Journal of Universal Computer Science, vol. 21, no. 13 (2015), 1849-1868
submitted: 4/5/15, accepted: 6/7/15, appeared: 28/12/15 © J.UCS

Evaluating the Relative Performance of Collaborative Filtering Recommender Systems

Humberto Jesús Corona Pampín, Houssem Jerbi and Michael P. O'Mahony
(Insight Centre for Data Analytics [see 1]
School of Computer Science, University College Dublin, Ireland
firstname.lastname@insight-centre.org)

Abstract: Past work on the evaluation of recommender systems indicates that collaborative filtering algorithms are accurate and suitable for the top-N recommendation task. Further, the importance of performance beyond accuracy has been recognised in the literature. Here, we present an evaluation framework based on a set of accuracy and beyond accuracy metrics, including a novel metric that captures the *uniqueness* of a recommendation list. We perform an in-depth evaluation of three well-known collaborative filtering algorithms using three datasets. The results show that the user-based and item-based collaborative filtering algorithms have a high inverse correlation between popularity and diversity and recommend a common set of items at large neighbourhood sizes. The study also finds that the matrix factorisation approach leads to more accurate and diverse recommendations, while being less biased toward popularity.

Key Words: Recommender Systems, Collaborative Filtering, Matrix Factorisation, Evaluation, Accuracy, Beyond Accuracy, Uniqueness.
Category: H.3.3, H.4, M.5

1 Introduction

Nowadays, it is becoming increasingly difficult to find relevant information across the large catalogs available on the web. For example, finding a new movie to watch, discovering new bands to listen to, or deciding whom to follow in online social networks is no longer a straightforward task. Recommender systems address this problem by automatically selecting the most relevant content based on user preferences. They are used to personalise the user experience in different scenarios, such as online shopping [Linden et al. 2003] and music discovery [Celma and Herrera 2008].

The vast amount of research in the recommender systems area has led to the development of a variety of recommendation algorithms. *Content-based* [Pazzani and Billsus 2007] and *case-based* [Smyth 2007] approaches rely on item properties and domain knowledge to recommend items similar to those the user liked in the past, while *collaborative filtering* approaches exploit user preference data to generate recommendations. Collaborative filtering approaches include

[1] The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289.

neighbourhood-based models, which make recommendations based on the similarities across users [Herlocker et al. 1999] or items [Karypis 2001, Sarwar et al. 2001], and latent factor models, where ratings are inferred based on a smaller number of latent factors [Koren 2010]. These approaches can be used to generate a rating prediction for a particular user-item pair (*rating prediction task*), or to present the user with the most relevant items (*top-N recommendation task*). In this work, we focus on the performance evaluation of three variants of collaborative filtering algorithms, which are known to be accurate [Schelter et al. 2013, Shi et al. 2012] and are widely used in commercial systems such as Amazon.com [Linden et al. 2003] and the Xbox gaming platform [Koenigstein et al. 2012].

Typically, there is no perfect recommender system solution for all business requirements and use cases. The suitability of a given recommender system depends on the company priorities and goals, as well as the type and characteristics of the data available. This requires the identification of a set of metrics to assess the quality of recommender systems. Initially, recommender systems evaluation focused on prediction accuracy, typically using error metrics [Shani and Gunawardana 2011] such as root mean squared error (*RMSE*) and mean absolute error (*MAE*), and the quality of the top-N recommendations using accuracy metrics [Gunawardana and Shani 2009], such as precision and recall. However, recommender systems have a variety of properties that may affect the user experience, such as the novelty and diversity [Vargas and Castells 2011] of recommended products. Recommender systems evaluation has been extensively studied in recent years, including surveys on the field [Herlocker et al. 2004, Shani and Gunawardana 2011]. However, there are still many open questions, and it is yet not clear how different properties such as the popularity bias and coverage of recommendations affect their performance and the user experience.

In this article, we identify a set of performance metrics that cover different properties of recommender systems. In particular, we define a novel metric, named *uniqueness*, which measures the difference between two recommendation lists. Based on these metrics, we present an in-depth analysis of three variants of collaborative filtering algorithms, focusing on the relationships and tradeoffs between these metrics. We aim to better understand how recommendations are generated in the top-N recommendation task, to highlight the weaknesses and strengths of the different algorithms considered, and to understand how these algorithms can be used to enhance content discovery in different scenarios.

This article is organised as follows. Section 2 discusses the main trends in recommender systems evaluation. Section 3 describes the evaluation metrics used in our study and section 4 presents the evaluated algorithms. Section 5 presents an in-depth evaluation of the considered algorithms. Later, in Section 6, we focus on the evaluation of the relative performance of the neighbourhood-based algorithms. Finally, in Section 7 we discuss conclusions and future work.

2 Related Work

Recent work has highlighted the key research challenges in the recommender systems field, focusing on the evaluation of the user experience as a whole, rather than treating the algorithm as a separate entity [Konstan and Riedl 2012]. Developing a common research infrastructure for evaluation, and the need for an overall performance evaluation, which is not limited to accuracy, are already well established problems [Fleder and Hosanagar 2009, McNee et al. 2006]. In this context, new metrics to measure different properties of the recommender systems have been proposed, together with experimental methodologies to evaluate them. In this section, we present a review of work which focuses on performance evaluation from both an *accuracy* and *beyond accuracy* perspective.

2.1 Accuracy

The ability of a recommender system to generate relevant recommendations is obviously a key factor of performance. As a consequence, accuracy metrics have been extensively used for recommender systems evaluation. Prediction accuracy metrics are better suited for the *rating prediction task*, whereas top-N accuracy metrics are more appropriate in the context of *top-N recommendation task*.

The relationship between prediction accuracy metrics (such as *RMSE*) and top-N accuracy metrics is studied in [Cremonesi et al. 2010]. The authors explain how prediction accuracy metrics are not appropriate to evaluate recommender systems in the *top-N recommendation task*, which was also pointed out in [Gunawardana and Shani 2009, Steck 2010]. The authors also explain that improvements in algorithms as measured by *RMSE* do not translate into improvements in top-N accuracy metrics.

