Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation

Daniel Valcarce, Javier Parapar, and Álvaro Barreiro

Information Retrieval Lab
Computer Science Department
University of A Coruña, Spain
{daniel.valcarce,javierparapar,barreiro}@udc.es

Abstract. Recently, Relevance-Based Language Models have been demonstrated as an effective Collaborative Filtering approach. Nevertheless, this family of Pseudo-Relevance Feedback techniques is computationally expensive for applying them to web-scale data. Also, they require the use of smoothing methods which need to be tuned. These facts lead us to study other similar techniques with better trade-offs between effectiveness and efficiency. Specifically, in this paper, we analyse the applicability to the recommendation task of four well-known query expansion techniques with multiple probability estimates. Moreover, we analyse the effect of neighbourhood length and devise a new probability estimate that takes into account this property yielding better recommendation rankings. Finally, we find that the proposed algorithms are dramatically faster than those based on Relevance-Based Language Models, they do not have any parameter to tune (apart from the ones of the neighbourhood) and they provide a better trade-off between accuracy and diversity/novelty.

Keywords: Recommender Systems, Collaborative Filtering, Query Expansion, Pseudo-Relevance Feedback.

1 Introduction

Recommender systems are recognised as a key instrument to deliver relevant information to the users. Although the problem that attracts most attention in the field of Recommender Systems is accuracy, the emphasis on efficiency is increasing. We present new Collaborative Filtering (CF) algorithms. CF methods exploit the past interactions betweens items and users. Common approaches to CF are based on nearest neighbours or matrix factorisation [17]. Here, we focus on probabilistic techniques inspired by Information Retrieval methods.

A growing body of literature has been published on applying techniques from Information Retrieval to the field of Recommender Systems [1,5,14,19–21]. These papers model the recommendation task as an item ranking task with an implicit query [1]. A very interesting approach is to formulate the recommendation problem as a profile expansion task. In this way, the users' profiles can be

expanded with relevant items in the same way in which queries are expanded with new terms. An effective technique for performing automatic query expansion is Pseudo-Relevance Feedback (PRF). In [4, 14, 18], the authors proposed the use of PRF as a CF method. Specifically, they adapted a formal probabilistic model designed for PRF (Relevance-Based Language Models [12]) for the CF recommendation task. The reported experiments showed a superior performance of this approach, in terms of precision, compared to other recommendation methods such as the standard user-based neighbourhood algorithm, SVD and several probabilistic techniques [14]. These improvements can be understood if we look at the foundations of Relevance-Based Language Models since they are designed for generating a ranking of terms (or items in the CF task) in a principled way. Meanwhile, others methods aim to predict the users' ratings. However, it is worth mentioning that Relevance-Based Language Models also outperform other probabilistic methods that focus on top-N recommendation [14].

Nevertheless, the authors in [14] did not analyse the computational cost of generating recommendations within this probabilistic framework. For these reasons, in this paper we analyse the efficiency of the Relevance-Based Language Modelling approach and explore other PRF methods [6] that have a better trade-off between effectiveness and efficiency and, at the same time, do not require any type of smoothing as it is required in [14].

The contributions of this paper are: (1) the adaptation of four efficient Pseudo-Relevance Feedback techniques (Rocchio's weights, Robertson Selection Value, Chi-Squared and Kullback-Leibler Divergence) [6] to CF recommendation, (2) the conception of a new probability estimate that takes into account the length of the neighbourhood in order to improve the accuracy of the recommender system and (3) a critical study of the efficiency of these techniques compared to the Relevance-Based Language Models as well as (4) the analysis of the recommenders from the point of view of the ranking quality, the diversity and the novelty of the suggestions. We show that these new models improve the trade-off between accuracy and diversity/novelty and provide a fast way for computing recommendations.

2 Background

The first paper on applying PRF methods to CF recommendation established an analogy between the query expansion and the recommendation tasks [14]. The authors applied Relevance-Based Language Models [12] outperforming state-of-the-art methods. Next, we describe the PRF task and its adaptation to CF.

