A Study of Priors for Relevance-Based Language Modelling of Recommender Systems

Daniel Valcarce, Javier Parapar, Álvaro Barreiro
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
{daniel.valcarce, javierparapar, barreiro}@udc.es

ABSTRACT

Probabilistic modelling of recommender systems naturally introduces the concept of prior probability into the recommendation task. Relevance-Based Language Models, a principled probabilistic query expansion technique in Information Retrieval, has been recently adapted to the item recommendation task with success. In this paper, we study the effect of the item and user prior probabilities under that framework. We adapt two priors from the document retrieval field and then we propose other two new probabilistic priors. Evidence gathered from experimentation indicates that a linear prior for the neighbour and a probabilistic prior based on Dirichlet smoothing for the items improve the quality of the item recommendation ranking.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Algorithms, Experimentation

Keywords

Recommender Systems; Collaborative Filtering; Relevance-Based Language Models; prior probability

1. INTRODUCTION

Recommender systems aim to find relevant pieces of information which may be of interest to the users. Since this objective can be modelled as a personalised item ranking task, the use of techniques from the Information Retrieval (IR) field is becoming more and more popular. An effective approach to recommendation is Collaborative Filtering (CF). This family of algorithms exploits the past interaction between users and items to generate personalised suggestions. Loosely speaking, CF algorithms analyse the users' taste in a system and intend to recommend relevant items.

Relevance-Based Language Models (RM) were conceived for expanding queries automatically [6]. However, they can be effectively applied to CF recommendation [8, 2, 10]. The task of recommending items to a user can be assimilated to the task of expanding a query with new terms. In this case, the user profile plays the role of the query. This approach has been proved to be notably effective in terms of ranking quality, surpassing other state-of-the-art algorithms such as nearest neighbours or matrix factorisation methods [8].

Information Retrieval algorithms often include the notion of a *document prior* which encodes the importance of a document independently of the user's query. These priors can be used for improving the performance of the document ranking. Therefore, they were thoroughly studied in the IR field [5, 4, 9, 3].

The use of probabilistic models such as RM for recommendation provides several advantages. One of them is the possibility of introducing prior probabilities into the recommendation process. In fact, the most effective estimation of the Relevance-Based Language Models for CF is RM2 whose formulation includes a user prior and an item prior. Previous works on the relevance modelling of recommender systems considered those priors uniform [8, 2] leaving open the possibility of further studying this aspect. Thus, in this paper, we analyse the effects of the user and the item priors in the RM2 model. First, we adapt two effective document length priors from IR to the CF task [5, 3]. Next, we propose two new variants of the probabilistic length prior devised in [3]. Finally, we conduct a series of experiments that show that the use of a linear length prior for the users and a probabilistic length prior based on Dirichlet smoothing for the items leads to significant improvements in terms of ranking accuracy.

2. RELEVANCE MODELLING OF RECOMMENDER SYSTEMS

Recommender systems help users with the finding of relevant items. The set of users of the system is denoted by $\mathcal{U}$ and the set of items by $\mathcal{I}$. When a user u rate an item i, we refer to that rating with the notation $r_{u,i}$. We use the term $\mathcal{I}_u$ to represent the set of items that were rated by the user u. Likewise, the set of users that rated the item i is denoted by $\mathcal{U}_i$. The goal of the recommender is to generate, for each user u, a personalised list of k items L_u^k ordered in decreasing order of estimated relevance.

Relevance-Based Language Models [6] are a probabilistic

RecSys'15, September 16–20, 2015, Vienna, Austria.

ACM 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2799677.

pseudo-relevance feedback technique for text retrieval. Their goal is to expand the user's query with new relevant terms to improve the retrieval performance. In order to achieve so, an initial retrieval is computed and relevance over the first top results is assumed (this set of documents is called pseudo-relevant set). Recently, RM have been adapted to CF showing high accuracy figures [8]. Under this scenario, users' profiles play the role of both documents and queries and the items are equivalent to the terms. In this way, instead of expanding queries with new terms, we can use RM to expand users' profiles with new relevant items. The pseudo-relevant set in the CF task is the neighbourhood of the target user. These neighbourhoods can be determined using, for instance, the traditional k-NN algorithm.