Bellogin et al. highlighted that top-N accuracy metrics, such as precision, recall and normalised discounted cumulative gain (*NDCG*), are often applied in different experimental configurations, which makes it difficult to compare works, even when the same metrics are used [Bellogin et al. 2011]. The *top-N recommendation* problem is also studied in [Pradel et al. 2012], where an analysis of the popularity effect and the rating distribution of missing ratings is considered. The study shows that, just as with known ratings, missing ratings can be modelled as being non-random. The study concludes that ignoring non-rated items biases the evaluation showing very positive results, while considering them as negative biases the evaluation in favour of algorithms that exploit popularity.

A study of neighbourhood models is presented in [Rafter et al. 2009], which focuses on the influence of neighbour selection on the performance of user-based (*UKNN*) and item-based (*IKNN*) collaborative filtering algorithms. The authors explain how algorithm performance is biased by the evaluation methodology and the distribution of ratings in widely used evaluation datasets. For example, they

show that neighbourhood selection has a limited effect on performance, and that the contributions of the neighbours are often disguised or misleading.

Meyer et al. proposed to combine new metrics, such as the *average measure of impact*, with accuracy metrics [Meyer et al. 2012]. They found no clear correlation between prediction accuracy metrics and the quality of the recommendations. Moreover, they also analysed the effect on recommendations based on a user segmentation that takes in account the long tail distribution of items and user ratings, showing that performance is closely related to these segments.

In this work we perform an evaluation of recommender systems in the *top-N recommendation task*. We use top-N accuracy metrics to evaluate the accuracy of the recommendations.

2.2 Beyond Accuracy

Generating diverse recommendations is important to enhance the user experience and enable content discovery [Corona Pampín et al. 2014]. Thus, several works have studied diversity in the context of recommender systems. For example, Said et al. use a furthest neighbour model to increase diversity, generating almost orthogonal recommendations compared to the traditional (nearest neighbour) *UKNN* approach [Said et al. 2012]. Hurley introduces diversity within the optimisation function in a matrix factorisation approach, obtaining more diverse recommendations without the need to post-filter the recommendation lists [Hurley 2013]. Zhang and Hurley also investigated the maximisation of diversity of the recommendation list, while maintaining adequate similarity to the user profile [Zhang and Hurley 2008]. The results show improvements in terms of accuracy and a newly proposed diversity metric.

Diversity in music recommendation is analysed in [Jawaheer et al. 2010], where it is argued that optimising recommendations for prediction accuracy metrics such as *RMSE* might lead to less diverse recommendations. The relationship between popularity and sales diversity is studied in [Fleder and Hosanagar 2009]. The authors conclude that some well-known collaborative filtering algorithms can lead to a reduction in sales diversity, since they recommend popular products (based on previous purchases and ratings). The authors claim that these recommenders can create a *rich-get-richer* effect for popular products.

A framework for evaluating both diversity and novelty is introduced in [Vargas and Castells 2011]. The work is motivated by the lack of agreement when evaluating these properties — different studies use different metrics, the relationship between them is not studied and some of them have flaws. The work studies novelty from two different perspectives: popularity and distance. More specifically, it considers ranking and relevance when measuring novelty in recommendations, giving a more complete picture of the system. The analysis of recommendation novelty was also the focus in [Celma and Herrera 2008], alongside

with popularity. Results show that content-based recommender systems (applied to the music domain) are not biased toward popularity, while a standard collaborative approach is perceived as providing higher quality recommendations even when the recommendations are not as novel. Further studies of accuracy-based metrics along with novelty and diversity are presented in [Vargas 2015].

Other metrics of interest have have also considered. For example, metrics are proposed to evaluate the usefulness of recommended items in [Ge et al. 2010]. The authors argue that these metrics correlate with real user experience better than previous metrics. Coverage is also evaluated in [Adamopoulos and Tuzhilin 2011], where a new recommender system based on the utility theory of economics is presented to produce high quality and unexpected recommendations and better coverage, when compared to standard collaborative filtering algorithms.

The related work described above clearly highlights that various metrics have been studied, often in isolation, to evaluate the different properties of recommender systems. In this work, we present an evaluation framework where properties such as recommendation accuracy, diversity, popularity and coverage are studied together to better understand their relationships and tradeoffs. In order to gain additional insights in to the performance of recommendation algorithms, we also introduce the concept of the *uniqueness* of recommendation lists generated by different algorithms, which is described in the next section.

3 Evaluation Metrics for Recommender Systems

In this section we describe the metrics we have used in our evaluation. In Section 3.1, we describe the top-N accuracy metrics. Then, we introduce in Section 3.2 other metrics that evaluate properties beyond accuracy.

3.1 Top-N Accuracy Metrics

The following metrics, adapted from the field of information retrieval, are used to evaluate accuracy in the *top-N recommendation task*.

- **Recall** (RCL) measures the proportion of relevant items which are recommended (Equation 1), while **Precision** (PRC) measures the proportion of relevant items from those recommended (Equation 2).

$$RCL_u = \frac{|I_u^+ \cap I_u^R|}{|I_u^+|}\,, \qquad (1) \qquad\qquad PRC_u = \frac{|I_u^+ \cap I_u^R|}{|I_u^R|}\,, \qquad (2)$$

  where I_u^+ are the items in the test set liked by user u, and I_u^R are the recommendations made for user u. These metrics are widely used in recommender systems evaluation but have one major drawback: a system that retrieves all items will have perfect recall, but poor precision.

– **F-1 measure** (Equation 3) solves the above problem by combining precision and recall, providing a single score for the relevance of items recommended in the list.

$$\text{F-1}_u = 2 \times \frac{PRC_u \times RCL_u}{PRC_u + RCL_u} \,. \tag{3}$$

3.2 Beyond Accuracy Metrics

In addition to accuracy metrics, previously studied properties such as diversity, popularity, and catalog coverage are considered [Herlocker et al. 2004, Shani and Gunawardana 2011]. Moreover, we also introduce the concept of *uniqueness* and present a novel definition of per-user item coverage.

