataset, the genres provided in the original files. In the case of Moviepilot we use four of the available tagging features [17]: genres, plot keywords, emotion keywords, and intended audience.

The evaluation metric we use in our experiments is the Root Mean Squared Error (RMSE). We report this metric because it is related with the concept of magic barrier (indeed, the magic barrier is defined as *the RMSE of an optimal function* [18]), although in the future we plan to explore alternative evaluation metrics, e.g., precision, recall, nDCG, etc. RMSE is calculated for a test set T as:

$$\text{RMSE} = \sqrt{\frac{1}{|T|} \sum_{(u,i) \in T} (r(u,i) - \tilde{r}(u,i))^2} \tag{4}$$

Table 4: Spearman's correlation between coherence and user magic barrier. $\emptyset$ indicates that no feature space was used. Note that the desired correlation is negative as the more coherent a user is, the better performing she is (i.e. she has a lower magic barrier).

Coherence	Genres	Emotion keywords	Intended audience	Plot keywords	$\emptyset$
Entropy	0.050	0.016	0.048	0.000	NA
KLD	0.098	0.055	0.067	0.068	NA
Mean	0.114	0.113	0.097	0.106	0.104
Weighted Mean	0.010	0.068	0.072	-0.028	0.104
Std. dev.	-0.331	-0.438	-0.383	-0.279	-0.432
Weighted Std. dev.	-0.398	**-0.455**	-0.432	-0.394	-0.432
Size	0.077	0.074	0.066	0.088	0.072
Random	-0.015				
Number of ratings	-0.072				
Average rating	-0.104				

where $r(u, i)$ and $\tilde{r}(u, i)$ denote the real and predicted ratings for every pair of item i and user u contained in T.

5.1 User Coherence and Magic Barrier

In this experiment, we assess the validity of the proposed coherence functions as good predictors for the magic barrier to answer the research question **RQ1**. With this goal in mind, we show in Table 4 the Spearman's correlation values between the coherence and the magic barrier per user (Pearson's correlation was very similar). Note that Pearson's correlation coefficient is designed to capture linear relationships between the two variables whereas Spearman's captures non-linear dependencies. Both correlations provide scores in the range of -1 to 1, where 1 denotes a perfect correlation, -1 represents an inverse correlation, and the absolute value is the strength of the relationship.

We observe in Table 4 that the correlations for the weighted version of the coherence function (that is, where the importance of each feature in the user profile is ignored) show more predictive power only when the standard deviation is used. Besides, entropy and KLD do not perform very well. Additionally, *Emotion keywords* and *Intended Audience* seem to be the best feature spaces for most of the coherence formulations, and especially, for the cases where a strong correlation is obtained. We have to however note that these feature spaces offer a low coverage in terms of the items identified with these features [20], thus this aspect should also be taken into account when selecting the feature to use.

We have also analysed the behaviour of the proposed coherence functions when the feature space is reduced to having only one feature (which is shared among all the items). The results for this case (column $\emptyset$ in Table 4) evidence that the actual feature space may not be so important, and that the proposed coherence functions (except Entropy and KLD, which produce the same value – a zero – when computed for an event space of size 1) are able to predict a user's magic barrier using exclusively ratings. This is especially true when standard deviation is used as function g. Note, however, that

Table 5: RMSE values using different features for the coherence and the training and test splits described in Section 4.2. ▲ and ▼ denote, respectively, the best and worst values obtained in each dataset (the lower the error, the better).

	Training and Test Splits				
Dataset	All	All-Easy	All-Diff	Easy-Easy	Diff-Diff
Moviepilot	23.097	20.079	26.278	19.279▲	28.219▼
Movielens	1.090	0.974	1.195	0.933▲	1.226▼

although the obtained correlations in this case are similar to those presented before, strongest relations are always found when a feature space is used.

In our analysis, we have also included a random magic barrier predictor to check its neutral correlations (around zero), along with two other baseline predictors based on the number of ratings each user has and her average rating. These results show that the proposed coherence function is not trivial, and that it is actually capturing something that other transformations based on the same information (ratings) are not able to provide.

This experiment hence confirms that the user coherence measured as proposed in Section 3 provides good predictions of the magic barrier; as a consequence, we should be able to exploit the ranking generated by sorting the users according to their coherence value to lower the magic barrier for the more coherent (or easy) users. In the next section, we show that this may be generalised when no information about the magic barrier is available, and only the final RMSE of the system can be measured.