There exist two estimates of Relevance Models that were proposed for recommendation. In this paper, we focus on RM2 since it was reported to provide the best results. This algorithm computes a Relevance Model for each user u calculates the relevance of each item i under it, $p(i|R_u)$:

$$p(i|R_u) \propto p(i) \prod_{j \in \mathcal{I}_u} \sum_{v \in V_u} \frac{p(i|v)p(v)}{p(i)} p(j|v) \tag{1}$$

where V_u refers to the set of neighbours of the user u. The conditional probability of an item given a user, $p(i|u)$, is computed by smoothing the maximum likelihood estimate. The original papers used Jelinek-Mercer smoothing [8]; however, in this paper, we employ Absolute Discounting smoothing which consists in subtracting a constant δ from each rating. The rationale behind this decision is that this smoothing method models the user bias yielding better recommendations. The reason is that the amount of smoothing applied from the background collection $p(i|\mathcal{C})$ is inversely proportional to the average rating of the user [10]:

$$p(i|u) = \frac{\max(\mathrm{r}_{u,i} - \delta, 0)}{\sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}} + \delta \frac{|\mathcal{I}_u|}{\sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}} p(i|\mathcal{C}) \tag{2}$$

Finally, we need to specify how to compute the user prior $p(v)$ and the item prior $p(i)$. In the original paper uniform distributions were used [8]. Our proposed priors are described next.

3. PROPOSED PRIORS

The recommendation formula that results from applying Relevance Models to the CF task (see Eq. 1) involves the use of a user prior for each of the neighbours, $p(v)$, and an item prior for each of the candidate items to be recommended, $p(i)$. Next, we introduce the user priors; from these, the deviation of the item priors is straightforward.

We will use the following priors to compute the prior probability of a neighbour in the RM2 algorithm.

Uniform (U).

This prior is drawn from a uniform distribution. That is to say, every user in the population has the same prior probability. We use this prior as our baseline.

$$p_u(u) = \frac{1}{|\mathcal{U}|} \tag{3}$$

Linear (L).

The linear document length prior was previously used in Information Retrieval [5, 3]. Its adaptation to recommendation boosts those users with larger rating profiles. In this way, we are promoting the recommendations that came from the power users of the system.

$$p_L(u) = p(u|\mathcal{C}) = \frac{\sum_{i \in \mathcal{I}_u} r_{u,i}}{\sum_{v \in \mathcal{U}} \sum_{j \in \mathcal{I}_v} r_{v,j}} \tag{4}$$

Probabilistic using Jelinek-Mercer (PJM).

An effective prior for Information Retrieval is the probabilistic document length prior proposed in [3]. It is indirectly based on the document length. For the CF task, this method computes the users' priors as a function of the statistics of the items they rated. The original formulation of this prior employs Jelinek-Mercer smoothing:

$$\begin{aligned} p_{PJM}(u) &\propto \sum_{i \in \mathcal{I}_u} p(i|u) \\ &= \sum_{i \in \mathcal{I}_u} \Big[(1-\lambda) \frac{r_{u,i}}{\sum_{j \in \mathcal{I}_u} r_{u,j}} + \lambda\, p(i|\mathcal{C}) \Big] \\ &= (1-\lambda) + \lambda \sum_{i \in \mathcal{I}_u} p(i|\mathcal{C}) \end{aligned} \tag{5}$$

Probabilistic using Dirichlet (PD).

In this work, we also propose to explore the previous Probabilistic prior using Dirichlet smoothing:

$$\begin{aligned} p_{PD}(u) &\propto \sum_{i \in \mathcal{I}_u} p(i|u) \\ &= \sum_{i \in \mathcal{I}_u} \frac{r_{u,i} + \mu\, p(i|\mathcal{C})}{\mu + \sum_{j \in \mathcal{I}_u} r_{u,j}} \\ &= \frac{\sum_{i \in \mathcal{I}_u} r_{u,i} + \mu \sum_{i \in \mathcal{I}_u} p(i|\mathcal{C})}{\mu + \sum_{i \in \mathcal{I}_u} r_{u,i}} \end{aligned} \tag{6}$$

Probabilistic using Absolute Discounting (PAD).