– **Diversity** (DIV) [Smyth 2007] captures how different the recommended items are from each other. It is evaluated based on a pairwise comparison of the items in the recommendation list (see Equation 4). This metric relies on item co-ratings to calculate the cosine similarity (*Sim*) between items. Generating diverse recommendations is important to enhance user experience. For example, recommending all five *Harry Potter* movies to a user who liked *Harry Potter and the Philosopher's Stone* might be accurate but of little use due to their obvious nature and the lack of diversity involved.

$$Diversity(I_u^R) = \frac{1}{|I_u^R|\big(|I_u^R|-1\big)} \sum_{\forall i \in I_u^R} \sum_{\forall j \in I_u^R, i \neq j} \big(1 - Sim(i,j)\big) \,. \tag{4}$$

– **Popularity** (POP) is defined as the total number of ratings for item i across all users U, $r_{i,U}$, divided by the total number of ratings for all items in the system, $r_{I,U}$ (Equation 5). Popularity plays a major role in retail, where traditionally around 80% of purchases come from the top 20% of items, while the remaining items are in the so-called long tail [Anderson 2004], an area of the market space largely unexplored in the hit-driven, physical retail world. However, digital retailers are increasingly exploiting consumer desire for more personalised and niche products and the ability of recommender systems to enable users to discover such items is key. Thus, popularity is an important metric, measuring the capability of recommender systems to suggest less popular items from the long tail.

$$POP_i = \frac{r_{i,U}}{r_{I,U}} \,. \tag{5}$$

– **Per-user Item Coverage** (UICov) (Equation 6) measures, for each user u, the proportion of items, across all items in the system I, which are candidates for recommendation $C_u \in I$. For example, only items contained in the user neighbourhood can be recommended by the *UKNN* algorithm. This property

is very important in e-commerce applications, where ideally every (relevant) item should be recommendable to users, to avoid filter-bubble like effects and reduced user satisfaction caused by limited recommendation scenarios.

$$UICov_u = \frac{|C_u|}{|I|} \,. \tag{6}$$

– **Catalog Coverage** (CCov) [Ge et al. 2010] measures the proportion of items I which ever get recommended (Equation 7). Although a recommender system can have perfect per-user item coverage, only a small subset of items are recommended in the top-N recommendation task; this metric measures the size of that subset over all users for a given recommendation algorithm. Thus, while the per-user item coverage metric captures the *potential* for items to be recommended, this metric measures the proportion of items which are actually suggested to users by standard recommender system algorithms.

$$CCov = \frac{1}{|I|} \Big| \bigcup_{\forall u \in U} I_u^R \Big| \,. \tag{7}$$

– **Uniqueness** (Equation 8) measures the ability of a recommendation algorithm to generate recommendations which are not generated by other algorithms. For example, given two recommendation lists, $I_u^{R_a}, I_u^{R_b}$, the uniqueness of $I_u^{R_a}$ is calculated as the cardinality of the difference with $I_u^{R_b}$, that is, the number of items of $I_u^{R_a}$ that are not elements of $I_u^{R_b}$. Measuring uniqueness, in conjunction with other metrics, provides additional insights in to the performance of recommendation algorithms. For example, if the recommendations generated by one algorithm are unique with respect to another while both have similar accuracy, this indicates that the algorithms have identified different sets of relevant items. This metric could, for example, be used to inform ensemble approaches to recommendation that combine the output of different algorithms to improve recommendation quality for the end user.

$$Uniqueness(I_u^{R_a}) = |I_u^{R_a} \setminus I_u^{R_b}| \,. \tag{8}$$

While other properties beyond accuracy, such as *novelty*, *serendipity* and *robustness* [O'Mahony et al. 2004], are also important when evaluating recommender system performance, a study of these properties is left for future work.

4 Algorithms

Using the above metrics, we evaluate three recommender systems algorithms. The neighbourhood-based algorithms, user-based and item-based collaborative filtering, leverage the ratings matrix directly to find similar users or items in

order to make recommendations. In contrast, latent factor models first factorise the ratings matrix and then infer item ratings for the target user based on a smaller number of latent factors. Here, we use the *weighted matrix factorisation* algorithm (*WMF*) as a representative latent factor approach since it provides good performance on unary rating datasets [Ning and Karypis 2011]. We note that other approaches, such as one-class collaborative filtering [Pan et al. 2008], have also been successfully applied to unary rating datasets; given limitations of space, such work is not considered here. In our study, we consider the top-N recommendation task, which is the typical form of recommender system output.

The **user-based collaborative filtering** (*UKNN*) [Herlocker et al. 1999] algorithm first identifies the most similar users (i.e. neighbours) to the target user using cosine similarity. Secondly, the items not already rated by the target user are ranked based on the aggregated similarity of the neighbours, where ties are broken randomly. Specifically, the score of a candidate item is an aggregation of the pairwise similarities between the target user and each neighbour who rated that item. Finally, the top-N ranked items are returned as recommendations.

The **item-based collaborative filtering** (*IKNN*) [Karypis 2001] algorithm first finds the similar items (i.e. neighbours) for each item rated by the target user, and then ranks these items according to an aggregation of their pairwise similarities with those rated by the target user. The top-N ranked items form the recommendation list.

Matrix Factorisation methods are known to outperform neighbourhood models in terms of accuracy [Campos et al. 2011, Ning and Karypis 2011] and scalability [Koenigstein et al. 2012], as they operate in a reduced space of latent features [Koren 2010]. Here, we use the *WMF* algorithm with a fast learning method proposed in [Hu et al. 2008] and a global parameter optimisation to give observed values higher weights.

5 A Bias Analysis

We present a comprehensive evaluation of the above algorithms using three different datasets. We also consider a naïve non-personalised benchmark algorithm (*Most Popular*) which presents each user with the most popular (given by the number of ratings in the training set) items unknown to her. Our aim is to investigate the relationships between the various properties, datasets and algorithms, paying special attention to the popularity bias of the different algorithms.

5.1 Datasets

Our experiments were performed using datasets with unary rating scales [Karypis 2001, Ning and Karypis 2011]. We leave to future work an analysis of datasets with multi-value rating scales. Two publicly available datasets are used, while

Dataset	# users	# items	# ratings	Mean (std. dev.) ratings per user	Mean (std. dev.) ratings per item	Sparsity
FB	1,428	5,846	64,612	45 (49)	11 (26)	0.9923
LastFM	1,864	6,945	82,037	44 (7)	12 (32)	0.9937
ML	2,040	7,459	374,352	183 (187)	50 (110)	0.9754

Table 1: Summary statistics for the datasets after pre-processing.

the third dataset is collected using the Facebook graph API. Each dataset was pre-processed such that only users with at least 12 unary ratings and items with at least 5 tags were considered (i.e. to remove users and items with few ratings). Table 1 shows the statistics for these datasets after pre-processing.