Pseudo-Relevance Feedback (PRF) is an automatic technique for improving the performance of a text retrieval system. Feedback information enables to improve the quality of the ranking. However, since explicit feedback is not usually available, PRF is generally a good alternative. This automatic query expansion method assumes that the top retrieval results are relevant. This assumption is reasonable because the goal of the system is to put the relevant results in the top positions of the ranking. Given this pseudo-relevant set of documents, the

system extracts from them the best term candidates for query expansion and performs a second search with the expanded query.

The goal of a recommender is to choose for each user of the system ($u \in \mathcal{U}$) items that are relevant from a set of items ($\mathcal{I}$). Given the user u, the output of the recommender is a personalised ranked list L_u^k of k elements. We denote by $\mathcal{I}_u$ the set of items rated by the user u. Likewise, the set of users that rated the item i is denoted by $\mathcal{U}_i$.

The adaptation of the PRF procedure for the CF task [14] is as follows. Within the PRF framework, the users of the system are analogous to queries in IR. Thus, the ratings of the target user act as the query terms. The goal is to expand the original query (i.e., the profile of the user) with new terms that are relevant (i.e., new items that may be of interest to the user). For performing the query expansion process, it is necessary a pseudo-relevant set of documents, from which the expansion terms are extracted. In the context of recommender systems, the neighbours of the target user play the role of pseudo-relevant documents. Therefore, similar users are used to extract items that are candidates to expand the user profile. These candidate items conform the recommendation list.

Parapar et al. [14] experimented with both estimates of the Relevance-Based Language Models [12]: RM1 and RM2. However, as Eqs. 1 and 2 shows, they are considerably expensive. For each user u, they compute a relevance model R_u and they estimate the relevance of each item i under it, $p(i|R_u)$. V_u is defined as the neighbourhood of the user u. The prior probabilities, $p(v)$ and $p(i)$, are considered uniform. In addition, the conditional probability estimations, $p_\lambda(i|v)$ and $p_\lambda(j|v)$, are obtained interpolating the Maximum Likelihood Estimate (MLE) with the probability in the collection using Jelinek-Mercer smoothing controlled by the parameter λ (see Eq. 3). More details can be found in [14].

$$\text{RM1}: \quad p(i|R_u) \propto \sum_{v \in V_u} p(v) p_\lambda(i|v) \prod_{j \in \mathcal{I}_u} p_\lambda(j|v) \tag{1}$$

$$\text{RM2}: \quad p(i|R_u) \propto p(i) \prod_{j \in \mathcal{I}_u} \sum_{v \in V_u} \frac{p_\lambda(i|v)p(v)}{p(i)} p_\lambda(j|v) \tag{2}$$

$$p_\lambda(i|u) = (1-\lambda)\frac{r_{u,i}}{\sum_{j \in \mathcal{I}_u} r_{u,j}} + \lambda \frac{\sum_{u \in \mathcal{U}} r_{u,i}}{\sum_{u \in \mathcal{U},\, j \in \mathcal{I}} r_{u,j}} \tag{3}$$

3 New Profile Expansion Methods

Next, we describe our PRF proposals for item recommendation based on well-known methods in the retrieval community [6,23] that were never applied to CF. For each user, the following PRF methods assign scores to all the non-rated items of the collection. Neighbourhoods, V_u, are computed using k Nearest Neighbours (k-NN) and C denote the whole collection of users and items.

Rocchio's Weights This method is based on the Rocchio's formula [16]. The assigned score is computed as the sum of the weights for each term of the pseudo-relevant set. This approach promotes highly rated items in the neighbourhood.

$$p_{Rocchio}(i|u) = \sum_{v \in V_u} \frac{r_{v,i}}{|V_u|} \tag{4}$$

Robertson Selection Value (RSV) The Robertson Selection Value (RSV) [15] technique computes a weighted sum of the item probabilities in the neighbourhood. The estimation of these probabilities is described below in this section.

$$p_{RSV}(i|u) = p(i|V_u) \sum_{v \in V_u} \frac{r_{v,i}}{|V_u|} \tag{5}$$

Chi-Squared (CHI-2) This method roots in the chi-squared statistic [6]. The probability in the neighbourhood plays the role of the observed frequency and the probability in the collection is the expected frequency.

