

# A Study of Smoothing Methods for Relevance-Based Language Modelling of Recommender Systems

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.** Language Models have been traditionally used in several fields like speech recognition or document retrieval. It was only recently when their use was extended to collaborative Recommender Systems. In this field, a Language Model is estimated for each user based on the probabilities of the items. A central issue in the estimation of such Language Model is smoothing, i.e., how to adjust the maximum likelihood estimator to compensate for rating sparsity. This work is devoted to explore how the classical smoothing approaches (Absolute Discounting, Jelinek-Mercer and Dirichlet priors) perform in the field of Recommender Systems. We tested the different methods under the recently presented Relevance-Based Language Models for collaborative filtering, and compared how the smoothing techniques behave in terms of precision and stability. We found that Absolute Discounting is practically insensitive to the parameter value being an almost parameter-free method and, at the same time, its performance is similar to Jelinek-Mercer and Dirichlet priors.

**Keywords:** Recommender systems, Collaborative filtering, Smoothing, Relevance Models

## 1 Introduction

In a world with a growing amount of available information, Recommender Systems are key in satisfying the increasing demands of the users. These systems generate personalised suggestions saving the customers the time of searching for relevant information. Many approaches to recommendation have been proposed being collaborative filtering one of the most successful techniques. This family of methods exploits the past ratings of the users in order to generate recommendations. Parapar et al. recently proposed the use of Relevance-Based Language Models for collaborative filtering [5] obtaining superior figures in precision w.r.t. the state of the art methods. Following previous results in using Language Models for the recommendation task [6], the authors decided to use Jelinek-Mercer for smoothing the different probabilities arguing that Dirichlet priors can demote the weight of those items recently introduced in the systems. In this paper we tested those intuitions and analysed the performance of different smoothing

techniques in the recommendation task. For doing so, we followed the Zhai and Lafferty study of smoothing techniques for document retrieval [7] and adapted their methodology to collaborative filtering. We analysed Jelinek-Mercer, Dirichlet priors and Absolute Discounting smoothing in the context of Relevance-Based Language Models for collaborative filtering in different collections. We somehow obtained similar trends as in document retrieval but, in contrast, the better behaviour in terms of stability of Absolute Discounting makes it more suitable for the recommendation task than the other approaches. In the following sections, first, we briefly introduce Relevance Models for recommendation, then we present the smoothing techniques, the experimental conditions, the results and finally we conclude with some remarks about our findings and future work.

## 2 Relevance Models for Recommendation

Recommender Systems help users with the finding of relevant items. We denote the set of users by  $\mathcal{U}$  and the set of items by  $\mathcal{I}$ . We refer to the rating that the user  $u$  expressed to the item  $i$  by the notation  $r_{u,i}$ . Also,  $\mathcal{I}_u$  is used to indicate the set of items that were rated by the user  $u$ .

In the context of Statistical Language Models, Relevance-Based Language Models (usually referred as Relevance Models or RM) [3], are a pseudo relevance feedback technique for text retrieval. Given a query and a set of pseudo relevant documents, RM suggest terms to expand the original query and, thus, improve the text retrieval performance. Recently, RM has been applied as a collaborative filtering technique achieving high accuracy figures. Users play the role of both documents and queries whilst items are equivalent to the terms. In this way, we can expand users with new items as we expanded queries with new terms. To perform query expansion via RM, we need a set of pseudo relevant documents that, in this case, is the neighbourhood of the target user. Parapar et al. proposed the use of Posterior Probability Clustering (PPC [2]), a matrix factorization clustering algorithm, for calculating the neighbourhoods.

Two approaches of Relevance Models were proposed for recommendation: RM1 and RM2. Recommendations are generated by computing the Relevance Model of every user,  $R_u$ , and estimating the relevance of each item under it. The probability of an item  $i$  under the Relevance Model of the user  $u$  is shown in (1) and (2) for methods RM1 and RM2, respectively.

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

$$p(i|R_u) \propto p(i) \prod_{j \in \mathcal{I}_u} \sum_{v \in V_u} \frac{p(i|v)p(v)}{p(i)} p(j|v) \quad (2)$$

where  $V_u$  is the set of neighbours of the user  $u$ . Also,  $p(i)$  and  $p(v)$  are considered uniform. Finally, the probability of an item given a user  $p(i|u)$  can be computed by smoothing the maximum likelihood estimate  $p_{ml}(i|u)$ :

$$p_{ml}(i|u) = \frac{r_{u,i}}{\sum_{j \in \mathcal{I}_u} r_{u,j}} \quad (3)$$

### 3 Smoothing Methods in Recommendation

Smoothing is a well studied aspect of Language Models for text retrieval [4, 7]. The maximum likelihood estimator suffers from data sparsity, i.e., in the recommendation task each item is only rated by some users. Therefore, it is necessary to apply smoothing to adjust the estimator to prevent the apparition of zeros in (3). Furthermore, smoothing also plays a similar role to the *idf* (inverse document frequency).

