Computing Neighbourhoods with Language Models in a Collaborative Filtering Scenario

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 represent a successful framework for many Information Retrieval tasks: ad hoc retrieval, pseudo-relevance feedback or expert finding are some examples. We present how language models can compute effectively user or item neighbourhoods in a collaborative filtering scenario (this idea was originally proposed in [14]). The experiments support the applicability of this approach for neighbourhood-based recommendation surpassing the rest of the baselines. Additionally, the computational cost of this approach is small since language models have been efficiently applied to large-scale retrieval tasks such as web search.

1 Introduction

The goal of a recommender systems is to present relevant items to the users. Given the increasing amount of information, recommendation techniques have become crucial since they are able to alleviate the problem of information overload. These systems learn from the data provided by the users in order to satisfy their information needs.

Several approaches to recommendation have been proposed [9]. In particular, this work is focused on neighbourhood-based collaborative filtering techniques. Collaborative filtering aims to recommend items exploiting the past interaction between users and data [6,4]. These models are based on the wisdom of the crowds because items are considered as black boxes: they only rely on the historical data of the users of the system to generate recommendations. Collaborative filtering approaches can be divided in two main families: neighbourhood-based (or memory-based) methods [6] and model-based methods [4]. Instead of learning a predictive model from the data (as model-based methods do), neighbourhood-based approaches directly employ part of the interactions of the users to compute tailored suggestions. The main advantage of these models is their efficiency since they do not usually require a training phase. However, they are based on groups of users or items called user and item neighbourhoods, respectively. The most common approach to calculate neighbourhoods is based on k-NN algorithm. This technique assigns each user (or item) the k most similar users (or items) according to a pairwise similarity metric (popular choices are Pearson's correlation coefficient and cosine similarity) [6]. Previous work has studied that different similarities perform very differently on the top-N recommendation task [3].

2 Computing Neighbourhoods using Language Models

Language models (LM) have been extensively applied to several tasks within the Information Retrieval field [15]. The first use of these models in retrieval was proposed by Ponte and Croft when they presented the query likelihood model to rank documents in the ad hoc retrieval task [8]. Additionally, they have been used for other tasks such as query expansion via pseudo-relevance feedback (e.g., relevance-based language models [5]) or expert finding [1].

Language models are a formal approach that models a probability distribution over the occurrences of words. Given a query, documents are ranked as follows:

$$p(d|q) \overset{\text{rank}}{=} p(d)\, p(q|d) = p(d) \prod_{t \in q} p(t|d)^{c(t,d)} \tag{1}$$

where the document prior $p(d)$ is usually considered uniform. The probability $p(t|d)$ is estimated using a smoothed maximum likelihood estimate [15].

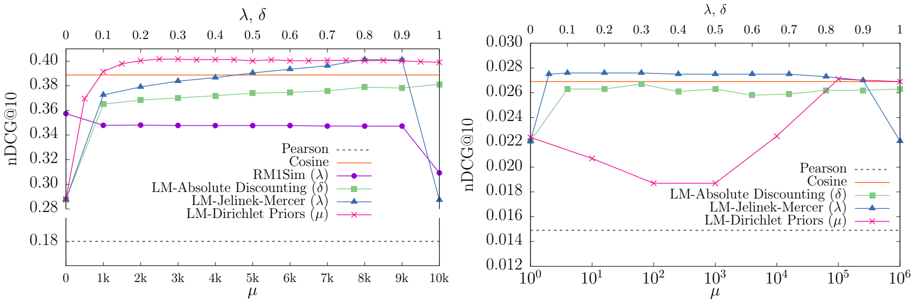
Following a recent and successful line of research of adapting Information Retrieval techniques to the recommendation [7,12,11,10,13], language models have been adapted the occurrences of ratings on user or item profiles [14]. Instead of inferring a language model for each document in the collection, we can formulate a language model for each user or item in the collection. In this way, the similarity between the user u and a candidate neighbour v can be measured as:

$$p(v|u) \overset{\text{rank}}{=} p(v)\, p(u|v) = p(v) \prod_{i \in \mathcal{I}_u} p(i|v)^{r_{u,i}} \tag{2}$$

where $\mathcal{I}_u$ are the items rated by user u and $r_{u,i}$ is the rating that the user u gave to the item i. The item-based similarity can be derived analogously. The conditional probabilities $p(i|v)$ are computed using a smoothing method over the maximum likelihood estimate of a multinomial distribution [12] with the probability in the collection $p(i|\mathcal{C})$. Table 1 describes these methods.

