Evaluating Decision-Aware Recommender Systems

Rus M. Mesas
Universidad Autónoma de Madrid
Madrid, Spain
rusmaria.mesas@estudiante.uam.es

Alejandro Bellogín
Universidad Autónoma de Madrid
Madrid, Spain
alejandro.bellogin@uam.es

ABSTRACT

The main goal of a Recommender System is to suggest relevant items to users, although other utility dimensions – such as diversity, novelty, confidence, possibility of providing explanations – are often considered. In this work, in order to increase the amount of relevant items presented to the user, we analyse how the system could measure the confidence on its own recommendations, so it has the capability of taking decisions about whether an item should be recommended or not. A direct consequence of this design is that the number of suggested items decreases, impacting in some of the beyond-accuracy dimensions (especially, coverage). We present an evaluation of different decision-aware techniques that can be applied to some families of recommender systems, and explore evaluation metrics that allow to combine more than one evaluation dimension. Empiric results show that large precision improvements are obtained when using these approaches at the expense of user and item coverage.

1 INTRODUCTION

Recommender Systems aim at suggesting as many relevant items to the users as possible, although other goals are being considered recently [3]: increasing the diversity or novelty of the recommendations, suggesting items in such a way that it is possible to explain where the recommendation is coming from, increasing the confidence of the user in the system, etc. In this work, we investigate about confidence but from the perspective of the system: what is the confidence a system has in its own recommendations; more specifically, we focus on different methods to embed awareness into the recommendation algorithms about deciding whether an item should be suggested. In this way, we hypothesise the system would only show the more reliable suggestions, hence, increasing the performance of such recommendations, at the expense of, presumably, reducing the number of potential recommendations.

The concept of confidence in recommendation has been applied to different aspects in the field. On the one hand, confidence is defined on the input data, where some authors have studied which combinations of users or (user, item) pairs let generate better recommendations, namely the *profile-level trust* and *item-level trust* approaches from [9], or like in [6], where it is used to interpret the confidence when transforming from implicit data (frequencies) into explicit. On the other hand, we find different mechanisms that help to contextualise the predictions made by the recommendation algorithms. In this way, in [4] a factor is defined (*significance weighting*) that is combined with the user similarity to devalue those cases where such similarity has been computed with not enough data. In a similar fashion, in [1] a method is proposed to filter out recommendations according to the rating deviation received by the items. Previously, in [12], the authors introduced the concepts of support and confidence in case-based recommenders computed from association rules. In parallel, in [8] it was studied how to introduce confidence measures when presenting the recommendations, that is, in the interface the user is interacting with. In that work, even though the confidence metric was very simple (number of ratings observed by the system for one item) it was enough to increase the satisfaction level of the user.

In this paper, we study how the performance of a recommendation algorithm evolves when it decides not to recommend in some situations. If the decision of avoiding a recommendation is sensible – i.e., not random but related to the information available to the system about the target user or item –, the performance is expected to improve at the expense of other quality dimensions such as coverage, novelty, or diversity. This balance is critical, since it is possible to achieve a very high precision recommending only one item to a unique user, which would not be a very useful recommender. Memory-based algorithms are well known for suffering from this issue: if very few neighbours are considered, the coverage is lower, but the recommendations are of higher quality (as observed in terms of error metrics [4]); whereas larger neighbourhoods may increase the likelihood of receiving a noisy recommendation, but the chances of recommending more items are also higher. Because of this, we explore some techniques to combine precision and coverage metrics, an open problem in the area, as stated by some authors [3].

In summary, the contributions of this paper are twofold: a taxonomy of techniques that can be applied to some families of recommender systems allowing to include mechanisms to decide if a recommendation should be generated, and a first exploration to the combination of precision and coverage evaluation metrics.

2 DECISION-AWARE RECOMMENDER SYSTEMS

Modifying recommendation algorithms so that they can decide whether a recommendation should be produced is not an easy task when formulated in a generic way, because each algorithm has different characteristics and hypothesis about the input data and produced suggestions. As a starting point, in this section we shall focus on one of the main types of recommender systems – Collaborative Filtering algorithms.