- The **Facebook** (FB) dataset was collected using a custom web application. It records the list of liked items by each user of the application, as well as those liked by her friends. This dataset contains 1,428 users, 5,846 items and 64,612 unary ratings (i.e. like actions of Facebook *musician/band* pages).
- **LastFM-hetrec** (LastFM) is a music dataset released for the *Hetrec2011 Workshop* [Cantador et al. 2011]. The original dataset contains 92,834 user-artist listening interactions (unary ratings) for 1,892 users and 17,632 artists.
- **MovieLens-hetrec** (ML) is a sampled version of the MovieLens dataset [Herlocker et al. 1999], which was expanded with additional metadata and also released for the *Hetrec2011 Workshop*. The original dataset contains 2,113 users, 10,197 items, and 855,598 ratings on a 5 point scale. Since Facebook and LastFM are unary datasets, we also converted this dataset to unary ratings, where only ratings of 4 and 5 are considered as positive preferences and included in the user-item matrix. This particular rating threshold has been used in previous work with similar datasets [Bellogin et al. 2011].

These datasets have a similar number of users (between 1,428 and 2,040) and items (between 5,846 and 7,459) but with different levels of sparsity. For example, the MovieLens dataset contains 374,352 ratings, while the FB dataset only contains 64,612 ratings. Thus, we expect to see differences in the performance of the algorithms between the datasets in some of the properties evaluated.

5.2 Methodology

For each dataset, 80% of ratings (randomly selected) are included in the training set, and the remaining 20% in the test set. We determine the optimal (with respect to accuracy) parameters for each algorithm by running an internal 5-fold cross-validation on the training set data as per [Trohidis et al. 2008]. Recommendation accuracy for *IKNN* was seen to increase with neighbourhood size

	Algorithm	Pop	CCov (%)	UICov (%)	DIV	PRC	RCL	F-1
FB	Most Popular	0.500	0.684	98.957*	0.706*	0.066	0.089	0.076
	UKNN (**60**)	0.310	5.132	16.049	0.711	0.136	0.181	0.156*
	IKNN (**300**)	0.251*	27.386	40.478	0.672*	0.132	0.182	0.153*
	WMF (**20,20**)	0.254*	7.030	98.957*	0.747	0.155	0.202	0.176
LastFM	Most Popular	0.507	0.374	98.675*	0.654	0.068	0.073	0.070
	UKNN (**50**)	0.286	7.790	9.709	0.730	0.167	0.183	0.175*
	IKNN (**300**)	0.239	30.194	38.815	0.714	0.180	0.201	0.190^{+}
	WMF (**20,50**)	0.234	5.37	98.675*	0.788	0.180	0.196	0.188^{*+}
ML	Most Popular	0.282	0.724	99.464*	0.490	0.221	0.082	0.120
	UKNN (**140**)	0.104	1.823	46.130	0.519	0.294	0.110	0.160*
	IKNN (**300**)	0.095	3.365	50.611	0.527	0.284	0.106	0.154*
	WMF (**25,40**)	0.079	8.861	99.464*	0.603	0.344	0.133	0.191

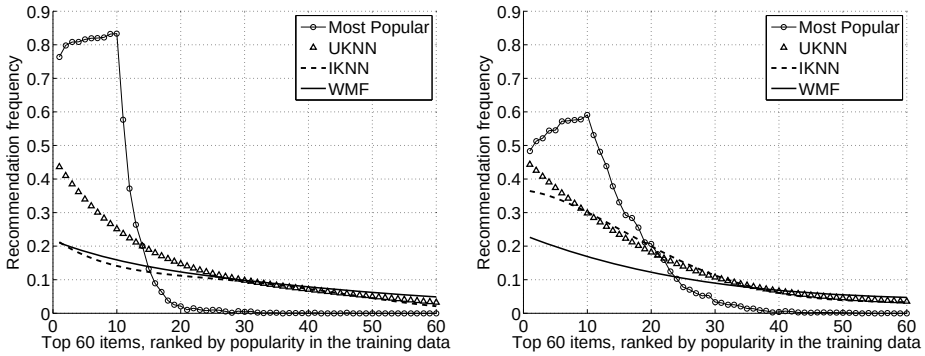
Table 2: Comparison of the performance of the recommendation algorithms. Bold numbers indicate optimal algorithm parameter values (neighbourhood size for UKNN and IKNN, number of factors and number of iterations for WMF). Pairs of non statistically significant results are annotated with the symbols * or $^{+}$.

(up to $k = 300$, beyond which improvements in accuracy were minor) for all datasets, which aligns with previous work performed on datasets with unary ratings [Karypis 2001]. Thus, we set the neighbourhood size for *IKNN* to $k = 300$. In the case of the *UKNN* algorithm, the maximum accuracy is achieved at lower (and different) neighbourhood sizes for each dataset; likewise the optimal number of factors and iterations for *WMF* varies for each dataset (see Table 2 for these parameter values). Once the optimal parameters for the various algorithms are determined for each dataset, top-10 recommendations are made using the full training set and compared to the test set items for each user.

5.3 Results

Here, we evaluate the algorithms described in Section 4 using the datasets described in Section 5.1. The recommendations are evaluated both in terms of top-N accuracy metrics (Section 3.1) and beyond accuracy metrics (Section 3.2).

For each dataset, we perform Kruskal-Wallis tests on popularity, per-user item coverage, diversity and accuracy (F-1) results, and post-hoc Tukey-Kramer tests to determine which pairs of algorithms produce significantly different results. Significance testing is performed at the .05 level. We do not perform a statistical test on catalog coverage, as it is given by a single number for each algorithm and dataset. As statistically significant differences are found in the majority of cases, we highlight those pairs of algorithms where no significant differences are seen in Table 2. For example, in the Facebook dataset, there is no significant difference in popularity between *IKNN* and *WMF*, whereas the differences between all other algorithm pairs are statistically significant.



(a) Facebook dataset (b) MovieLens dataset

Figure 1: Recommendation frequency of the 60 most popular items. For clarity, *UKNN*, *IKNN* and *WMF* are approximated by a 5-degree polynomial function.