In this paper, we studied the effect of applying three different smoothing methods for recommendation. All these techniques employ a background model which is the following collection model.

$$p(i|\mathcal{C}) = \frac{\sum_{v \in \mathcal{U}} r_{v,i}}{\sum_{j \in \mathcal{I}, v \in \mathcal{U}} r_{v,j}} \quad (4)$$

**Jelinek-Mercer.** JM performs a linear interpolation between the maximum likelihood estimator and the collection model controlled by the parameter  $\lambda$ .

$$p_\lambda(i|u) = (1 - \lambda) p_{ml}(i|u) + \lambda p(i|\mathcal{C}) \quad (5)$$

**Bayesian Smoothing with Dirichlet Priors.** DP uses Dirichlet priors for Bayesian analysis which results in the following expression with parameter  $\mu$ .

$$p_\mu(i|u) = \frac{r_{u,i} + \mu p(i|\mathcal{C})}{\mu + \sum_{j \in \mathcal{I}_u} r_{u,j}} \quad (6)$$

**Absolute Discounting.** AD subtracts a constant,  $\delta$ , from the count of the seen words.

$$p_\delta(i|u) = \frac{\max(r_{u,i} - \delta, 0) + \delta |\mathcal{I}_u| p(i|\mathcal{C})}{\sum_{j \in \mathcal{I}_u} r_{u,j}} \quad (7)$$

## 4 Experiments

### 4.1 Evaluation

We conducted our experiments in the *Movielens 100k*<sup>1</sup>, the *R3 Yahoo! Music*<sup>2</sup> and the *Movielens 1M*<sup>1</sup> collections. The statistics of these datasets are presented in the Table 1.

In this work, we considered the precision of the recommendations which is the fraction of items included in the recommendation list that are relevant to the user. We evaluated this metric at a cut-off rank of five, following the *TestItems* methodology described in [1].

<sup>1</sup> <http://grouplens.org/datasets/movielens>

<sup>2</sup> <http://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

Table 1: Datasets statistics

Dataset	#users	#items	#ratings	Sparsity
Movielens 100k	943	1682	100,000	6.30%
R3 Yahoo! Music	15400	1000	365,703	2.37%
Movielens 1M	6040	3952	1,000,209	4.19%

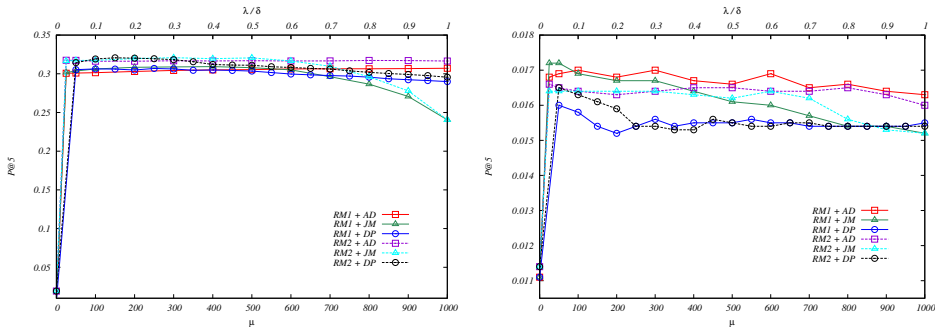


Fig. 1: Precision at 5 of the RM1 and the RM2 algorithms using Absolute Discounting (AD), Jelinek-Mercer (JM) and Dirichlet priors (DP) smoothing methods in the Movielens 100k (left) and the R3 Yahoo! Music (right) collections.

## 4.2 Results and Discussion

First, we studied the precision at five of the RM1 and the RM2 algorithms with the different smoothing techniques in the three collections. The results of the Movielens 100k and Yahoo datasets are illustrated in Fig. 1 (in this experiment, Movielens 1M presented the same trends as 100k). We have to remark that the precision values for the Yahoo dataset are low because of the very few of available testing ratings (only ten per user) which makes obtaining accurate recommendations a very hard task.

We notice that smoothing plays a key role in accuracy and a small amount of smoothing is sufficient to achieve good results. We can appreciate that Jelinek-Mercer performance deteriorates with a high value of  $\lambda$ . The same behaviour is observed when Dirichlet priors are applied, although on a lesser scale. It is very interesting that only AD does not present statistically significant differences (Wilcoxon test,  $p < 0.05$ ) in precision when varying the smoothing parameter, in contrast to JM and DP. This points out that the performance of the system when using JM or DP will be dependent on choosing the optimal smoothing value, which unfortunately depends on the data in collection, as can be observed in Fig. 1. In fact, in the R3 Yahoo! Music dataset, the demotion of precision when using DP and increasing the smoothing is more visible. Moreover, in this collection, RM1 works better than RM2. It seems that RM1 may be better for dealing with very sparse datasets, although further work is required to establish