Finally, the same applying Absolute Discounting:

$$\begin{aligned} p_{PAD}(u) &\propto \sum_{i \in \mathcal{I}_u} p(i|u) \\ &= \sum_{i \in \mathcal{I}_u} \frac{\max(r_{u,i} - \delta, 0) + \delta\, |\mathcal{I}_u|\, p(i|\mathcal{C})}{\sum_{j \in \mathcal{I}_u} r_{u,j}} \\ &= \frac{\sum_{i \in \mathcal{I}_u} \max(r_{u,i} - \delta, 0) + \delta\, |\mathcal{I}_u| \sum_{i \in \mathcal{I}_u} p(i|\mathcal{C})}{\sum_{j \in \mathcal{I}_u} r_{u,j}} \end{aligned} \tag{7}$$

The straightforward item-based counterparts of the proposed priors are not showed here for brevity.

4. EVALUATION

Next, we describe the collections, methodology, experiments and the results analysing the priors proposed in the previous section.

The experiments were conducted in three collections: *MovieLens 100k*[1] (films), *R3-Yahoo! Webscope Music*[2] (music) and LibraryThing[3] (books). Their details can be found in Table 1.

[1] `http://grouplens.org/datasets/movielens/`

[2] `http://webscope.sandbox.yahoo.com/catalog.php`

[3] `http://www.macle.nl/tud/LT/`

Table 1: Datasets statistics

Dataset	Users	Items	Ratings	Density
MovieLens 100k	943	1682	100,000	6.305%
R3-Yahoo!	15,400	1,000	365,703	2.375%
LibraryThing	7,279	37,232	749,401	0.277%

We performed five-fold cross-validation using the splits provided by MovieLens 100k to train the prior parameters and choose the best priors. Then, we applied these priors and their parameters to the R3-Yahoo! and LibraryThing collections. Since LibraryThing does not include a default split, we randomly selected 80% of the ratings of each user as training subset and the rest as test subset. Thus, the splits of the MovileLens collection were used for tuning parameters whilst the splits of other datasets were employed solely for evaluation purposes, that is, the training split was used as seed data for the recommender system and the testing split was used for assessing the performance.

We followed the *TestItems* approach described in [1] for estimating the ranking quality of the recommendations. For each user in the test subset, we compute recommendations including all the items in the test subset. Although this methodology underestimates the true value of the precision-oriented metrics (because it considers that non-rated items are irrelevant), it provides more reliable results.

We decided to use nDCG (Normalised Discounted Cumulative Gain) at a cut-off value of 10 for assessing the quality of the top ranking. We employed the *standard formulation* [11] using the ratings in test as graded relevance judgements.

For computing the neighbourhoods, we utilised k-NN with Pearson's correlation coefficient as similarity metric. We fixed the value of k to 400 neighbours. We used the optimal smoothing method (Absolute Discounting with $\delta = 0.1$) reported in [10].

Regardless the fact that we tested all possible combinations of priors (and their parameters), we decided to present the results in two steps for the sake of the discussion. First, we introduce the results of tuning only the user prior and, second, we show the results of tuning the item prior using the best user prior. We denote that we use the prior X for users and the prior Y for items with the following notation X-Y.

4.1 Tuning the User Prior

Taking a uniform distribution for the item prior, we tested all the priors proposed in Sec. 3. Figure 1 illustrates the results, in terms of nDCG@10, of this experiment.

We can observe that using a uniform distribution is the worst scenario: any of the others user priors improves the ranking quality. In spite of the fact that the Probabilistic priors improve the nDCG figures, the Linear prior is the one that provides the best results implying that we should rely on neighbours with a high number of ratings. This finding suggests that users with a tiny rating profile are not very reliable for computing recommendations since we have few data about them.

An advantage of this finding is that the Linear prior is a parameter-free technique. This allows to improve the performance of a RM2 recommender without introducing more complexity into the model.