$$p_{CHI-2}(i|u) = \frac{\big(p(i|V_u) - p(i|\mathcal{C})\big)^2}{p(i|\mathcal{C})} \tag{6}$$

Kullback-Leibler Divergence (KLD) KLD is a non-symmetric measure for assessing the relative entropy between two probability distributions. Carpineto et. al proposed its use for PRF [6] obtaining good results in the text retrieval task. The idea behind this method is to choose those terms of the pseudo-relevant set which diverge more from the collection in terms of entropy.

$$p_{KLD}(i|u) = p(i|V_u) \log \frac{p(i|V_u)}{p(i|\mathcal{C})} \tag{7}$$

From their equations, we can observe that the complexity of these methods is notably smaller than RM1 and RM2 and are parameter-free. These item ranking functions (except Rocchio's Weights) use probability estimations, $p(i|V_u)$ and $p(i|\mathcal{C})$. We compute these probabilities using the Maximum Likelihood Estimate (MLE) under a multinomial distribution of X. We represent by $\mathcal{U}_X$ the set of users that rated the items from the set X. Likewise, $\mathcal{I}_X$ denotes the set of items that were rated by the users of the set X.

$$p_{MLE}(i|X) = \frac{\sum_{u \in \mathcal{U}_X} r_{u,i}}{\sum_{u \in \mathcal{U}_X,\, j \in \mathcal{I}_X} r_{u,j}} \tag{8}$$

4 Neighbourhood Length Normalisation

When we use a hard clustering algorithm, the number of users in each cluster is variable. Even algorithms such as k-NN can lead to neighbourhoods with

different sizes: a similarity measure based on the common occurrences among users may not be able to find k neighbours for all users when k is too high or when the collection is very sparse—we consider that a neighbour should have at least one common item. In these cases, the information of the neighbourhood is even more important since the user differs strongly from the collection. In IR, this situation would be associated with difficult queries that returned a very limited amount of documents. Therefore, the information of the relevant set should be promoted whilst the global collection information should be demoted.

We incorporated this intuition into the recommendation framework adding a bias to the probability estimate. Thus, we normalise the MLE by dividing the estimate by the number of users in the population as follows:

$$p_{NMLE}(i|X) \overset{\text{rank}}{=} \frac{1}{|\mathcal{U}_X|} \frac{\sum_{u \in \mathcal{U}_X} r_{u,i}}{\sum_{u \in \mathcal{U}_X, j \in \mathcal{I}_X} r_{u,j}} \tag{9}$$

This improvement does not make sense for the RSV item ranking function because the ranking would be the same (the scores will be rescaled by a constant); however, it can be useful for CHI-2 and KLD methods as it can be seen in Sec. 5.

5 Evaluation

We used three film datasets from GroupLens[1]: *MovieLens 100k*, *MovieLens 1M* and *MovieLens 10M*, for the efficiency experiment. Additionally, we used the *R3-Yahoo! Webscope Music*[2] dataset and the *LibraryThing*[3] book collection for the effectiveness tests. The details of the collections are gathered in Table 1. We used the splits provided by the collections. However, since Movielens 1M and LibraryThing do not offer predefined partitions, we selected 80% of the ratings of each user for the training subset whilst the rest is included in the test subset.

5.1 Evaluation methodology

In CF evaluation, a great variety of metrics have been applied. Traditionally, recommenders were designed as rating predictors and, thus, the evaluation was based on error metrics. However, there is a consensus among the scientific community that it is more useful to model recommendation as a ranking task (top-N recommendation) which leads to the use of precision-oriented metrics [2, 10, 13]. In addition, it was stated that not only accuracy but diversity and novelty are key properties of the recommendations [10]. For this reason, in this study we use metrics for these aspects.