Different from other works in the literature, our approaches do not exploit or analyse the input data (unlike the works previously mentioned [6, 8]), but intrinsic aspects of the recommendation algorithms or of the components used during prediction are considered, similar to the weights introduced for similarity computation in [4]. More specifically, we exploit the support of the prediction score for nearest-neighbour algorithms (Section 2.1) and the uncertainty in the prediction score for a probabilistic matrix factorisation algorithm (Section 2.2).

RecSys '17, August 27-31, 2017, Como, Italy

ACM ISBN 978-1-4503-4652-8/17/08…$15.00.
https://doi.org/10.1145/3109859.3109888

2.1 Based on prediction support

It is well known that, in order to compute scores for (user, item) pairs, some algorithms use more data than others, depending on the actual users and items under consideration. A paradigmatic example of this situation are the nearest-neighbour recommenders. These algorithms estimate the user preferences based on similar users or items, however, it is required that those neighbours have rated the same item (in the user-based scenario) or that the target user has rated those similar items (in the item-based case).

Indeed, the number of ratings used to predict the user preferences (denoted as *support*) provides an indication of the confidence level the system has about the produced recommendation, since the system cannot trust in the same way a score produced based only on two neighbours or based on one hundred. This is actually the same idea behind the significance weighting approach proposed by Herlocker and colleagues in [4].

Hence, we propose that a nearest-neighbour recommender could decide that **an item is worth a recommendation if at least n out of the k neighbours have participated in the preference computation**. In other words, a prediction would be ignored if less than n neighbours have contributed to the prediction score.

2.2 Based on prediction uncertainty

As described in the previous section, some algorithms compute the prediction scores based on an aggregation of values, usually an average or a weighted average (such as the aforementioned neighbour-based recommenders). Whenever an average is being calculated, it is also possible to compute a standard deviation of the predicted score. When doing this, the standard deviation can be interpreted as a confidence parameter of the algorithm about the score: the larger the deviation, the more uncertainty on the prediction, and hence, the lower the confidence on it.

Hence, we propose that an algorithm, for which it is possible to compute the standard deviation of a prediction score, could decide that **an item is worth a recommendation if the standard deviation of the prediction (*uncertainty*) does not exceed a specified threshold σ_τ**.

More specifically, we apply this approach to a probabilistic matrix factorisation recommender. Although it is possible to apply this strategy to nearest neighbour recommenders, we found this approach is less effective for these algorithms, probably because there is no theoretical formulation for the deviation in those cases.

We describe next how we can compute the standard deviation of a probabilistic matrix factorisation algorithm. For this, we make use of a Bayesian approximation for matrix factorisation proposed in [7]. In this algorithm, the preference scores are computed by approximating a distribution, whose average and deviation should be estimated. Since we need an explicit formulation for the standard deviation, we derived it using mean-field variational inference:

$$Var(r_{ui}) = \tau^2 + \text{trace}\left(\left(\phi_u + \overline{u}\overline{u}^\top\right)\left(\psi_i + \overline{i}\overline{i}^\top\right)\right) - \overline{i}^\top \overline{u}\overline{u}^\top \overline{i} \quad (1)$$

where it is assumed the rating follows a normal distribution with mean $u^\top \cdot i$ and deviation τ, and ϕ_u y $\overline{u}$ represent the covariance matrix and vector of means for user u; ψ_i and $\overline{i}$ denote the same concepts but for item i.

3 EVALUATING DECISION-AWARE RECOMMENDER SYSTEMS

As soon as a recommender system has control over which items should not be recommended for a particular user, it is very likely that the user coverage and the item coverage decrease, even though precision and other accuracy-related metrics increase. For instance, a very high precision could be achieved if an algorithm returns only one (relevant) item for one user, at the expense of a very low coverage. Because of this, it becomes very important to study and define metrics that, somehow, combine precision and coverage, especially in situations of confidence-awareness like the one we propose in this paper.

As noted by Herlocker and colleagues in [5] there is no general coverage metric that, at the same time, gives more weight to relevant items when accounting for coverage, and combines coverage and accuracy measures. Moreover, Gunawardana and Shani mentioned the problem of balancing coverage and accuracy metrics in [3], and leave it as an open issue in the area. We aim to address this problem by combining the values of the different metrics to be compared, especially focused on deriving a metric that assess when a recommender does not return an item.

