Exploring Statistical Language Models for Recommender Systems

Daniel Valcarce
Information Retrieval Lab
Department of Computer Science
University of A Coruña, Spain
daniel.valcarce@udc.es

ABSTRACT

Even though there exist multiple approaches to build recommendation algorithms, algebraic techniques based on vector and matrix representations are predominant in the field. Notwithstanding the fact that these algebraic Collaborative Filtering methods have been demonstrated to be very effective in the rating prediction task, they do not generally provide good results in the top-N recommendation task. In this research, we return to the roots of recommender systems and we explore the relationship between Information Filtering and Information Retrieval. We think that probabilistic methods taken from the latter field such as statistical Language Models can be a more effective and formal way for generating personalised ranks of recommendations. We compare our improvements against several algebraic and probabilistic state-of-the-art algorithms and pave the way to future and promising research directions.

Categories and Subject Descriptors

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

Keywords

Recommender systems; Language Models.

1. INTRODUCTION

The goal of an Information Retrieval (IR) system is to retrieve the relevant pieces of information according to an information need, typically in the form of a query. Prior to the arrival of the Web, IR was a narrow area of research —only the small part of the society that had access to digital libraries, librarians and information experts, were interested on IR techniques [1]. Nevertheless, nowadays, IR has become a vital part of the Web: the difficulty of finding relevant information in the largest repository of knowledge makes imperative the use of specialised techniques.

On the other hand, Information Filtering (IF) consists in selecting relevant items for the users from an information stream [9]. Passive IF system remove unwanted pieces of information. For example, anti-spam techniques are passive filters that keep only useful messages. In contrast, active IF systems push relevant information to the users. Nowadays, recommenders are probably the most prominent type of active information filters. Recommenders deliver suggestions to the users based on their profiles. There exist several approaches to build recommenders system. Traditionally, they are classified in three main categories: Collaborative Filtering (CF), Content-Based (CB) and hybrid techniques [14]. CB methods use the features of the items that are part of the user's profile to find similar items that may be of interest. On the other hand, CF algorithms exploit the recorded data about the interactions betweens users and items (ratings, clicks, etc.). Finally, hybrid methods are the fruit of the combination of techniques from the previous approaches.

In the end, both IR and IF have the same objective: provide relevant information to the users. The main difference lies largely in the representation of the information need: a traditional IR system employs an explicit query prompted by the user while an IF system uses the user's profile. Therefore, some authors consider IF as a part of IR [3], meanwhile others think that they are two sibling fields [9]. In any case, in spite of the similarities between IR and IF, there has been little research about applying classic IR techniques to recommender systems, specially to Collaborative Filtering recommenders.

Many CF algorithms rely on finding neighbours using vector similarities [8], in a similar way to the Vector Space Model from IR [1]. However, matrix factorisation methods, such as SVD, are the most popular techniques in the recommender systems literature [11, 6]. These algebraic approaches rely on computing low-rank approximations of the user-item matrix. SVD was already used in text retrieval under the name of Latent Semantic Indexing [7], however its effectiveness is now surpassed by other techniques [1].

The introduction of probabilistic models represented a breakthrough in IR. In particular, statistical Language Models have become a state-of-the-art technique for the text retrieval task [20]. We consider that these models with a solid statistical foundation may bring significant improvements to the field of Recommender Systems as it did in IR.

In our work, we aim to explore different probabilistic IR techniques (specially statistical Language Models) for recommendation tasks. Mostly, but not exclusively, we would want to investigate the following questions:

RecSys'15, September 16–20, 2015, Vienna, Austria.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2796547

- Are probabilistic IR models suitable to tackle Collaborative Filtering tasks such as neighbourhood finding or item ranking?
- Can probabilistic IR models be adapted to deal with temporal and/or extra contextual information?
- Is there a principled formulation of statistical Language Models that effectively merges Content-Based and Collaborative Filtering approaches?

2. RELATED WORK

It was acknowledged that rating prediction does not model effectively the recommendation task since users are interested in obtaining a short list of relevant items, they are not particularly concerned about the predicted rating values [10, 6]. This is known as the top-N recommendation task [6]. An advantage of the Information Retrieval methods is that they are traditionally focused on generating a ranked list of items. Thus, there exists an emerging interest in applying IR techniques to the field of Recommender Systems [18, 17, 5, 13].

Recently, the performance of probabilistic graphical models for Collaborative Filtering tasks has been analysed [2] showing better results than other algebraic state-of-the-art recommenders [6].