We followed the *TestItems* approach described by Bellogín et al. [2] for estimating the precision of the recommendations. For each user, we compute a ranking for all the items having a test rating by some user and no training

[1] `http://grouplens.org/datasets/movielens/`
[2] `http://webscope.sandbox.yahoo.com`
[3] `http://www.macle.nl/tud/LT/`

Table 1: Datasets statistics

Dataset	Users	Items	Ratings	Density
MovieLens 100k	943	1682	100,000	6.305%
MovieLens 1M	6,040	3,952	1,000,209	4.190%
MovieLens 10M	71,567	10,681	10,000,054	1.308%
R3-Yahoo!	15,400	1,000	365,703	2.375%
LibraryThing	7,279	37,232	749,401	0.277%

rating by the target user. It has been acknowledged that considering non-rated items as irrelevant may underestimate the true metric value (since non-rated items can be of interest to the user); however, it provides a better estimation of the recommender quality [2, 13].

The employed metrics are evaluated at a specified cut-off rank, i.e., we consider only the top k recommendations of the ranking for each user because these are the ones presented to the user. For assessing the quality of the ranking we employed nDCG. This metric uses graded relevance of the ratings for judging the ranking quality. Values of nDGG increases when highly relevant documents are located in the top positions of the ranking. We used the standard formulation as described in [22]. We also employed the complement of the Gini index for quantifying the diversity of the recommendations [9]. The index is 0 when only a single item is recommended for every user. On the contrary, a value of 1 is achieved when all the items are equally recommended among the users. Finally, to measure the ability of a recommender system to generate unexpected recommendations, we computed the mean self-information (MSI) [25]. Intuitively, the value of this metric increases when unpopular items are recommended.

5.2 Baselines

To assess the performance of the proposed recommendation techniques, we chose a representative set of state-of-the-art recommenders. We used a standard user-based neighbourhood CF algorithm (labelled as UB): the neighbours are computed using k-NN with Pearson's correlation as the similarity measure [8]. We also tested Singular Value Decomposition (SVD), a matrix factorisation technique which is among the best methods for rating prediction [11]. Additionally, we included an algorithm which has its roots in the IR probabilistic modelling framework [20], labelled as UIR-Item. Finally, as the strongest baselines, we chose the RM1 and RM2 models [14]. Instead of employing Jelinek-Mercer smoothing as it was originally proposed [14], we used Absolute Discounting because recent studies showed that it is more effective stable than Jelinek-Mercer [18].

5.3 Efficiency experiment

The principal motivation for this work was to propose more efficient PRF recommendation techniques than RM1 and RM2. To assess the efficiency of our

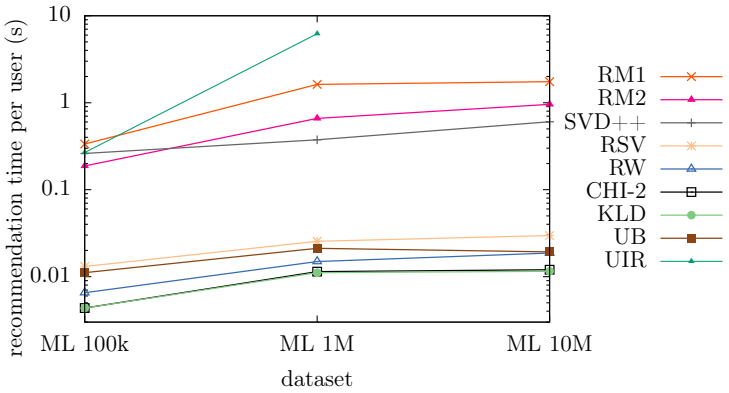

Fig. 1: Recommendation time per user (in logarithmic scale) using UIR-Item (UIR), RM1, RM2, RSV, Rocchio's Weights (RW), CHI-2 and KLD algorithms with NMLE as the probability estimate on the MovieLens 100k, 1M and 10M collections.

proposals, we measured the user recommendation times on the MovieLens 100k, 1M and 10M datasets. The neighbourhoods are precomputed using k-NN with Pearson's correlation and $k = 100$. Since the time of computing the neighbours is common to each method, we can ignore it. We measured the algorithms in a desktop computer with an Intel i7-4790 @3.60GHz and 16 GB DDR3 1600 MHz.

