Text retrieval methods for item ranking in Collaborative Filtering

Alejandro Bellogín[1], Jun Wang[2], and Pablo Castells[1]

[1] Universidad Autónoma de Madrid
Escuela Politécnica Superior
Francisco Tomás y Valiente, 11, 28049 Madrid, Spain
{alejandro.bellogin,pablo.castells}@uam.es,
[2] Department of Computer Science
University College London
Male Place, London, WC1E 6BT, UK
jun.wang@cs.ucl.ac.uk

Abstract. Collaborative Filtering (CF) aims at predicting unknown ratings of a user from other similar users. The uniqueness of the problem has made its formulation distinctive to other information retrieval problems. While the formulation has proved to be effective in rating prediction tasks, it has limited the potential connections between these algorithms and Information Retrieval (IR) models. In this paper we propose a common notational framework for IR and rating-based CF, as well as a technique to provide CF data with a particular structure, in order to be able to use any IR weighting function with it. We argue that the flexibility of our approach may yield to much better performing algorithms. In fact, in this work we have found that IR models perform well in item ranking tasks, along with different normalization strategies.

Keywords: Collaborative Filtering, Text Retrieval, Unified Models

1 Introduction

Recommender Systems (RS) suggest *interesting* items to users by taking into account users' profiles. Interestingly, although from the beginning IR techniques have been used in RS and their underlying goals are essentially equivalent [2], no exact equivalences have been established between the models and structures used in IR and those in RS. In particular, we are interested in Collaborative Filtering (CF), one of the most extended types of RS. Specifically, in this work we aim at answering the following research questions: (**RQ1**) is it possible to use IR models in the rating-based CF framework? and, in that case (**RQ2**) are IR formulations of CF better or worse than classic CF algorithms? We believe these questions are important because they would allow a better understanding of CF strategies. Furthermore, IR researchers could design better and sound recommender systems using their knowledge on IR and a proper mapping between CF data and IR structures.

Table 1. Query and document weighting components for different retrieval methods.

Method	w_t^q	w_t^d
Binary	1 if $t \in q$	1 if $t \in d$
TF dot	$\mathrm{qf}(t)$	$\mathrm{tf}(t,d)$
TF-IDF	$\mathrm{qf}(t)$	$\mathrm{tf}(t,d) \log\left(\frac{N}{\mathrm{df}(t)}\right)$
BM25	$\frac{(k_3+1)\,\mathrm{qf}(t)}{k_3+\mathrm{qf}(t)}$	$\log\left(\frac{N-\mathrm{df}(t)+0.5}{\mathrm{df}(t)+0.5}\right) \frac{(k_1+1)\,\mathrm{tf}(t,d)}{k_1\left((1-b)+b\cdot\mathrm{dl}(d)/\overline{\mathrm{dl}}\right)+\mathrm{tf}(t,d)}$

Our main contribution is a common notational framework for IR and rating-based CF, which provides a general retrieval function adopting any text retrieval weighting function with CF preference data. We also evaluate how well IR methods perform against standard techniques when applied to item ranking, that is, returning a ranking for each user. We have found that IR methods perform particularly well in this task (which is actually equivalent to the classic ad-hoc IR task), whereas different normalization strategies also provide good results.

2 A New Collaborative Filtering Framework

Recommender Systems have usually been seen as an IR technique applied when no explicit query has been provided, but a user profile is known instead. However, these two areas have been developed independently. Recently, some works have started to seek explicit links between one area and the other [3, 4, 9]. Rating-based CF is not yet fully integrated with IR, mainly because the input data and the final goal are different: in IR we have query and documents represented by terms, while in rating-based CF we have a set of ratings, which are used to infer the preferences of users towards items, where how to represent the users or the items is unclear. In the next sections, we define a general framework for these two areas, presenting a novel formulation for CF. Based on this framework, we propose a unification of the rating-based CF framework with a text-oriented IR framework, thus enabling the application of classic IR models to a CF system. With this formulation, we can also study different normalisation techniques, beyond those historically used in classic strategies for CF.

2.1 A General Text Retrieval Framework

Representing documents and queries as vectors in a common space is known as the vector space model (VSM) [8] and is fundamental to different IR operations [6]. Documents are represented in the vocabulary space using the bag of words approach, in which each document can be described based on the words occurrence frequency (captured by tf or term frequency), so $\boldsymbol{d}^i = [\mathrm{tf}(t_1, d^i), \ldots, \mathrm{tf}(t_T, d^i)]$ is the vector representation for document d^i, while $\boldsymbol{q} = [\mathrm{qf}(t_1), \ldots, \mathrm{qf}(t_T)]$ is the representation for a query q. Since many plausible weighting functions can be used to represent the importance of a term in a document (query), we keep our

formulation general by representing this as follows: $w(\boldsymbol{d}^i) = [w_1^i, \ldots, w_T^i]$, and $w(\boldsymbol{q}) = [w_1^q, \ldots, w_T^q]$, where T is the number of terms in the collection and w_j^i is the weight of term t_j in document d^i (or in the query, for w_j^q).

On top of the representation model, the core function in any retrieval system is the scoring of a document for a particular query. In the VSM, since queries and documents live in the same space we may use the dot product to obtain a score between a query and a document. We can generalise and formalise this as follows:

$$s(q, d) = \sum_{t \in g(q)} s(q, d, t) \tag{1}$$

where the function $g(\cdot)$ returns the term components of a query, and $s(q, d, t)$ is a function of a term, the query and the document. If $s(q, d, t) = w_t^q \cdot w_t^d$ we get the dot product, which is a simpler but fairly generic formulation (it receives the name of *factored form* in [7]). In Table 1 we can see several examples of different weighting functions; further functions could also be represented this way, such as the ones in language models (as proposed in [7]).