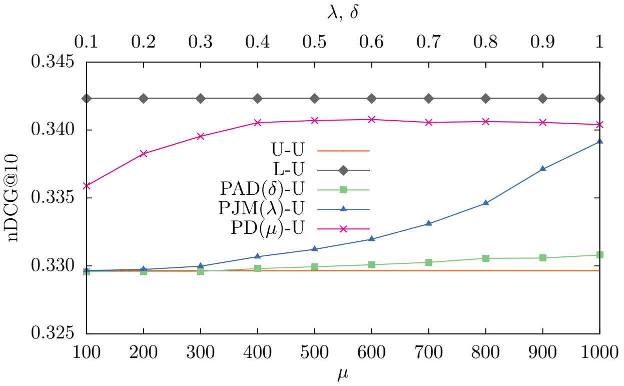

Figure 1: Values of nDCG@10 for RM2 varying the user prior and taking a uniform distribution for the item prior. We show Uniform (U) and Linear (L) priors as well as Probabilistic priors using Jelinek-Mercer (PJM), Dirichlet (PD) and Absolute Discounting (PAD) smoothings.

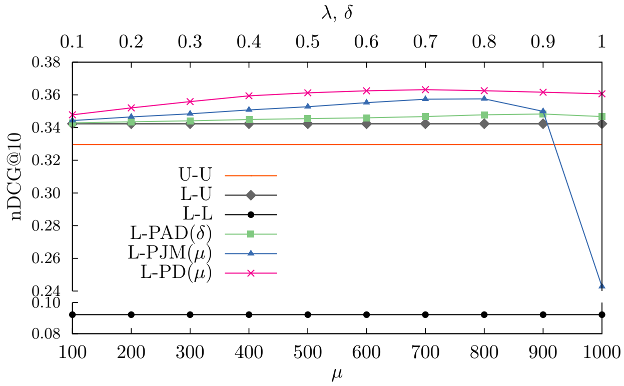

Figure 2: Values of nDCG@10 for RM2 varying the item prior and taking a Linear prior (L) for the user prior. We show Uniform (U) and Linear (L) priors as well as Probabilistic priors using Jelinek-Mercer (PJM), Dirichlet (PD) and Absolute Discounting (PAD) smoothings. We also present the baseline consisting in using both uniform priors.

4.2 Tuning the Item Prior

Now that we have the optimal user (neighbour) prior, we fixed it and we focus on finding the best item prior. We show the values of nDCG@10 obtained for each item prior in Fig. 2. For the sake of comparison, we also present the main baseline which consists in using uniform distributions for users and items (U-U).

In contrast to the previous scenario, we can appreciate that the performance of the Linear prior is very poor. We should note that the item prior divides the RM2 estimate. Thus, the Linear prior is demoting items that have a high number of ratings. Intuitively, lowering the importance of popular items is not a good idea if we want to generate recommendations although it may promote more novel and diverse suggestions.

On the other hand, the results show that the use Probabilistic priors affects the ranking quality positively. It is interesting to observe that Jelinek-Mercer with the parameter $\lambda = 1$ degrades the performance of the recommender.

Table 2: Test values of nDCG@10 using the classic user-based CF neighbour algorithm (UB), Single Value Decomposition (SVD), RM2 with uniform priors (U) and RM2 with Linear prior (L) for the neighbours and Probabilistic Prior with Dirichlet (PDP) for the items. Bolded cells correspond to the best tested method for each dataset. Statistically significant improvements according to the two-sided Wilcoxon test ($p < 0.01$) with respect to UB, SVD, RM2-U-U and RM2-L-PD are superscripted with a, b, c and d, respectively. The complementary statistically significant decreases are subscripted in the same way.

Method	**MovieLens 100k**	**R3-Yahoo!**	**LibraryThing**
UB	0.0468_{bcd}	0.0106_{cd}	0.0055^{b}_{cd}
SVD	0.0936^{a}_{cd}	0.0103_{cd}	0.0014_{acd}
RM2-U-U	0.3296^{ab}_{d}	0.0205^{ab}	0.0900^{ab}_{d}
RM2-L-PD ($\mu = 700$)	$\mathbf{0.3632}^{abc}$	$\mathbf{0.0207}^{ab}$	$\mathbf{0.0942}^{abc}$

This is expected because in this extreme case since the effect of this prior in the denominator would demote items that were rated by power users. In contrast, if we lower λ too much, we are obtaining a uniform prior.