3.1 F-score or Harmonic Mean

The F-score is an evaluation metric very popular in Machine Learning to combine precision and recall measures. It produces the harmonic mean of both metrics, and it ranges between 0 and 1, 0 being the worst value and 1 the optimal value. Based on this idea, we propose to combine precision and coverage through the harmonic mean, whose general formulation is as follows:

$$F_\beta(P,Q) = (1+\beta^2) \cdot \frac{P \cdot Q}{\beta^2 \cdot P + Q} \quad (2)$$

where β is used to control the importance of each metric in the final result: if $\beta = 1$ both metrics P and Q have the same importance, whereas if $\beta < 1$ P is more important than Q.

3.2 G-score or Geometric Mean

Instead of using the harmonic mean, we could also use the geometric mean as follows:

$$G_{\alpha_1,\alpha_2}(P,Q) = \left(P^{\alpha_1} \cdot Q^{\alpha_2}\right)^{1/(\alpha_1+\alpha_2)} \quad (3)$$

where α_1, α_2 control the importance of each metric. In general, the result obtained for the G-score will always be larger (or equal) to the one obtained for the F-score.

3.3 Correctness

The two metrics defined in the previous sections simply combine the result of some evaluation measures; however we believe this is not enough for the problem we want to address. Typical ranking-based metrics – such as precision – assume that no returning an item which was previously asked to predict a rating for, is an advocate of that item being considered as not relevant by a specific recommendation method. However, this is in contrast with the (desired) situation that a recommender may not provide suggestions in some situations due to a low confidence in the accuracy of such predictions [3, 5].

We propose an evaluation metric that is able to assess when a recommender decides not to recommend a specific item. To do this, we adapt an extension of accuracy proposed in the context of Question Answering by Peñas and Rodrigo in [10]. In that work, the authors assume that there are several questions to be answered by a system, each question has several options, but one (and only one) of those options is correct. If it is possible to give no response for a given question, this action should not be correct, but not incorrect either. Hence, the authors propose a general formulation giving a weight – proportional to the number of correctly answered questions – to the value of unanswered questions.

To apply this evaluation metric to recommendation, we first assume that the set of recommenders we want to compare will receive the same list of items to be ranked, a standard situation shared by many evaluation methodologies [2]. Then, the equivalence between a Question Answering system and a recommender is made – in a user basis – by considering each recommendation algorithm as a different system that will answer (or not) to the questions available, represented as the candidate items to be ranked by a specific methodology. We instantiate four versions of this metric, two of them based on users and two for items:

$$\text{User Correctness}(u, r, N) = \frac{1}{N}\left(TP(u) + TP(u)\frac{NR(u)}{N}\right) \tag{4}$$

$$\text{Recall User Correctness}(u, r, N) = \frac{1}{N}\left(TP(u) + \frac{TP(u)}{|T(u)|}NR(u)\right) \tag{5}$$

$$\text{Item Correctness}(i, r, N) = \frac{1}{|U|}\left(TP(i) + \frac{TP(i)}{|U|}NR(i)\right) \tag{6}$$

$$\text{Recall Item Correctness}(i, r, N) = \frac{1}{|U|}\left(TP(i) + \frac{TP(i)}{|T(i)|}NR(i)\right) \tag{7}$$

where N is the amount of items requested to the recommender r, $TP(u)$ denotes the relevant items recommended to the user, $FP(u)$ the not relevant items being recommended, which combined gives $|T(u)| = TP(u) + FP(u)$; finally, $NR(u) = N - (TP(u) + FP(u))$ denotes the number of unanswered recommendations. The analogous quantities could be defined for the item-based version of the metrics.

4 EXPERIMENTS AND RESULTS

4.1 Experimental Settings

In this paper we have used three datasets from two different domains: two versions of the MovieLens (ML)[1] (ML-100K and ML-1M) dataset and Jester[2]. ML-100K includes 100, 000 ratings by 943 users on 1, 681 items (movies), ML-1M contains 1, 000, 209 ratings by 6, 040 users on 3, 883 movies, and Jester includes 1, 710, 677 ratings on 150 items (jokes) by 59, 132 users. The rating scale on the first two datasets is [1, 5], and on Jester is [−10, 10] (that we moved into [0, 20] to avoid negative ratings).