5.2 User Coherence and Recommendation Performance

Now we aim to address research question **RQ2**, where we investigate if we can improve the recommendation performance by clustering the population of users according to their coherence.

Based on the results presented in the previous section, we are going to restrict our analysis on the weighted version of the standard deviation function for user coherence. Moreover, to ensure fair comparisons between Movielens and Moviepilot datasets, we use genres as feature space; recall that in this situation the correlation was not the strongest, but it was also significant (almost -0.4).

Table 5 shows the RMSE values for the different training and test splits presented in Table 3. We observe that the best result is always obtained when only easy users are included in the training set, that is, those classified as more coherent, and according to the correlation analysis, those having a lower magic barrier. Similarly, the difficult (less coherent) users produce the worst recommendation performance. This is an indicative that the magic barrier is a valid estimation of the final performance of the system.

Moreover, we also notice that the baseline performance (from the *All* split) is reduced when only easy (more coherent) users are evaluated. Besides, if we average the error found in *All-Easy* and *All-Diff* (since in each of these splits half of the users were evaluated) in Movielens, we obtain slightly better results than when we evaluate the complete dataset (specifically, we have an average RMSE of 1.0845). Note that in the three splits the recommendation model learnt from the training data is the same and the only difference is in the test set used to compute the RMSE.

These results evidence that the coherence function we have proposed in this paper is able to detect users that exhibit an inherent lower noise, even when no information about the magic barrier is available. On top of this, the user-based recommender we have tested takes advantage of this aspect and learns (and predicts) their preferences more accurately. On the other hand, the difficult users do not only receive bad recommendations, but they can improve their accuracy by training the recommendation model with more data; that is, whereas coherent users obtain decent performance by using only ratings from other coherent users, less coherent users need information from outside of their own cluster, showing their higher level of noise.

In summary, this experiment answers positively to the second research question, namely, that it is possible to exploit the coherence values to build different training and test models in such a way that the error decreases for the easy users, and in some cases (i.e., the Movielens dataset), even the average error obtained for the easy and difficult users is balanced out and outperform the overall error.

6 Related Work

Aiming to understand how recommenders fail for certain users, and attempting to characterise those users has been researched by some authors. Rashid et al. propose in [16] a measure of the effect of a user in the recommendations received by an algorithm, named as influence. Their original definition is very expensive, since it measures the effect a user has over the rest via the predictions they receive, for which they need to compute predictions for items using a training model where the target user has been removed.

In [6], Ekstrand & Riedl examine why some recommenders fail in the context of hybrid recommendation, with the goal of selecting better components to build more efficient ensembles. They found that recommenders fail on different users and items, and obtained specific user features – such as the user's rating count, the average rating, and their variance – that allow to predict the performance of an algorithm.

In [11], Kille & Albayrak assign a difficulty value reflecting the expected evaluation outcome of the user. The authors propose to measure this difficulty in terms of the diversity of the rating predictions and rankings when comparing the output of several recommender systems. Some diversity metrics from [13] are proposed, but they are not tested nor implemented in any real dataset.

By drawing from Information Retrieval related quantities, Bellogín et al. present in [3, 4] a family of performance predictors for users. Correlations found between ranking-based metrics and such predictors are strong, and the authors propose to exploit them in at least two applications: dynamic neighbourhood building and dynamic ensemble recommendation, where the weights for the neigbours or the recommenders would dynamically change depending on the predicted performance of each variable.

More recently, a similar approach was developed using a machine learning method based on decision trees. In [7], Griffith et al. aim to predict the user's performance in terms of the user's average error by extracting user's rating information (such as the number of ratings, average rating, standard deviation, number of neighbours, average similarity, etc.). The correlations obtained are very strong (around 0.8) but no actual applications are proposed in that paper.

In summary, the idea of predicting the performance of recommenders has attracted a lot of attention in the field so far, however, to the best of our knowledge, no other work has been able to predict an actual measure of user inconsistency – like the magic barrier – and successfully apply it to improve the performance of the whole (or even of a subset) of the system, as we have presented in this work.

7 Conclusions and Future Work

The research presented here aims to provide a deeper understanding of what user characteristics are related with the appropriateness and relevance of the recommender's suggestions for each user. We have observed that being statistically coherent – in terms of rating deviation – gives enough information to predict the user inconsistency as measured by her magic barrier, especially if such coherence is measured within an item's feature (e.g., genres). This opens up the possibility for a (production) recommender system to perform different actions on the users depending on their predicted inconsistencies, such as proactively asking some specific users (the ones predicted as most *difficult*) to rate more items or training separate models for the *easy* and *difficult* users. Our experiments show that by creating such separate training models improvements with respect to the global model can be achieved, specifically, around 14% for the subset of easier users. It should be feasible to improve these results if an ad-hoc tuning over the set of more difficult users is performed.