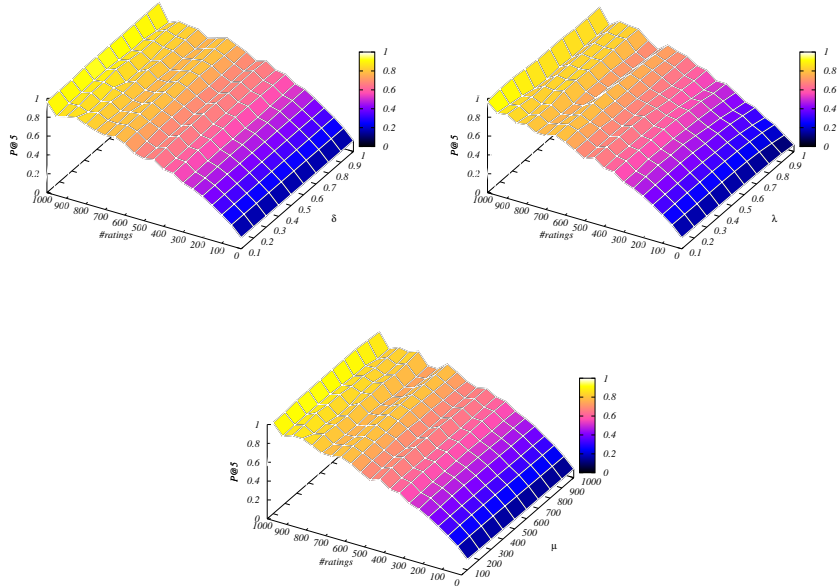


Fig. 2: Precision at 5 of RM2 algorithm using Absolute Discounting (top left), Jelinek-Mercer (top right) and Dirichlet priors (bottom) smoothing methods when varying the smoothing intensity and considering different  $|\mathcal{I}_u|$ , i.e., the number of ratings in the user profiles (they have been binned in steps of 50 and the average precision for each bin is plotted).

this. These trends are similar to the reported by Zhai and Lafferty for text retrieval [7], except for the fact that AD is more stable w.r.t. the smoothing parameter and that DP does not outperform the other methods.

The second experiment analyses the effect of the smoothing, in terms of precision, when recommending to users with different amount of rated items. Losada and Azzopardi have extensively studied the effects of the document length in text retrieval [4]. We aim to determine if there is such parallelism with the length of the user profiles. In the Fig. 2, we show the average precision achieved by the RM2 algorithm with each method when varying the intensity of smoothing and the size of the user profiles. The precision of the system improves with the number of rated items, achieving near perfect recommendations for users with a long rating history. The performance of DP and AD is very similar, although DP slightly degrades with high values of smoothing. The same effect, more intense, is observed in the case of the JM method. Additionally, JM does not seem to be a good technique for recommending to users with many ratings.

In the light of the results, we can recommend the use of AD because parameter optimization is not critical as long as a small amount of smoothing is applied. Furthermore, it obtains a good performance for each size of user profile.

## 5 Conclusions and Future Work

In this paper, we studied three techniques of Language Models smoothing in the context of Relevance Modelling for Recommender Systems. Through empirical analysis, we get insights of the behaviour of smoothing for the recommendation task. The evidence indicates that smoothing methods are crucial for achieving high precision: only with tiny values of the smoothing parameters the results are notably superior.

The current findings suggest that there is no big difference in terms of optimal precision among the studied smoothing techniques. However, Dirichlet priors and, specially, Jelinek-Mercer suffer a significant decrease in precision when a high amount of smoothing is applied, in contrast to Absolute Discounting. Thus, AD is the best election for a recommender system that makes use of Relevance Modelling because it saves the developers the time to tune properly the smoothing parameter for each domain and collection. An almost parameter-free smoothing method is very useful when no training data is available.

As a future work, it would be interesting to study how these smoothing methods behave w.r.t. different aspects such as novelty and diversity.

## 6 Acknowledgments

This work was funded by grants TIN2012-33867 and GPC2013/070 from the *Ministerio de Economía y Competitividad* and the Galician Government.

## References

1. Alejandro Bellogín, Pablo Castells, and Iván Cantador. Precision-oriented evaluation of recommender systems. In *RecSys '11*, page 333, New York, New York, USA, October 2011. ACM Press.
2. Chris Ding, Tao Li, Dijun Luo, and Wei Peng. Posterior probabilistic clustering using NMF. In *SIGIR '08*, pages 831–832, New York, NY, USA, 2008. ACM.
3. Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, New York, New York, USA, September 2001. ACM Press.
4. David E. Losada and Leif Azzopardi. An analysis on document length retrieval trends in language modeling smoothing. *Information Retrieval*, 11(2):109–138, April 2008.
5. Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. Relevance-based language modelling for recommender systems. *Information Processing & Management*, 49(4):966–980, July 2013.
6. Jun Wang. Language Models of Collaborative Filtering. In *AIRS '09*, volume 5839 of *Lecture Notes in Computer Science*, pages 218–229, Berlin, Heidelberg, October 2009. Springer Berlin Heidelberg.
7. Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, April 2004.