Figure 1 illustrates the recommendation times on the three datasets. We report times (in logarithmic scale) for UIR-Item, RM1, RM2, RSV, Rocchio's Weights, CHI-2 and KLD. These results demonstrate that the proposed new methods are dramatically faster than RM1 and RM2 (our proposals obtain speed-ups up to 200x) meanwhile the variations in time among our proposed methods are small. Additionally, the differences in time between the probability estimates (MLE and NMLE) are insignificant. We do not report the recommendation time of UIR-Item on the MovieLens 10M collection because its performance was so poor that the experiment did not finish in a week.

5.4 Effectiveness experiment

We present now the results of our methods as well as the baselines on the MovieLens 100k, Movielens 1M, R3-Yahoo! and LibraryThing collections. We used k-NN with Pearson's similarity for computing the neighbourhoods and we tuned k from 50 to 950 neighbours (in steps of 50) for each method in the MovieLens 100k dataset. Those values were then used in the rest of the collections. We also tuned the number of latent factors of SVD and the λ parameter of UIR-Item. All parameters were tuned in order to optimise nDCG@10 using cross-validation with the five folders provided by the MovieLens 100k collection. In order to facilitate the reproducibility of these experiments we show, for each method, the optimal values for the tuned parameters in Table 2.

Table 2: Values of nDCG@10 for each recommender approach. Statistically significant improvements according to Wilcoxon Test ($p < 0.05$) with respect to the baselines UB, SVD, UIR-Item, RM1, RM2 are superscripted with a, b, c, d and e, respectively. The complementary statistically significant decreases are subscripted in the same way. The values in bold indicate the best recommender for the each dataset. The values underlined are not statistically different from the best value.

Algorithm		**Tuned param.**	**ML 100k**	**ML 1M**	**R3-Yahoo!**	**LibraryThing**
UB		$k = 50$	0.0468_{bcde}	0.0313_{bcde}	0.0108_{cde}	0.0055^{b}_{cde}
SVD		$factors = 400$	0.0936^{a}_{cde}	0.0608^{a}_{cde}	0.0101_{cde}	0.0015_{acde}
UIR-Item		$\lambda = 0.5$	0.2188^{ab}_{de}	0.1795^{abd}_{e}	0.0174^{abd}_{e}	0.0673^{abd}_{e}
RM1		$k = 400,\ \delta = 0.1$	0.2473^{abc}_{e}	0.1402^{ab}_{ce}	0.0146^{ab}_{ce}	0.0444^{ab}_{ce}
RM2		$k = 550,\ \delta = 0.1$	$\mathbf{0.3323}^{abcd}$	$\mathbf{0.1992}^{abd}$	$\mathbf{0.0207}^{abcd}$	0.0957^{abcd}
Rocchio's Weights		$k = 600$	0.2604^{abcd}_{e}	0.1557^{abd}_{ce}	0.0194^{abcd}_{e}	0.0892^{abcd}_{e}
RSV	MLE	$k = 600$	0.2604^{abcd}_{e}	0.1557^{abd}_{ce}	0.0194^{abcd}_{e}	0.0892^{abcd}_{e}
KLD	MLE	$k = 850$	0.2693^{abcd}_{e}	0.1264^{ab}_{cde}	$\underline{0.0197}^{abcd}$	$\mathbf{0.1576}^{abcde}$
	NMLE	$k = 700$	0.3120^{abcd}_{e}	0.1546^{ab}_{cde}	$\underline{0.0201}^{abcd}$	0.1101^{abcde}
CHI-2	MLE	$k = 500$	0.0777^{a}_{bcde}	0.0709^{ab}_{cde}	0.0149^{ab}_{ce}	0.0939^{abcd}
	NMLE	$k = 700$	0.3220^{abcd}_{e}	0.1419^{ab}_{cde}	$\underline{0.0204}^{abcd}$	0.1459^{abcde}

The obtained nDCG@10 values are reported in Table 2 with statistical significance tests (two-sided Wilcoxon test with $p < 0.05$). Generally, RM2 is the best recommender algorithm as it was expected—better probabilistic models should lead to better results. Nevertheless, it can be observed that in the R3-Yahoo! dataset, the best nDCG values of our efficient PRF methods are not statistically different from RM2. Moreover, in the LibraryThing collection, many of the proposed models significantly outperform RM2 with important improvements. This may be provoked by the sparsity of the collections which leads to think that RM2 is too complex to perform well under this more common scenario. Additionally, although we cannot improve the nDCG figures of RM2 on the MovieLens 100k, we significantly surpass the other baselines.