Aiming to unify IR with recommenders systems, Bellogín et al. presented a general framework that is able to employ any IR system for generating CF recommendations [5]. These methods showed better figures than traditional CF algorithms in terms of precision and ranking metrics.

Another approach is the proposal of Wang et al. based on the probability ranking principle [18]. Wang also derived a CF method utilising Language Models using a risk-averse model that penalises less reliable scores [17]. Nevertheless, these methods are intended to employ implicit feedback.

An idea that showed very satisfactory results is to define the CF task as a Pseudo-Relevance Feedback problem using a specific analogy between IR and IF and applying Relevance-Based Language Models [13]. Following this line of research, we plan to continue to explore the relationship between these fields.

3. RELEVANCE-BASED LANGUAGE MODELS FOR CF

Parapar et al. adapted the Pseudo-Relevance Feedback framework to the Collaborative Filtering recommendation task [13]. Pseudo-Relevance Feedback is a family of methods for expanding the user's query with new terms to improve the performance of a retrieval system. The terms are extracted from a set of documents that are assumed to be relevant. This pseudo-relevant set is obtained from the top documents of an initial retrieval. A second retrieval is performed using the new expanded query and its results are the ones presented to the user.

The Pseudo-Relevance Feedback scheme can be tailored to Collaborative Filtering as follows. Users have a dual role: they act as queries when they are the target user of the recommendations but they also act as documents. On the other hand, items are modelled as terms (they can be query terms as well as document terms). The neighbourhood of the target user is modelled as the pseudo-relevant set. In this way, the problem of recommending items to users becomes the task of expanding queries with novel terms taken from their neighbours.

In the original paper, they applied Relevance-Based Language Models to CF recommendation outperforming the state of the art [13]. Relevance-Based Language Models (or RM for short) [12] are a very effective method for Pseudo-Relevance Feedback. In particular, RM2 model presents high figures in accuracy:

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

Given a set of $\mathcal{U}$ users and $\mathcal{I}$ items, a relevance model R_u is computed for each user $u \in \mathcal{U}$ and the relevance of each item $i \in \mathcal{I} \setminus \mathcal{I}_u$ in this model is estimated, $p(i|R_u)$. $\mathcal{I}_u$ is used for representing the set of items the user u rated. Also, V_u refers to the neighbourhood of the user u. Additionally, the prior probabilities, the neighbour prior $p(v)$ and the item prior $p(i)$, should be estimated. Finally, the conditional probability estimations, $p(i|v)$ and $p(j|v)$, are obtained smoothing the Maximum Likelihood Estimate (MLE) with the probability in the collection. The MLE is calculated as follows:

$$p_{ml}(i|u) = \frac{r_{u,i}}{\sum_{j \in \mathcal{I}_u} r_{u,j}} \tag{2}$$

whereas the probability in the collection is:

$$p(i|\mathcal{C}) = \frac{\sum_{v \in \mathcal{U}} r_{v,i}}{\sum_{j \in \mathcal{I},\, v \in \mathcal{U}} r_{v,j}} \tag{3}$$

The notation $r_{u,i}$ denotes the rating that the user u gave to the item i.

4. ONGOING WORK

In this section, we present our current work on the Relevance-Based Language Modelling of recommender systems. The previous RM2 formula (see Eq. 1) contains different probabilities that should be estimated. First, we describe our findings about smoothing methods for estimating the conditional probabilities of RM2 and, then, we explore some prior estimates.

4.1 Smoothing Methods

We started by analysing different smoothing methods for estimating the conditional probabilities of RM2 [16]. We considered three smoothing techniques.

Jelinek-Mercer.

It performs a linear interpolation between the MLE (see Eq. 2) and the collection model (see Eq. 3) controlled by the parameter λ. This method was used in the original work of RM2 for recommendation [13].

$$p_\lambda(i|u) = (1-\lambda)\, p_{ml}(i|u) + \lambda\, p(i|\mathcal{C}) \tag{4}$$

Dirichlet priors.

It utilises Dirichlet priors for Bayesian analysis yielding the following expression with parameter μ:

$$p_\mu(i|u) = \frac{\mathrm{r}_{u,i} + \mu\, p(i|\mathcal{C})}{\mu + \sum_{j \in \mathcal{I}_u} \mathrm{r}_{u,j}} \tag{5}$$

Absolute Discounting.