We have explored only one type of recommendation algorithm (i.e., collaborative filtering, and in particular, user-based methods), but it is an open question whether all recommenders may improve their performance in the same way with respect to the proposed coherence function. More importantly, apart from the error-based evaluation metrics used in this work, we are interested in evaluating with ranking-based metrics – like precision – to analyse if user inconsistencies are also reflected in terms of, or if they affect whatsoever, their ranking performance. We also plan to develop other coherence-related predictors proposed in the fields of Information Retrieval [8] and Machine Learning [14, 15], to capture additional insights about the user behaviour with respect to the recommender system.

8 Acknowledgments

This work was carried out during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2014) under grant agreements no. 246016 (ERCIM) and no. 610594 (CrowdRec).

References

1. Amatriain, X., Pujol, J.M., Oliver, N.: I like it... I like it not: Evaluating user ratings noise in recommender systems. In: UMAP. pp. 247–258 (2009)
2. Amatriain, X., Pujol, J.M., Tintarev, N., Oliver, N.: Rate it again: increasing recommendation accuracy by user re-rating. In: RecSys. pp. 173–180 (2009)
3. Bellogín, A.: Predicting performance in recommender systems. In: RecSys. pp. 371–374 (2011)
4. Bellogín, A., Castells, P., Cantador, I.: Predicting the performance of recommender systems: An information theoretic approach. In: ICTIR. pp. 27–39 (2011)
5. Cosley, D., Lam, S.K., Albert, I., Konstan, J.A., Riedl, J.: Is seeing believing?: how recommender system interfaces affect users' opinions. In: CHI. pp. 585–592 (2003)
6. Ekstrand, M.D., Riedl, J.: When recommenders fail: predicting recommender failure for algorithm selection and combination. In: RecSys. pp. 233–236 (2012)
7. Griffith, J., O'Riordan, C., Sorensen, H.: Investigations into user rating information and predictive accuracy in a collaborative filtering domain. In: SAC. pp. 937–942 (2012)
8. He, J., Larson, M., de Rijke, M.: Using coherence-based measures to predict query difficulty. In: ECIR. pp. 689–694 (2008)
9. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.: Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst. 22(1), 5–53 (2004)
10. Hill, W.C., Stead, L., Rosenstein, M., Furnas, G.W.: Recommending and evaluating choices in a virtual community of use. In: CHI. pp. 194–201 (1995)
11. Kille, B.: Modeling difficulty in recommender systems. In: RUE '12. pp. 30–32. RecSys (2012)
12. Kluver, D., Nguyen, T.T., Ekstrand, M.D., Sen, S., Riedl, J.: How many bits per rating? In: RecSys. pp. 99–106 (2012)
13. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning 51(2), 181–207 (2003)
14. Misra, H., Cappé, O., Yvon, F.: Using lda to detect semantically incoherent documents. In: CoNLL. pp. 41–48. Stroudsburg, PA, USA (2008)
15. Newman, D., Bonilla, E.V., Buntine, W.L.: Improving topic coherence with regularized topic models. In: NIPS. pp. 496–504 (2011)
16. Rashid, A.M., Karypis, G., Riedl, J.: Influence in ratings-based recommender systems: An algorithm-independent approach. In: SDM (2005)
17. Said, A., Berkovsky, S., Luca, E.W.D.: Movie recommendation in context. ACM TIST 4(1), 13 (2013)
18. Said, A., Jain, B.J., Narr, S., Plumbaum, T.: Users and noise: The magic barrier of recommender systems. In: UMAP. pp. 237–248 (2012)
19. Said, A., Jain, B.J., Narr, S., Plumbaum, T., Albayrak, S., Scheel, C.: Estimating the magic barrier of recommender systems: a user study. In: SIGIR. pp. 1061–1062 (2012)
20. Said, A., Kille, B., De Luca, E.W., Albayrak, S.: Personalizing tags: a folksonomy-like approach for recommending movies. In: HetRec '11. pp. 53–56. RecSys, New York, NY, USA (2011)
21. Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating ”word of mouth”. In: Katz, I.R., Mack, R.L., Marks, L., Rosson, M.B., Nielsen, J. (eds.) CHI. pp. 210–217. ACM/Addison-Wesley (1995)