In most of the cases, the proposals that use collection statistics (i.e., KLD and the CHI-2 methods) tend to perform better than those that only use neighbourhood information (Rocchio's Weights and RSV). Regarding the proposed neighbourhood length normalisation, the experiments show that NMLE improves the ranking accuracy compared to the regular MLE in the majority of the cases. Thus, the evidence supports the idea that the size of the users' neighbourhoods is an important factor to model in a recommender system.

Now we take the best baselines (UIR-Item and RM2) and our best proposal (CHI-2 with NMLE) in order to study the diversity and novelty of the top ten recommendations. Note that we use the same rankings which were optimized for nDCG@10. The values of Gini@10 and MSI@10 are presented in Tables 3 and 4, respectively. In the case of Gini, we cannot perform paired significance analysis since it is a global metric.

Table 3: Gini@10 values of UIR-Item, RM2 and CHI-2 with NMLE (optimised for nDCG@10). Values in bold indicate the best recommender for the each dataset. Significant differences are indicated with the same criteria as in Table 2.

Algorithm	**ML 100k**	**ML 1M**	**R3-Yahoo!**	**LibraryThing**
UIR-Item	0.0124	0.0050	0.0137	0.0005
RM2	0.0256	0.0069	0.0207	0.0019
CHI-2 NMLE	**0.0450**	**0.0106**	**0.0506**	**0.0539**

Table 4: MSI@10 values of UIR-Item, RM1, RM2 and CHI-2 with NMLE (optimised for nDCG@10). Values in bold indicate the best recommender for the each dataset. Significant differences are indicated with the same criteria as in Table 2.

Algorithm	**ML 100k**	**ML 1M**	**R3-Yahoo!**	**LibraryThing**
UIR-Item	5.2337_{e}	8.3713_{e}	3.7186_{e}	17.1229_{e}
RM2	6.8273^{c}	8.9481^{c}	4.9618^{c}	$\mathbf{19.27343}^{c}$
CHI-2 NMLE	$\mathbf{8.1711}^{ec}$	$\mathbf{10.0043}^{ec}$	$\mathbf{7.5555}^{ec}$	8.8563

We observe that CHI-2 with NMLE generates more diverse recommendations than RM2, which is the strongest baseline in terms of nDCG. Also, CHI-2 with NMLE presents good novelty figures except for the LibraryThing collection. However, as we mentioned before, the performance of RM2 on the LibraryThing dataset is quite poor in terms of nDCG compared to the other models. It is easy to improve diversity and novelty decreasing the accuracy values [25]; however, we aim for an effective method in terms of all the metrics. In summary, the results showed that CHI-2 with NMLE is among the best performing studied methods with a good trade-off between accuracy and diversity/novelty.

The advantages in terms of the trade-offs among ranking precision and diversity and novelty are reported in Fig. 2 where we present the G-measure for both relations when varying the size of the neighbourhood. The G-measure is the geometric mean of the considered metrics which effectively normalizes the true positive class (in this case, relevant and diverse or relevant and novel). In this particular scenario, the use of other kind of means is not appropriate [7] due to the strong dependency and the difference in scale among the analysed variables. In the graphs, we observe that with values of $k > 400$, our proposal is even better than the strongest baseline, RM2, for both trade-offs. Therefore, we presented a competitive method in terms of effectiveness which is up to 200 times faster than previous PRF algorithms for CF.