This method subtracts a constant, δ, from each rating.

$$p_\delta(i|u) = \frac{\max(\mathrm{r}_{u,i} - \delta, 0) + \delta\,|\mathcal{I}_u|\,p(i|\mathcal{C})}{\sum_{j\in\mathcal{I}_u} \mathrm{r}_{u,j}} \tag{6}$$

In previous experiments [16], we found out that Absolute Discounting is the best smoothing method. Not only does it yield better rankings but it is also a very stable technique: different values of the parameter barely modify the performance of the recommendation algorithm. Moreover, Absolute Discounting effectively tackles the user bias taking into account the average rating of each user.

4.2 Prior Probabilities

One advantage of the RM2 probabilistic algorithm over other algebraic proposal is its interpretability. The modelling of prior probabilities provide a principled way of introducing business rules into the recommendation algorithm. The original proposal of RM2 for Collaborative Filtering employed uniform priors [13]; however, it would be interesting to explore another approaches. Next, we describe the standard uniform prior and a new proposal for user priors. More work in this line of research is being performed.

Uniform Prior.

This classic prior is drawn from a uniform distribution: we assign the same probability to each user.

Linear Prior.

This prior promotes those users with larger rating profiles. The rationale behind this decision is that we should rely more on the users for which we have more information about.

$$p_L(u) = p(u|\mathcal{C}) = \frac{\sum_{i\in\mathcal{I}_u} r_{u,i}}{\sum_{v\in\mathcal{U}}\sum_{j\in\mathcal{I}_v} r_{v,j}} \tag{7}$$

5. EVALUATION AND RESULTS

The evaluation of recommender systems is crucial in order to choose between models or to tune parameters. In this section, we present the evaluation methodology followed in our experiments that we plan to apply during the development of the PhD thesis. In addition, we performed a series of experiments to analyse the performance RM2-based methods against other recommendation approaches. We chose to report these experiments only on the *MovieLens 100k*[1] dataset, a well-known film collection for Collaborative Filtering, because of space reasons.

5.1 Evaluation methodology

We aim to analyse the performance of recommender systems in the top-N recommendation task, that is, measuring how well recommenders put relevant items in the top of the list [6]. Consequently, we are interested in precision-oriented metrics that analyse only the top N recommendations. With that purpose in mind, we followed the *TestItems* approach described in [4] consisting in scoring, for each user, every item included in the test set. We considered a recommendation relevant if the item is rated by the user in the test set. Although this methodology is very restrictive and may underestimate the true value of the metric, it provides comparable and trustworthy results [4].

Table 1: Values of nDCG@10, Gini@10 and MSI@10 on the MovieLens 100k collection for different CF algorithms.

Algorithm	**nDCG@10**	**Gini@10**	**MSI@10**
SVD	0.09456	0.01094	14.61295
SVD++	0.11126	0.01264	14.95739
NNCosNgbr	0.17710	0.03440	16.82222
UIR-Item	0.21876	0.01242	5.23371
PureSVD	0.35946	0.13645	11.88408
RM2-DP	0.29226	0.01760	6.05446
RM2-JM	0.31748	0.02323	6.69447
RM2-AD	0.32964	0.02561	6.82732
RM2-AD-L	0.34232	0.02644	6.78484

Not only ranking accuracy but also novelty and diversity are crucial aspects in recommendation [10]. We used nDCG (Normalised Discounted Cumulative Gain) for assessing the quality of the ranking [19], we employed the complement of the Gini index as a measure of recommendation diversity [15] and, finally, we utilised MSI (Mean Self-Information) as a tool for quantifying novelty [21]. These metrics can be measured at a given cut-off, that is, considering only the top results which are the ones presented to the user.

5.2 Experiments

We compared the performance of RM2-based methods against several state-of-the-art recommendation algorithms. All of these techniques were tuned to optimise the values of nDCG@10. First, we describe the baselines and, then, the proposed algorithms. The results are presented in Table 1.

We used the common SVD and SVD++ approaches to rating prediction with 400 dimensions [11]. We also used NNCosNgbr, a nearest neighbour approach oriented to top-N ranking recommendation [6]. For this method, we used cosine similarity with a shrinking factor of 100 and a L2 regularization factor of 0.9 for computing the user and item biases. Additionally, we implemented the probabilistic approach (UIR-Item) proposed in [18] with $\lambda = 0.5$. Finally, we tested PureSVD (with 50 dimensions) because Cremonesi et al. in [6] showed that it is a very effective matrix factorisation algorithm for top-N recommendation. It computes a global factorisation using the well-known SVDLIBC, in contrast with SVD and SVD++ methods that only minimise the error on the known ratings.