Some of the algorithms used in the experiments are based on implementations found in RankSys [13], specifically, the nearest-neighbour algorithms and their modifications. The probabilistic matrix factorisation method was implemented by ourselves.

Finally, regarding the evaluation, two publicly available frameworks were used: RankSys and RiVal [11]. On top of the latter framework we implemented the following evaluation metrics: User Space Coverage representing the ratio of users that have received at least one (*USC*) or N (*USC@N*) items as recommendations, Item Space Coverage representing the ratio of items that were recommended to any user by a system returning at most N items (*ISC@N*), and the Correctness metrics as defined in Section 3.3: *UC@N* (User Correctness), *RUC@N* (Recall User Correctness), *IC@N* (Item Correctness) and *RIC@N* (Recall Item Correctness). The RankSys framework was used to obtain novelty and diversity metrics, specifically EPC and AggrDiv [14].

4.2 Performance of decision-aware strategies

In this section we evaluate the performance of the different decision-aware strategies presented in Section 2 using the evaluation metrics described in Section 3. When we apply the strategy based on prediction support, the change in coverage is not significant, even though the performance increases slightly; hence, there is no balance to solve, and the algorithm with the highest precision is the clear winner – see in Table 1 the results for ML-100K. User coverage remains almost unchanged until $n \geq 7$, although precision increases even for smaller values of n until $n = 7$. Considering this information, the three versions of the harmonic mean, G, and $G_{2,1}$ all agree on the ranking of the systems, where the best algorithms are those with n equals 5, 6, and 4, in that order. We observe that *UC* and *RUC* do not discriminate much more than that, however, when *IC* and *RIC* are analysed, the best recommenders are not the same as before ($n = 4, 5$) which makes sense because these techniques take the item coverage into account, which decreases more abruptly than the user coverage. Similarly, $G_{1,2}$ also changes the ranking of systems because it gives a higher weight to coverage than precision.

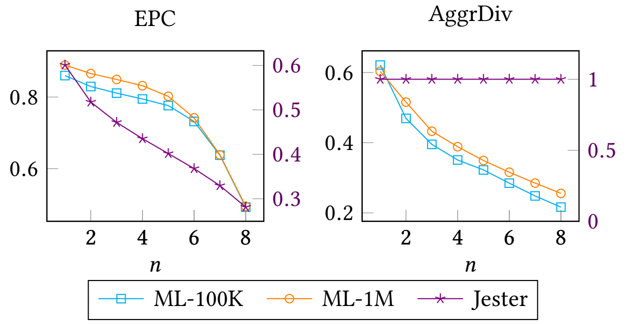

Figure 1: Novelty and diversity metrics at cutoff 10 for a decision-aware strategy based on prediction support. Jester is plotted in the secondary (right) axis.

Figure 1 shows the impact that a decision-aware strategy based on prediction support has on novelty and diversity. We observe that, for larger n, both the diversity and novelty of the lists decrease (which means that the recommended items are more and more popular), except for the Jester dataset. The rationale behind these results is that, when the constraint n becomes more strict, more users are required to have seen (rated) those items, which, in the long term, produces that more popular items are being recommended. This behaviour is consistent for the three analysed datasets.

Now we analyse the decision-aware strategy based on prediction uncertainty. As we show in Table 2, this strategy evidences a strong tradeoff between coverage and precision, since introducing a threshold of 0.82 increases the performance by a factor of 4 but reduces the coverage a 70% with respect to no threshold. This situation is very interesting, because the optimal recommender depends on the evaluation metric: for instance, $\sigma_\tau = 0.84$ obtains the highest value for F_1 and $G_{2,1}$ but not in the other metrics, this is because they are too sensitive to the precision value, since that recommender achieves the second best value. This example evidences the differences between the proposed correctness metrics and the other combination metrics: whereas the latter simply combine two values, the former include further assumptions that, in principle, helps to interpret the comparison between recommenders. Following the previous example, $\sigma_\tau = 0.84$ is preferred over $\sigma_\tau = 0.86$ by F_1, but *UC* inverts this relation; we can infer that $\sigma_\tau = 0.86$ is better suited for deciding when an item should be recommended, since