5.3.1 Popularity Bias

Figure 1 shows the recommendation frequency of the top-60 most popular items in the Facebook and MovieLens datasets (results for the LastFM dataset are omitted since similar trends are seen for this dataset and Facebook), i.e. how often such items are found in the top-10 recommendations made by each algorithm for each user. The vertical axis shows the recommendation frequency (normalised by the total number of users in the system), and the horizontal axis shows the 60 most popular items in each dataset. For example, item 1 is the most frequently rated item in each dataset.

In general, the results show that the most popular items are most frequently recommended by the algorithms. As expected, the popularity of the benchmark algorithm is significantly higher compared to other algorithms, and is seen to increase up to the 10^{th} most popular item, followed by a sharp decrease (this effect is an artefact of the experimental methodology, in which each user is presented with top-10 recommendations, and users are never recommended items which are contained in their training set). After the benchmark algorithm, *UKNN* is the most biased toward making popular recommendations, while the popularity trends of the *IKNN* and *WMF* algorithms are different across datasets.

Table 2 shows the average popularity values per algorithm for each dataset, as per Equation 5. Here, the popularity of recommendations made for each user are calculated as the average of the popularity for the top-10 recommendations. The results correlate with those shown in Figure 1, highlighting that the *UKNN* algorithm is the most biased toward making popular recommendations compared to *IKNN* and *WMF* across all datasets (these findings are statistically significant at .05 level). In the Facebook and LastFM datasets, the *IKNN* and *WMF* algorithms are less biased towards popularity and their performances are

comparable. For example, the average popularity for *WMF* is 18% less than for the *UKNN* algorithm in the Facebook dataset.

Slightly different trends are found for the MovieLens dataset, which is over three times more dense compared to the Facebook and LastFM datasets. Here, the average popularity for both the *UKNN* and *IKNN* algorithms is high — 32% and 20% above that seen for the *WMF* algorithm, respectively.

In summary, the *WMF* algorithm performed best in two of the three datasets evaluated, and in all cases it was less biased toward making popular recommendations than the *UKNN* algorithm. Hence, for this evaluation criteria (popularity bias), *WMF* is found to be the best performing approach.

5.3.2 Other Properties

Here, we describe the results shown in Table 2 according to the rest of metrics presented in Section 3. Precision, recall, F-1 and diversity, are related to the list of recommendations presented to each user. However, catalog coverage relates to the recommendation candidates (e.g., in the *UKNN* algorithm, only items rated by the user neighbours are considered as recommendation candidates), and per-user item coverage relates to all the items recommended by the system.

The results for *top-N accuracy metrics* (precision, recall and F-1) show that, at the .05 level of significance, *WMF* performs best in terms of F-1 for the Facebook and MovieLens datasets, while the accuracy of the *UKNN* and *IKNN* algorithms are similar. For example, *WMF* outperforms *UKNN* and *IKNN* by 19% and 24% (F-1) in the MovieLens dataset. In contrast, similar F-1 performance is seen across the three algorithms in the case of the LastFM dataset.

Per-user item coverage (UICov, Equation 6), measures the potential of an algorithm to select recommendation candidates. Here, the results show that the *WMF* algorithm considers almost every item as a candidate (UICov $> 98\%$). The *UKNN* algorithm performs poorly in this regard, since by definition, only items which are in the user's neighbourhood can be considered as recommendation candidates. *IKNN* was seen to outperform *UKNN* in all datasets.

Catalog coverage (CCov, Equation 7) measures the proportion of items that are recommended out of the total set of items. Here, the *IKNN* algorithm, which exploits item similarity, performs significantly better than the other algorithms (in the Facebook and LastFM datasets), covering up to 30% of the item catalog in its recommendations across users and up to 6 times more items than the *UKNN* and *WMF* algorithms. In terms of *diversity* (Equation 4), the *WMF* algorithm performs clearly better than the rest, with a performance around 9% higher on average than the best neighbourhood-based approach, showing a gap in performance between this approach and the neighbourhood based algorithms.

The analysis of popularity shows that the *UKNN* approach is inherently biased toward making popular recommendations. The results also show that

the *WMF* approach is less biased toward making popular recommendations, while still generating more diverse and accurate recommendations than the rest of the evaluated algorithms (in two of the three datasets evaluated). It is also noteworthy that the *IKNN* algorithm has much higher catalog coverage than all the other evaluated alternatives, recommending a much larger subset of items, at least for two of the three datasets evaluated. As expected, the coverage of the *UKNN* algorithm is poor in all the datasets evaluated, as the recommendations are limited to the items seen in the user neighbourhood.

Finally, some of the results are not consistent through the different datasets. All the properties (with the exemption of diversity and per-user item coverage) show a different order in the best performing approach in one of the three datasets. This highlights the necessity of performing the evaluation in different datasets with different statistical properties. For example, in the MovieLens dataset, which is three times more dense than the Facebook and LastFM datasets, the catalog coverage of the *IKNN* algorithm is ~ 10 times smaller than for the LastFM and Facebook datasets. Hence, similar analyses should be performed for each particular domain under consideration, and the algorithm that delivers optimal performance across the metrics of interest, should be selected.

In this section, we presented interesting findings in terms of the correlation between the evaluated metrics, the algorithms and the datasets. In the next section, we focus on the evaluation of the two neighbourhood-based algorithms, while further studying the diversity and popularity of recommendations with respect to the number of neighbours. Furthermore, the results will also be analysed in terms of uniqueness of the recommendation lists.

6 A Comparative Analysis

Although neighbourhood-based models (Section 4) have been extensively evaluated, it is not completely clear how different factors, such as neighbourhood selection and item popularity bias, affect the overall performance of these algorithms. Here, we study the effect of neighbourhood selection for the *UKNN* algorithm, which relies on finding similar users to generate recommendations. More specifically, we experiment with different values for the neighbourhood size (k) and compare the results with the *IKNN* approach.

To explore how the neighbourhood size affects the performance of these algorithms, a study of recommendation accuracy, diversity and popularity is performed, comparing the results for different values of k. Moreover, we study the uniqueness of a recommendation list (introduced in Section 3.2), and evaluate the accuracy of the common and unique set of recommended items. This analysis allows us to understand the tradeoffs between the different evaluated properties and the values of this parameter.