The Probabilistic prior based on Absolute Discounting smoothing produces worse results than the other Probabilistic priors because of the counterintuitiveness in its formulation. Although this method normalises the bias of the users (see Sec. 2), when we amplify the effect of the power users we are also decreasing the importance of popular items because we discount a constant δ from each rating (i.e., the more ratings an item has, the more discount we apply).

Finally, the Probabilistic prior based on Dirichlet presents the best figures, with $\mu = 700$ as the optimal value of the parameter. Dirichlet smoothing has been thoroughly studied in Information Retrieval [7] concluding that it demotes large documents. In this case, as the item prior is in the denominator, we are promoting, in a controlled way, items with a larger number of ratings. Moreover, it shows a quite stable performance with respect to the parameter μ.

4.3 Testing on the Other Collections

To assess if the previous findings generalise to other collections, we apply those priors to the R3-Yahoo! and the LibraryThing datasets. We also report the results of the standard user-based Collaborative Filtering algorithm (using the 50 nearest neighbours according to Pearson's correlation) denoted by UB and a standard matrix factorisation algorithm (using 350 latent factors) denoted by SVD. We present the nDCG@10 values of these recommendation methods in Table 2.

From this data, it can be seen that the use of the Linear prior for modelling neighbour prior probabilities and Probalistic prior (with Dirichlet smoothing) for items improves the ranking precision in the three collections. Nevertheless, the results obtained for the R3-Yahoo! dataset are not statistically significant according to the Wilcoxon test ($p < 0.01$).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have studied the effect that the user and item priors play in the RM2 algorithm, the most effective Relevance-Based Language Model estimate for Collaborative Filtering recommendation. This study has identified the Linear prior as the optimal one for modelling neighbourhoods. Additionally, the Probabilistic prior based on Dirichlet smoothing is a good choice for computing the item prior probability. We have found that the correct modelling of these prior probabilities can significantly improve the ranking accuracy in two out of three collections.

As future work, it would be interesting to analyse if the combination of different priors can lead to better results as it was shown in Information Retrieval [9]. Moreover, future research may study priors such as PageRank [4].

Acknowledgments

This work was funded by grant TIN2012-33867 from the Spanish Government and grants GPC2013/070, R2014/034 and R2014/002 from the Galician Government.

6. REFERENCES

[1] A. Bellogín, P. Castells, and I. Cantador. Precision-Oriented Evaluation of Recommender Systems. In *RecSys '11*, page 333, Oct. 2011.

[2] A. Bellogín, J. Parapar, and P. Castells. Probabilistic Collaborative Filtering with Negative Cross Entropy. In *RecSys '13*, pages 387–390, 2013.

[3] R. Blanco and A. Barreiro. Probabilistic Document Length Priors for Language Models. In *ECIR '08*, pages 394–405, 2008.

[4] S. Brin and L. Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *WWW '98*, volume 30, pages 107–117, 1998.

[5] W. Kraaij, T. Westerveld, and D. Hiemstra. The Importance of Prior Probabilities for Entry Page Search. In *SIGIR '02*, pages 27–34, 2002.

[6] V. Lavrenko and W. B. Croft. Relevance-Based Language Models. In *SIGIR '01*, pages 120–127, Sept. 2001.

[7] D. E. Losada and L. Azzopardi. An Analysis on Document Length Retrieval Trends in Language Modeling Smoothing. *Inf. Retr.*, 11(2):109–138, Dec. 2008.

[8] J. Parapar, A. Bellogín, P. Castells, and A. Barreiro. Relevance-based language modelling for recommender systems. *Inf. Process. Manage.*, 49(4):966–980, July 2013.

[9] J. Peng, C. Macdonald, B. He, and I. Ounis. Combination of Document Priors in Web Information Retrieval. In *RIAO '07*, pages 596–611, 2007.

[10] 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, 2015.

[11] 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, 2013.