Turning to our probabilistic approaches, we analysed the performance of RM2-based methods using the 400 nearest neighbours according to the Pearson's correlation coefficient. We tested RM2 using Dirichlet Priors (RM2-DP) with $\mu = 100$, Jelinek-Mercer (RM2-JM) with $\lambda = 0.1$ and Absolute Discounting (RM2-AD) with $\delta = 0.1$. In addition, we chose the best smoothing method, Absolute Discounting, and we used a linear prior for estimating $p(v)$ in Eq. 1. We denoted this approach by RM2-AD-L.

All the differences in nDCG@10 reported in Table 1 are statistically significant according to the Wilcoxon signed-rank test ($p < 0.01$). As it was expected, rating prediction methods (SVD and SVD++) performed poorly compared to the rest of the techniques. It can be pointed out that RM2 methods provided higher figures in nDCG@10 than the rest of the baselines expect for PureSVD. This latter method is still the best algorithm in terms of nDCG. These findings confirmed the precision and recall results presented in [6].

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

Additionally, PureSVD also showed great figures of diversity and novelty. Nevertheless, it is important to note that RM2 provide another advantages such as interpretability and a principled way of introducing business rules into the model using prior probabilities.

It is interesting to remark that there is still room for improvement in the Relevance-Based Language Modelling framework where we plan to work on. The use of the correct smoothing method or the adequate prior estimate can boost significantly the quality of the recommendations. With regard to diversity and novelty, we can notice that PureSVD provides very good values. Thus, it would be worthwhile to explore how to improve RM2 in those aspects.

6. CONCLUSIONS AND FUTURE WORK

The probabilistic modelling of recommender systems is a broad area that has reached little attention. Probabilistic models provided significant improvements to the IR task. Since IR and IF are two closed fields, we think that probabilistic approaches may also lead to important improvements to the recommendation task. In this paper, we explore the performance of Relevance-Based Language Models against several state-of-the-art algorithms. We discovered that RM2 is superior to all baselines except for PureSVD, a recent matrix factorisation approach designed for top-N recommendation.

The experiments showed that different smoothing methods can lead to significant improvements. Moreover, we devised a prior estimate that enhanced the quality of the recommendations. We intend to further study this topic.

Additionally, as future work, we also envision to study other clustering techniques for computing neighbourhoods. In this work, we employed the simple k-NN algorithm using Pearson's correlation coefficient. Further investigation on clustering algorithms may shed light on how to increase the diversity and novelty figures of this probabilistic approach. Aspects such as user and item biases, temporal dynamics or modelling item content and user context information are still open research lines in this probabilistic modelling approach of the recommendation task. We think it is feasible to merge CF and CB approaches in a principle manner using a probabilistic framework.

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.

7. REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison Wesley, 2011.

[2] N. Barbieri and G. Manco. An Analysis of Probabilistic Methods for Top-N Recommendation in Collaborative Filtering. In *ECML PKDD '11*, pages 172–187, 2011.

[3] N. J. Belkin and W. B. Croft. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Commun. ACM*, 35(12):29–38, Dec. 1992.

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

[5] A. Bellogín, J. Wang, and P. Castells. Bridging Memory-Based Collaborative Filtering and Text Retrieval. *Inf. Retr.*, 16(6):697–724, Nov. 2013.

[6] P. Cremonesi, Y. Koren, and R. Turrin. Performance of Recommender Algorithms on Top-n Recommendation Tasks. In *RecSys '10*, pages 39–46. ACM, Sept. 2010.

[7] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by Latent Semantic Analysis. *JASIST*, 41(6):391–407, 1990.

[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] U. Hanani, B. Shapira, and P. Shoval. Information Filtering: Overview of Issues, Research and Systems. *User Model. User-Adapt. Interact.*, 11(3):203–259, 2001.

[10] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, Jan. 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, Sept. 2001.

[13] 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.

[14] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook*. Springer-Verlag New York, Inc., 1st edition, 2010.

[15] G. Shani and A. Gunawardana. Evaluating Recommendation Systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*, pages 257–297. Springer US, 2011.

[16] 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.

[17] J. Wang. Language Models of Collaborative Filtering. In *AIRS '09*, pages 218–229. Springer-Verlag, 2009.

[18] J. Wang, A. P. de Vries, and M. J. Reinders. A User-Item Relevance Model for Log-based Collaborative Filtering. In *ECIR '06*, volume 3936, pages 37–48. Springer-Verlag, 2006.

[19] 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.

[20] C. Zhai. *Statistical Language Models for Information Retrieval*. Synthesis lectures on human language technologies. Morgan & Claypool, 2009.

[21] 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, Mar. 2010.