6 Related Work

Exploring Information Retrieval (IR) techniques and applying them to Recommender Systems is an interesting line of research. In fact, in 1992, Belkin and

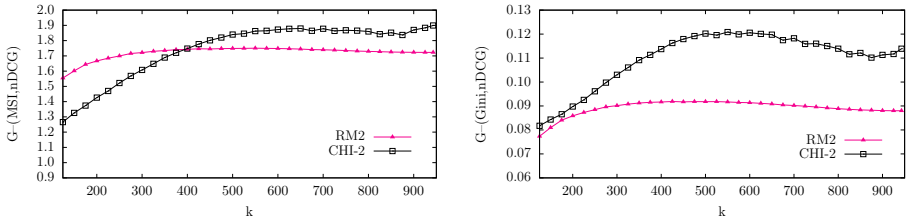

Fig. 2: Values of the G-measure in the MovieLens 100k collection plotted against the size of the neighbourhood (k), for the nDCG@10-MSI@10 (left) and the nDCG@10-Gini@10 (right) trade-offs.

Croft already stated that Information Retrieval and Information Filtering (IF) are two sides of the same coin [1]. Recommenders are automatic IF systems: their responsibility lies in selecting relevant items for the users. Consequently, besides the work of Parapar et al. on applying Relevance-Based Language Models to CF recommendation [14], there is a growing amount of literature about different approaches that exploit IR techniques for recommendation [5, 19–21].

Wang et al. derived user-based and item-based CF algorithms using the classic probability ranking principle [20]. They also presented a probabilistic relevance framework with three models [21]. Also, Wang adapted the language modelling scheme to CF using a risk-averse model that penalises less reliable scores [19].

Another approach is the one formulated by Bellogín et al. [5]. They devised a general model for unifying memory-based CF methods and text retrieval algorithms. They show that many IR methods can be used within this framework obtaining better results than classic CF techniques for the item ranking task.

Relevance-Based Language Models were also adapted to CF in a different manner. Bellogín et al. [4] formulate the formation of user neighbourhoods as a query expansion task. Then, by using the negative cross entropy ranking principle, they used the neighbours to compute item recommendations.

7 Conclusions and Future Work

Since Relevance Models [12] are an effective tool for item recommendation [14], the aim of this work was to assess if other faster PRF methods could be used for the same task. The results of this investigation revealed that, indeed, simpler and more efficient PRF techniques are suitable for this CF task. We have carried out experiments that showed that the proposed recommendation algorithms (Rocchio's Weigths, RSV, KLD and CHI-2) are orders of magnitude faster than the Relevance Models for recommendation. These alternatives offer important improvements in terms of computing time while incurring, in some cases, in a modest decrease of accuracy. Furthermore, these methods lack of parameters:

they only rely on the neighbourhood information. In a large-scale scenario, a speed-up of 200x can lead to notable savings in computational resources.

In terms of ranking accuracy, various methods achieve statistically comparable performance to RM2 in several datasets and they even outperform all the baselines in one collection. Additionally, if we analyse the diversity and novelty figures, we can conclude that the proposed models offer more novel and diverse recommendations than RM2. Additionally, the empirical findings of this study support the idea of neighbourhood length normalisation that we introduced into the Maximum Likelihood Estimate. Overall, we can conclude that CHI-2 with NMLE provide highly precise and fast recommendations with a good trade-off between accuracy and diversity/novelty.

We think that exploring other state-of-the-art PRF techniques such as Divergence Minimization Models or Mixture Models [24] for recommendation may be a fruitful area for further research.

Moreover, a future study investigating different techniques for generating neighbourhoods would be very interesting. In this paper, we employed k-NN algorithm because of its efficiency. Nevertheless, exploring other clustering methods may produce important improvements. For example, the combination of Relevance-Based Language Models with Posterior Probability Clustering, a type of non-negative matrix factorisation, has been proved to generate highly precise recommendations [14]. Similarly, it may be of interest the use of Normalised Cut (a spectral clustering method) since it has been reported that it improves the effectiveness of the standard neighbourhood-based CF algorithms [3].

Acknowledgments. This work was supported by the *Ministerio de Economía y Competitividad* of the Goverment of Spain under grants TIN2012-33867 and TIN2015-64282-R. The first author also wants to acknowledge the support of *Ministerio de Educación, Cultura y Deporte* of the Government of Spain under the grant FPU014/01724.