[1] Availlable at https://grouplens.org/datasets/movielens/

[2] Available at http://eigentaste.berkeley.edu/dataset/

Table 1: Comparison of performance metrics at cutoff 10 when using a decision-aware strategy based on prediction support, for a nearest-neighbour recommender with $k = 10$ on ML-100K.

n	P	USC	ISC	F_1	F_2	$F_{0.5}$	$G_{1,1}$	$G_{1,2}$	$G_{2,1}$	UC	RUC	IC	RIC
1	0.037	**100.0**	**62.1**	0.070	0.159	0.045	0.191	0.332	0.110	0.037	0.037	0.000	0.015
2	0.133	100.0	46.9	0.234	0.433	0.160	0.364	0.510	0.260	0.133	0.133	0.002	0.021
3	0.188	100.0	39.5	0.317	0.537	0.225	0.434	0.573	0.329	0.189	0.189	0.002	0.026
4	0.230	100.0	35.1	0.374	0.599	0.272	0.480	0.613	0.376	0.234	0.236	0.003	0.029
5	**0.245**	99.7	32.3	**0.393**	**0.618**	**0.288**	**0.494**	**0.624**	**0.391**	**0.259**	**0.266**	**0.003**	**0.029**
6	0.241	96.4	28.5	0.386	0.603	0.284	0.482	0.607	0.383	0.257	0.263	0.003	0.026
7	0.237	85.9	24.8	0.371	0.563	0.277	0.451	0.559	0.364	0.231	0.231	0.002	0.023
8	0.226	66.9	21.7	0.338	0.480	0.260	0.389	0.466	0.324	0.180	0.171	0.002	0.018

Table 2: Comparison of performance metrics at cutoff 10 when using a decision-aware strategy based on prediction uncertainty, for a probabilistic matrix factorisation algorithm on ML-100K.

σ_τ	P	USC	ISC	F_1	F_2	$F_{0.5}$	$G_{1,1}$	$G_{1,2}$	$G_{2,1}$	UC	RUC	IC	RIC
–	0.093	**100.0**	22.7	0.170	0.338	0.113	0.304	0.453	0.205	0.093	0.093	0.001	0.009
0.82	**0.326**	28.2	9.1	0.303	0.290	**0.316**	0.303	0.296	0.311	0.100	0.094	0.001	0.006
0.84	0.283	59.0	15.1	**0.382**	0.484	0.316	0.408	0.462	**0.361**	0.174	0.170	0.002	0.011
0.86	0.214	80.9	19.6	0.338	**0.520**	0.251	**0.416**	0.519	0.333	**0.177**	**0.176**	**0.002**	0.012
0.88	0.181	95.6	22.2	0.304	0.514	0.216	0.415	**0.548**	0.315	0.176	0.176	0.002	**0.013**
0.90	0.165	99.5	24.8	0.283	0.495	0.198	0.405	0.546	0.300	0.165	0.165	0.002	0.013
0.92	0.156	100.0	26.0	0.269	0.480	0.187	0.395	0.538	0.289	0.156	0.156	0.002	0.012
0.94	0.145	100.0	27.3	0.254	0.459	0.175	0.381	0.526	0.276	0.145	0.145	0.002	0.011
0.96	0.139	100.0	28.2	0.245	0.447	0.168	0.373	0.518	0.269	0.139	0.139	0.002	0.011
0.98	0.133	100.0	**28.6**	0.235	0.435	0.161	0.365	0.511	0.261	0.133	0.133	0.002	0.011

we are rewarding unanswered recommendations above incorrect recommendations.

Finally, Figure 2 compares novelty and diversity evolution for different thresholds using a decision-aware strategy based on prediction uncertainty. We observe the same result as in the previous strategy (Figure 1): when the constraints are more strict (σ_τ decreases) both novelty and diversity decrease in the three tested datasets. In this case the rationale is slightly different to what happened before: those items with a lower standard deviation seem to correspond with popular items (like before), however, this constraint really imposes a limit on the number of different items that can be recommended, which ends up producing very low diversity scores.