Method	**Expresion**
Absolute Discounting (AD)	$p_\delta(i|u) = \frac{\max(\mathrm{r}_{u,i} - \delta, 0) + \delta\, |\mathcal{I}_u|\, p(i|\mathcal{C})}{\sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}}$
Jelinek-Mercer (JM)	$p_\lambda(i|u) = (1-\lambda)\, \frac{\mathrm{r}_{u,i}}{\sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}} + \lambda\, p(i|\mathcal{C})$
Dirichlet Priors (DP)	$p_\mu(i|u) = \frac{\mathrm{r}_{u,i} + \mu\, p(i|\mathcal{C})}{\mu + \sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}}$

Table 1: Smoothed estimates of LM for computing neighbourhoods where $p(i|\mathcal{C}) = \sum_{v \in \mathcal{U}} \mathrm{r}_{v,i} / \sum_{j \in \mathcal{I},\, v \in \mathcal{U}} \mathrm{r}_{v,j}$.



(a) MovieLens 100k dataset, user-based approach, $k = 50$ neighbours

(b) R3-Yahoo! dataset, item-based approach, $k = 100$ neighbours

Fig. 1: Comparison in terms of nDCG@10 among different strategies for computing neighbourhoods. Recommendations are computed using WSR.

3 Experiments and Discussion

Using Weighted Sum Recommender (WSR [14]), a very simple and effective neighbourhood-based top-N recommender, we compare the three estimations of language models for computing neighbourhoods against Pearson and cosine similarities in terms of ranking accuracy (measured with nDCG@10). Also, we used RM1Sim [2] as a user-based baseline. We tested the user-based approach on the MovieLens 100k dataset[1] and the item-based approach on the R3-Yahoo! collection[2]. Figure 1 shows the results of the experiments.

Cosine similarity is the strongest baseline while Pearson's correlation coefficient performs very poorly. However, with the appropriate parameter tuning, Jelinek-Mercer and Dirichlet Priors methods can outperform cosine. Since the R3-Yahoo! dataset is more sparse than MovieLens 100k, we need to put a higher amount of smoothing to obtain good results. Additionally, the computational complexity of these methods is linear in the size of the user (or item) profiles which is the same as cosine or Pearson's.

4 Conclusions and Future Work

We have presented how language models can compute user or item neighbourhoods in a collaborative filtering scenario. Using Jelinek-Mercer and Dirichlet Priors smoothing methods, we can outperform all the baselines. Additionally, this approach is efficient and can make use of inverted indexes and other data structures used in Information Retrieval to deal with large-scale scenarios. We have used a uniform estimate for the user and item priors. However, it would be interesting to explore other estimates since prior probabilities have been recognised as a useful way of improving recommendation quality [11].

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

[2] `https://webscope.sandbox.yahoo.com/catalog.php?datatype=r`

Acknowledgments. This work was supported by the Ministerio de Economía y Competitividad of the Goverment of Spain and FEDER Funds under the research project 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. Krisztian Balog, Leif Azzopardi, and Maarten de Rijke. A Language Modeling Framework for Expert Finding. *Inf. Process. Manage.*, 45(1):1–19, 2009.
2. Alejandro Bellogín, Javier Parapar, and Pablo Castells. Probabilistic Collaborative Filtering with Negative Cross Entropy. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 387–390, New York, NY, USA, 2013.
3. Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of Recommender Algorithms on Top-N Recommendation Tasks. In *RecSys '10*, pages 39–46, 2010.
4. Yehuda Koren and Robert Bell. Advances in Collaborative Filtering. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 77–118. Boston, MA, 2nd edition, 2015.
5. Victor Lavrenko and W. Bruce Croft. Relevance-Based Language Models. In *SIGIR '01*, pages 120–127, 2001.
6. Xia Ning, Christian Desrosiers, and George Karypis. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 37–76. 2nd edition, 2015.
7. Javier Parapar, Alejandro Bellogín, Pablo Castells, and Álvaro Barreiro. Relevance-Based Language Modelling for Recommender Systems. *Inf. Process. Manage.*, 49(4):966–980, 2013.
8. Jay M. Ponte and W. Bruce Croft. A Language Modeling Approach to Information Retrieval. In *SIGIR '98*, pages 275–281, 1998.
9. Francesco Ricci, Lior Rokach, and Bracha Shapira. *Recommender Systems Handbook*. 2nd edition, 2015.
10. Daniel Valcarce. Exploring Statistical Language Models for Recommender Systems. In *RecSys '15*, pages 375–378, 2015.
11. Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. A Study of Priors for Relevance-Based Language Modelling of Recommender Systems. In *RecSys '15*, pages 237–240, 2015.
12. Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. A Study of Smoothing Methods for Relevance-Based Language Modelling of Recommender Systems. In *ECIR '15*, volume 9022, pages 346–351, 2015.
13. Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. Efficient Pseudo-Relevance Feedback Methods for Collaborative Filtering Recommendation. In *ECIR '16*, pages 602–613, 2016.
14. Daniel Valcarce, Javier Parapar, and Álvaro Barreiro. Language Models for Collaborative Filtering Neighbourhoods. In *ECIR '16*, pages 614–625, 2016.
15. ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Synthesis lectures on human language technologies. 2009.