6.1 Dataset

To evaluate the proposed approach, we select the **MovieLens-1M** dataset [Herlocker et al. 1999] (ml-1M), which contains circa 1 million ratings from 4,335 users and 3,561 items. Only users with at least 12 liked items are included in our evaluation. All items which received a rating of at least 4 (out of 5) are considered liked and are added to our unary dataset. The resulting dataset has an average of 211 ratings per user (std. dev. = 208), and 257 ratings per item (std. dev. = 347). Finally, the sparsity of the dataset is equal to 0.940.

In our previous work [Corona Pampín et al. 2014], we also performed an evaluation in the **MovieLens-100K** dataset. However, as the results obtained were very similar, those are excluded from the analysis presented here.

6.2 Methodology

In previous work, the motivation for considering neighbourhood size in *IKNN* was that of complexity rather than accuracy, and the accuracy of *IKNN* generally grows with the number of neighbours [Karypis 2001]. However, this parameter is known to effect the accuracy of *UKNN* in a different way, where accuracy is seen to decline at larger neighbourhood sizes [Herlocker et al. 1999]. Given that accuracy is a key consideration from a recommendation perspective, here we focus on evaluating neighbourhood size for the *UKNN* algorithm only, and leave an evaluation of neighbourhood size for *IKNN* to future work. Thus, we fix the neighbourhood size for the *IKNN* algorithm to $k = 300$ (i.e. optimised for accuracy, based on a cross-validation approach using training data as per Section 5.2), while we vary the number of neighbours for *UKNN* ($k \in [10, 200]$).

The test set for each user is created using 10 randomly selected liked items. Hence, each user's test set has the same number of items, all of which are liked. The training set is created using all items which are not included in the test set.

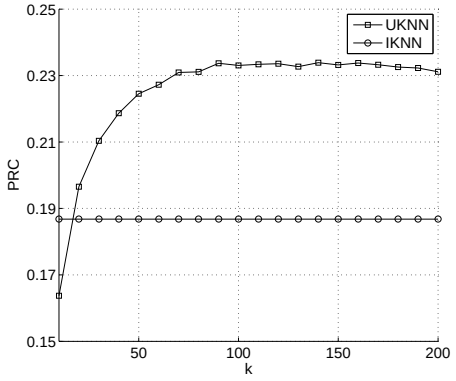
6.3 Results

In this section we present the results of the evaluation of the *UKNN* and *IKNN* algorithms. Recommendations are evaluated in terms of precision, popularity, diversity and uniqueness, and performance is compared as the neighbourhood size (k) is varied in the *UKNN* approach. For each user, the recommendations made by each algorithm are based on training data only and evaluated on that user's test set items. Since each user's test set contains exactly 10 liked items, and top-10 recommendations are made, the values of precision, recall and F-1 are equivalent. To begin, precision results are first discussed followed by the other properties, focusing on the relationship between each metric and precision.

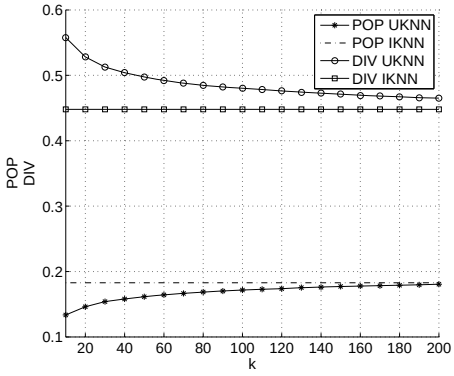
Figure 2a shows that the overall *precision* of *UKNN* increases up to $k \approx 90$, and thereafter remains relatively constant. It is noteworthy that the precision of *UKNN* exceeds that of *IKNN* at much smaller neighbourhood sizes (at $k = 15$).

The results for recommendation *popularity* and *diversity* are shown in Figure 2b. The results indicate that the average popularity of recommended items increases with k. Thus, as expected, there is a high bias toward popularity at larger sizes of k, and a corresponding decrease in recommendation diversity. In the limit, if every user was considered a neighbour, the recommendations would resemble the distribution of item ratings in the dataset. As a consequence, only the most popular items would be recommended.

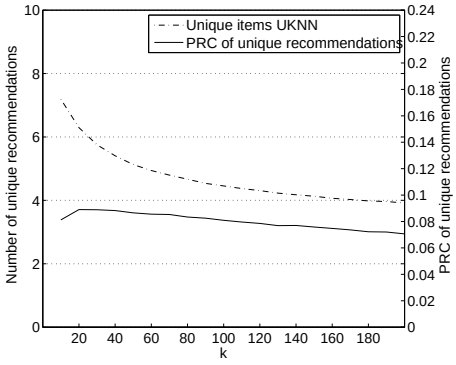
The average recommendation *diversity* decreases as k increases for the *UKNN* algorithm, and approaches that of the *IKNN* algorithm at $k = 200$. However, this reduction in diversity for *UKNN* is not compensated by a corresponding gain in precision at larger neighbourhood sizes. As previously described, the *UKNN*



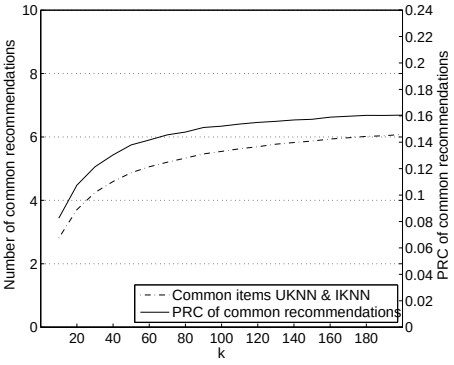
(a) Precision vs. neighbourhood size.



(b) POP and DIV vs. neighbourhood size.



(c) Number and precision of unique *UKNN* recommendations vs. neighbourhood size.



(d) Number and precision of common recommended items vs. neighbourhood size.

Figure 2: Performance vs. neighbourhood size in the ml-1M dataset.

algorithm attains its maximum accuracy at $k \approx 90$, yet the loss in diversity is a monotonically decreasing function. We can conclude that there is a high inverse correlation between popularity and diversity, and both suffer at larger values of k; i.e. recommendation lists contain mainly popular items which have a high degree of similarity to each other.