5 CONCLUSIONS AND FUTURE WORK

In this work we have studied how to increase the user confidence on the system by making the system aware of the decisions taken. For this, we have proposed two strategies to decide if an item should be included in a recommendation list of a specific user, based on the consistency and reliability of the data that will be used by the recommender system to estimate the preferences. These strategies (one based on the support of the prediction and another on its uncertainty) have been evaluated in terms of precision, coverage, novelty, and diversity. We have shown that a balance between these evaluation dimensions – especially between precision and coverage – is critical, and different metrics have been studied to draw conclusions from them.

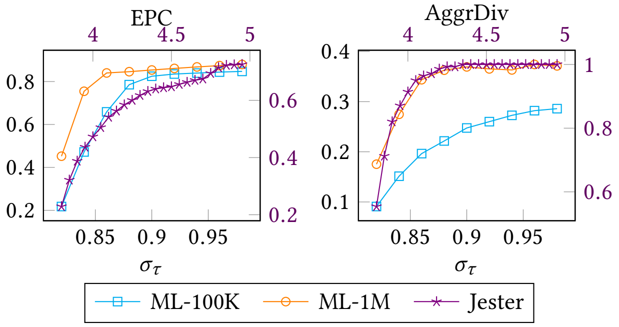

Figure 2: Novelty and diversity metrics at cutoff 10 for a decision-aware strategy based on prediction uncertainty. Jester is plotted in the secondary (right and upper) axis.

As a first step towards improving the understanding of this trade-off, we have proposed a family of metrics (correctness) based on the assumption that it is better to avoid a recommendation rather than providing a bad recommendation. However, further analysis is needed in the future, especially to find an objective way to discriminate between these systems and decide which of these metrics correlates better with the user satisfaction. We also aim at extending the correctness family of metrics so that other evaluation dimensions could be combined under the same framework: diversity, novelty, or even other accuracy metrics like nDCG. Additionally, the psychological aspect of the recommendations should also be considered, since if a user expects to receive N recommendations, she may decrease her confidence in the system if less than N recommendations are presented. Moreover, we aim at validating these results in an online setting with real users, in particular, how users value incorrect recommendations in comparison with unanswered/missing recommendations.

Acknowledgments

This work was funded by the national Spanish Government under project TIN2016-80630-P.

REFERENCES

[1] Gediminas Adomavicius, Sreeharsha Kamireddy, and Youngok Kwon. 2007. *Towards more confident recommendations: Improving recommender systems using filtering approach based on rating variance.* Social Science Research Network, 152–157.

[2] Alejandro Bellogin, Pablo Castells, and Ivan Cantador. 2011. Precision-oriented evaluation of recommender systems: an algorithmic comparison. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 333–336.

[3] Asela Gunawardana and Guy Shani. 2015. Evaluating Recommender Systems. In *Recommender Systems Handbook*. Springer, 265–308.

[4] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. 2002. An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms. *Inf. Retr.* 5, 4 (2002), 287–310.

[5] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.* 22, 1 (2004), 5–53. DOI: http://dx.doi.org/10.1145/963770.963772

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *ICDM*. IEEE Computer Society, 263–272.

[7] Yew Jin Lim and Yee Whye Teh. 2007. Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD cup and workshop*, Vol. 7. Citeseer, 15–21.

[8] Sean M. McNee, Shyong K. Lam, Catherine Guetzlaff, Joseph A. Konstan, and John Riedl. 2003. Confidence Displays and Training in Recommender Systems. In *INTERACT*. IOS Press.

[9] John O'Donovan and Barry Smyth. 2005. Trust in recommender systems. In *IUI*. ACM, 167–174.

[10] Anselmo Peñas and Álvaro Rodrigo. 2011. A Simple Measure to Assess Non-response. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (Eds.). The Association for Computer Linguistics, 1415–1424. http://www.aclweb.org/anthology/P11-1142

[11] Alan Said and Alejandro Bellogín. 2014. Rival: a toolkit to foster reproducibility in recommender system evaluation. In *RecSys*. ACM, 371–372.

[12] Barry Smyth, David C. Wilson, and Derry O'Sullivan. 2002. Data Mining Support for Case-Based Collaborative Recommendation. In *AICS (Lecture Notes in Computer Science)*, Vol. 2464. Springer, 111–118.

[13] Saúl Vargas. 2015. *Novelty and diversity enhancement and evaluation in Recommender Systems*. Ph.D. Dissertation. Universidad Autónoma de Madrid.

[14] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 109–116.