References

1. N. J. Belkin and W. B. Croft. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Commun. ACM*, 35(12):29–38, 1992.
2. A. Bellogín, P. Castells, and I. Cantador. Precision-oriented Evaluation of Recommender Systems. In *RecSys '11*, page 333. ACM, 2011.
3. A. Bellogín and J. Parapar. Using Graph Partitioning Techniques for Neighbour Selection in User-based Collaborative Filtering. In *RecSys '12*, pages 213–216. ACM, 2012.
4. A. Bellogín, J. Parapar, and P. Castells. Probabilistic Collaborative Filtering with Negative Cross Entropy. In *RecSys '13*, pages 387–390. ACM, 2013.
5. A. Bellogín, J. Wang, and P. Castells. Bridging Memory-Based Collaborative Filtering and Text Retrieval. *Inf. Retr.*, 16(6):697–724, 2013.
6. C. Carpineto, R. de Mori, G. Romano, and B. Bigi. An Information-Theoretic Approach to Automatic Query Expansion. *ACM Trans. Inf. Syst.*, 19(1):1–27, 2001.

7. F. Coggeshall. The arithmetic, geometric, and harmonic means. *Q. J. Econ.*, 1(1):83–86, 1886.
8. C. Desrosiers and G. Karypis. A Comprehensive Survey of Neighborhood-based Recommendation Methods. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 2011.
9. D. Fleder and K. Hosanagar. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Manage. Sci.*, 55(5):697–712, 2009.
10. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
11. Y. Koren and R. Bell. Advances in Collaborative Filtering. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 145–186. Springer, 2011.
12. V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *SIGIR '01*, pages 120–127. ACM, 2001.
13. M. R. McLaughlin and J. L. Herlocker. A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience. In *SIGIR '04*, pages 329–336. ACM, 2004.
14. J. Parapar, A. Bellogín, P. Castells, and A. Barreiro. Relevance-based language modelling for recommender systems. *Inf. Process. Manage.*, 49(4):966–980, 2013.
15. S. E. Robertson. On Term Selection for Query Expansion. *J. Doc.*, 46(4):359–364, 1990.
16. J. J. Rocchio. Relevance Feedback in Information Retrieval. In G. Salton, editor, *The SMART Retrieval System - Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971.
17. J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering Recommender Systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, *The Adaptive Web*, volume 4, pages 291–324. Springer-Verlag, 2007.
18. D. Valcarce, J. Parapar, and A. Barreiro. A Study of Smoothing Methods for Relevance-Based Language Modelling of Recommender Systems. In *ECIR '15*, volume 9022, pages 346–351. Springer, 2015.
19. J. Wang. Language Models of Collaborative Filtering. In *AIRS '09*, pages 218–229. Springer-Verlag, 2009.
20. J. Wang, A. P. de Vries, and M. J. Reinders. A User-Item Relevance Model for Log-based Collaborative Filtering. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *ECIR '06*, volume 3936, pages 37–48. Springer-Verlag, 2006.
21. J. Wang, A. P. de Vries, and M. J. T. Reinders. Unified Relevance Models for Rating Prediction in Collaborative Filtering. *ACM Trans. Inf. Syst.*, 26(3):1–42, 2008.
22. Y. Wang, L. Wang, Y. Li, D. He, W. Chen, and T.-Y. Liu. A Theoretical Analysis of NDCG Ranking Measures. In *COLT '13*, pages 1–30. JMLR.org, 2013.
23. W. S. Wong, R. W. P. Luk, H. V. Leong, L. K. Ho, and D. L. Lee. Re-examining the Effects of Adding Relevance Information in a Relevance Feedback Environment. *Inf. Process. Manage.*, 44(3):1086–1116, 2008.
24. C. Zhai and J. Lafferty. Model-based Feedback in the Language Modeling Approach to Information Retrieval. In *CIKM '01*, pages 403–410. ACM, 2001.
25. T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang. Solving the Apparent Diversity-Accuracy Dilemma of Recommender Systems. *PNAS*, 107(10):4511–5, 2010.