Finally, Figures 2c and 2d depict the average numbers of recommendations which are *unique* to the *UKNN* algorithm and those which are *common* to both algorithms, and the corresponding precision results over the unique and common recommended items. We can observe that the number of unique recommendations decreases significantly as k increases. For example, at $k = 100$, on average more than 5 of the 10 recommended items produced by both algorithms are the same (such a finding is useful in the context of ensemble recommenders, for example, where algorithms that produce many of the same recommendations can be identified). It is also apparent that recommendation accuracy is largely due to the common items which are recommended by both algorithms. For example, at $k = 100$, the precision of unique and common items is approximately 0.08 and 0.14, respectively. However, it should be noted that the unique items recommended by the *UKNN* algorithm are not necessarily irrelevant. Although these items are not present in the test set (which is used as the ground truth), they may still represent useful recommendations to users. Moreover, at smaller values of k, items recommended by *UKNN* are less popular, and such items by definition are less likely to appear in test sets. This is a well-known limitation of the standard 'offline' approach to evaluate the precision of recommender systems, and it can only be addressed by live user trials which is left to future work.

In summary, the results indicate the bias of the *UKNN* algorithm toward popular recommendations at larger neighbourhood sizes. However, the loss of diversity and bias toward popular recommendations at larger values of k is not compensated by a gain in precision. From the results, we also observe that the uniqueness of recommendations also depends on k, and that smaller neighbourhood sizes lead to more unique, less popular, and more diverse recommendations.

7 Conclusions and Future Work

Collaborative recommender systems have proven to be accurate, powering the recommendations in many e-commerce and entertainment platforms. Notwithstanding that the evaluation process is key in order to determine system performance, there is no common agreement on how best the evaluation process should be approached, which properties should be evaluated and which metrics should be used. In this article, we presented an evaluation framework to identify and measure the relationships between a set of performance metrics applied to the user-based (*UKNN*) and item-based (*IKNN*) neighbourhood-based algorithms,

and one latent factor model, weighted matrix factorisation (*WMF*). Our experiments show interesting findings in terms of the correlation between the different recommender system algorithms, metrics and datasets used in the evaluation.

In the first experiment (Section 5), an evaluation of three collaborative filtering algorithms was carried out using three different datasets. Not surprisingly, the matrix fatorisation approach (*WMF*) produces more accurate and diverse recommendations which are less biased towards popularity. However, our study points out that *IKNN* covers a wider range of the item space (in two of the three evaluated datasets), where the item coverage of *IKNN* is up to six times greater than that of *WMF*. This is a very important factor for online retailers where recommender systems are expected to promote the sales of their long-tail items.

We conducted a second experiment on the neighbourhood-based algorithms *UKNN* and *IKNN* (Section 6). Interestingly, the results showed that there is a high inverse correlation between diversity and popularity for both *UKNN* and *IKNN*. While previous studies demonstrated that recommending diverse items is at the cost of popularity bias [Said et al. 2012, Fleder and Hosanagar 2009], the reverse correlation indicates that promoting long-tail items using neighbourhood-based algorithms will lead to an improved user experience through the recommendation of diverse items. In particular, our study demonstrates that choosing a smaller number of neighbours for *UKNN* leads to more diverse and less popular recommendations, while at the cost of a relatively small decrease in accuracy. Results also show that the recommendation list accuracy is largely derived from the items which are commonly recommended by *UKNN* and *IKNN*. Moreover, at large neighbourhood sizes, these algorithms are likely to achieve similar levels of user satisfaction since many of the same items tend to be recommended.

The work presented in this article can be further developed in future work. For example, we plan to perform user trials to investigate the effect of neighbourhood size in both the *UKNN* and *IKNN* algorithms on the perceived quality of recommendations, and how this perceived quality correlates with our metrics. Also, we plan to further explore the popularity bias and its effect on recommendations. To this end, we will study the temporal variation of popularity (i.e. the peaks and cycles of item popularity), to understand the correlation between temporal popularity, the recommendations made by different algorithms, and the user's perceived quality of those recommendations.

References

[Adamopoulos and Tuzhilin 2011] Adamopoulos, P., Tuzhilin, A.: "On unexpectedness in recommender systems: or how to expect the unexpected"; Proceedings of the Workshop on Novelty and Diversity in Recommender Systems; 35–42; 2011.

[Anderson 2004] Anderson, C.: "The Long Tail"; Wired Magazine; 12 (2004).

[Bellogin et al. 2011] Bellogin, A., Castells, P., Cantador, I.: "Precision-oriented evaluation of recommender systems: an algorithmic comparison"; Proceedings of the Fifth ACM Conference on Recommender Systems; 333–336; 2011.

[Campos et al. 2011] Campos, P., Díez, F., Sánchez-Montañés, M.: "Towards a more realistic evaluation: Testing the ability to predict future tastes of matrix factorization-based recommenders"; Proceedings of the Fifth ACM Conference on Recommender Systems; 309–312; 2011.

[Cantador et al. 2011] Cantador, I., Brusilovsky, P., Kuflik, T.: "Second workshop on information heterogeneity and fusion in recommender systems"; Proceedings of the Fifth ACM Conference on Recommender Systems; 387–388; 2011.

[Celma and Herrera 2008] Celma, O., Herrera, P.: "A new approach to evaluating novel recommendations"; Proceedings of the Second ACM Conference on Recommender Systems; 179–186; ACM Press, 2008.

[Corona Pampín et al. 2014] Corona Pampín, H. J., Jerbi, H., O'Mahony, M. P.: "Evaluating the relative performance of neighbourhood-based recommender systems"; Proceedings of the Third Spanish Conference on Information Retrieval; 25–36; 2014.

[Cremonesi et al. 2010] Cremonesi, P., Koren, Y., Turrin, R.: "Performance of recommender algorithms on top-N recommendation tasks"; Proceedings of the Fourth ACM Conference on Recommender Systems; 39–46; 2010.

[Fleder and Hosanagar 2009] Fleder, D., Hosanagar, K.: "Blockbuster culture's next rise or fall: the impact of recommender systems on sales diversity"; Management Science; 55 (2009), 5, 697–712.

[Ge et al. 2010] Ge, M., Delgado-Battenfeld, C., Diettmar, J.: "Beyond accuracy: Evaluating recommender systems by coverage and serendipity"; Proceedings of the Fourth ACM Conference on Recommender Systems; 257–260; 2010.

[Gunawardana and Shani 2009] Gunawardana, A., Shani, G.: "A survey of accuracy evaluation metrics of recommendation tasks"; The Journal of Machine Learning Research; 10 (2009), 2935–2962.

[Herlocker et al. 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J.: "An algorithmic framework for performing collaborative filtering"; Proceedings of the 22nd ACM Conference on Research and Development in Information Retrieval; 230–237; 1999.

[Herlocker et al. 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J.: "Evaluating collaborative filtering recommender systems"; ACM Transactions on Information Systems; 22 (2004), 1, 5–53.

[Hu et al. 2008] Hu, Y., Koren, Y., Volinsky, C.: "Collaborative filtering for im-

plicit feedback datasets"; Proccedings of the Eight Conference on Data Mining; 263–272; 2008.

[Hurley 2013] Hurley, N.: "Personalised ranking with diversity"; Proceedings of the Seventh ACM Conference on Recommender Systems; 379–382; 2013.

[Jawaheer et al. 2010] Jawaheer, G., Szomszor, M., Kostkova, P.: "Comparison of implicit and explicit feedback from an online music recommendation service"; Proceedings of the First Workshop on Information Heterogeneity and Fusion in Recommender Systems; 47–51; 2010.

[Karypis 2001] Karypis, G.: "Evaluation of item-based top-N recommendation algorithms"; Proceedings of the Tenth Conference on Information and Knowledge Management; 247–254; 2001.

[Koenigstein et al. 2012] Koenigstein, N., Nice, N., Paquet, U., Schleyen, N.: "The Xbox recommender system"; Proceedings of the Sixth ACM Conference on Recommender Systems; 281–284; 2012.

[Konstan and Riedl 2012] Konstan, J. A., Riedl, J.: "Recommender systems: From algorithms to user experience"; User Modeling and User-Adapted Interaction; 22 (2012), 1-2, 101–123.

[Koren 2010] Koren, Y.: "Collaborative filtering with temporal dynamics"; Communications of the ACM; 53 (2010), 4, 89–97.

[Linden et al. 2003] Linden, G., Smith, B., York, J.: "Amazon.com recommendations. Item-to-item collaborative filtering"; IEEE Internet Computing; 1 (2003), February, 76–80.

[McNee et al. 2006] McNee, S. M., Riedl, J., Konstan, J. A.: "Accurate is not always good: How accuracy metrics have hurt recommender systems"; CHI'06 Extended Abstracts on Human Factors in Computing Systems; 1097–1101; 2006.

[Meyer et al. 2012] Meyer, F., Fessant, F., Clérot, F., Gaussier, E.: "Toward a new protocol to evaluate recommender systems"; Proceedings of the Workshop on Recommendation Utility Evaluation: Beyond RMSE; 9–14; 2012.

[Ning and Karypis 2011] Ning, X., Karypis, G.: "SLIM: Sparse linear methods for top-N recommender systems"; Proceedings of the 11th IEEE Conference on Data Mining.; 497–506; 2011.

[O'Mahony et al. 2004] O'Mahony, M. P., Hurley, N., Kushmerick, N., Silvestre, G.: "Collaborative recommendation: a robustness analysis"; ACM Transactions on Internet Technology; 4 (2004), 4, 344–377.

[Pan et al. 2008] Pan, R., Zhou, Y., Cao, B., Liu, N. N., Lukose, R., Scholz, M., Yang, Q.: "One-Class collaborative filtering"; Proceedings of the Eighth IEEE Conference on Data Mining; 502–511; 2008.

[Pazzani and Billsus 2007] Pazzani, M. J., Billsus, D.: "Content-Based recommendation systems"; The Adaptive Web; 325–341; Springer Berlin Heidelberg, 2007.

[Pradel et al. 2012] Pradel, B., Usunier, N., Gallinari, P.: "Ranking with non-random missing ratings: Influence of popularity and positivity on evaluation metrics"; Proceedings of the Sixth ACM Conference on Recommender Systems; 147–154; 2012.

[Rafter et al. 2009] Rafter, R., O'Mahony, M. P., Hurley, N., Smyth, B.: "What have the neighbours ever done for us? A collaborative filtering perspective"; User Modeling, Adaptation, and Personalization; 355–360; 2009.

[Said et al. 2012] Said, A., Kille, B., Jain, B. J., Albayrak, S.: "Increasing diversity through furthest neighbor-based recommendation"; Proceedings of the Fifth ACM Conference on Web Search and Data Mining; 1–4; 2012.

[Sarwar et al. 2001] Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: "Item-based collaborative filtering recommendation algorithms"; Proceedings of the 10th Conference on World Wide Web; 285–295; 2001.

[Schelter et al. 2013] Schelter, S., Boden, C., Schenck, M., Alexandrov, A., Markl, V.: "Distributed matrix factorization with MapReduce using a series of broadcast-joins"; Proceedings of the Seventh ACM Conference on Recommender Systems; 281–284; 2013.

[Shani and Gunawardana 2011] Shani, G., Gunawardana, A.: "Evaluating recommendation systems"; Recommender Systems Handbook; 257–297; Springer US, 2011.

[Shi et al. 2012] Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., Hanjalic, A.: "CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering"; Proceedings of the Sixth ACM Conference on Recommender Systems; 139–146; 2012.

[Smyth 2007] Smyth, B.: "Case-based recommendation"; The Adaptive Web; 342–376; Springer Berlin Heidelberg., 2007.

[Steck 2010] Steck, H.: "Training and testing of recommender systems on data missing not at random"; Proceedings of the 16th ACM Conference on Knowledge Discovery and Data Mining; 713–722; 2010.

[Trohidis et al. 2008] Trohidis, K., Tsoumakas, G., Kalliris, G., Vlahavas, I.: "Multi-Label classification of music into emotions."; ISMIR; 8 (2008), 325–330.

[Vargas 2015] Vargas, S.: Novelty and Diversity Evaluation and Enhancement in Recommender Systems; Ph.D. thesis; Universidad Autonoma de Madrid (2015).

[Vargas and Castells 2011] Vargas, S., Castells, P.: "Rank and relevance in novelty and diversity metrics for recommender systems"; Proceedings of the Fifth ACM Conference on Recommender Systems; 109–116; 2011.

[Zhang and Hurley 2008] Zhang, M., Hurley, N.: "Avoiding monotony: Improving the diversity of recommendation lists"; Proceedings of the Third ACM Conference on Recommender Systems; 123–130; 2008.