2.2 A General Collaborative Filtering Framework

Rating-based CF algorithms deal directly with the set of ratings given by the user community to a set of items. Let us represent as r_i^u the rating assigned to item i by user u, where $r_i^u \in \{1, \ldots, R, \perp\}$, assuming a rating scale from 1 to R, the symbol $\perp$ representing an unknown rating value. The main goal in a rating-based CF system is to predict the user rating for an unknown item, that is, to find the most accurate prediction $\hat{r}_i^u$ [1]. This can be formulated in a fairly general way by the following equation:

$$\hat{r}_i^u = \sum_{e \in h(u)} f(u, i, e) \tag{2}$$

where the functions f and h depend on the CF strategy (user- or item-based). More specifically, in the item-based approach [1] we have $h(u) = I_u$, the subset of items rated by user u, and

$$f(u, i, e) = \frac{\mathrm{sim}(i, e)}{\sum_{e \in h(u)} |\mathrm{sim}(i, e)|} r_e^u \tag{3}$$

A popular similarity function $\mathrm{sim}(i, e)$ is the Pearson correlation [5]. We can find an analogy between Eq. 2 and 1 simply by equating users to queries and items to documents. In fact, similarly to what we did in the previous section, we may also split up function f in two parts: one depending exclusively on the user (f_e^u) and another on the target item (f_e^i), in such a way that $f(u, i, e) = f_e^u \cdot f_e^i$, obtaining a dot product too. In Table 2 we show the different possible values for these components according to the CF strategy (user- or item-based). For the user-based approach, the similarity function $\mathrm{sim}(i, j)$ between two items can also

Table 2. User and item components for function f in user- and item-based CF. E represents the space where e belongs, that is, $e \in E$.

Approach	f_e^u	f_e^i	E	w_e^u	w_e^i
User-based	$\frac{\text{sim}(u,e)}{\sum_{e \in N[u]} \lvert \text{sim}(u,e) \rvert}$	r_i^e	users	$\text{sim}(u,e)$	r_i^e
Item-based	r_e^u	$\frac{\text{sim}(i,e)}{\sum_{e \in I_u} \lvert \text{sim}(i,e) \rvert}$	items	r_e^u	$\text{sim}(i,e)$

be calculated, for instance, using Pearson correlation, and function $h(u) = N[u]$ is the set of neighbours for user u.

2.3 Unifying Text Retrieval and Collaborative Filtering

In the last two sections we have presented two general scoring frameworks for text retrieval and rating-based CF under a unified formulation. Now we will explicitly derive the relation between these two frameworks, that is, we define how the IR models listed in Table 1 can be used in the framework defined in Section 2.2. The simplest way to do this is to define what tf and qf mean in the CF space and then apply the formulas for $w(\boldsymbol{q})$ and $w(\boldsymbol{d}^i)$ shown in Section 2.1.

It can be shown that taking $\text{qf}(t) = w_e^u$ and $\text{tf}(t,d) = w_e^i$ as defined in Table 2, we can obtain equivalent scoring functions to those defined by standard CF algorithms (such as Eq. 3), specifically, when applying the TF model. This implies a positive answer for **RQ1**, since it is possible to make an equivalence between a rating-based item-based CF system and an IR system by means of identifying the terms with the items in the collection, the frequency of a term in the query with the rating assigned by a user to that item, and the frequency of a term in a document with the similarity between the two items involved. Equivalence with respect to user-based CF can be found analogously.

Now, we may rephrase the second research question **RQ2** we address in this paper as: can other IR models, different from basic TF, obtain better results than standard CF techniques (which are equivalent to the TF model)? In the next section, we answer this question.

3 Empirical Evaluation

Rating-based CF is generally used for predicting an unknown rating, and the methods are consequently evaluated using error metrics. In contrast, since our approach applies to item ranking task, it needs to be evaluated based on precision metrics. This is more similar to how IR algorithms are usually evaluated.

Our experiments have been carried out using two different datasets from Movielens[3]: *Movielens 100K* and *Movielens 1M*. In our evaluation, we performed a 5-fold cross validation where each split takes 80% of the data for training, and

[3] www.grouplens.org/node/73

Table 3. Results for different normalisation functions in the item ranking task for item-based (*Movielens 1M*). Standard CF algorithm is underlined, † represents the best method for each normalisation method, ‡ represents the best performing method. Typical values are used for constants ($k_1 = 1.2, k_3 = 8$).

(a) Normalization s_{00}

Method	nDCG	MAP	P@10
BM25	0.10	0.00	0.00
TF-IDF	0.15	0.01	0.01
TF	0.19†	0.01	0.01

(b) Normalization s_{10}

Method	nDCG	MAP	P@10
BM25 L_1	0.23†	0.02†	0.00
TF-IDF L_1	0.10	0.01	0.00
TF L_1	0.07	0.00	0.00
BM25 L_2	0.16	0.01	0.00
TF-IDF L_2	0.13	0.01	0.01†
TF L_2	0.16	0.01	0.00

(c) Normalization s_{01}

Method	nDCG	MAP	P@10
BM25 L_1	0.09	0.01	0.00
TF-IDF L_1	0.05	0.00	0.00
TF L_1	0.05	0.00	0.00
BM25 L_2	0.24	0.03	0.03
TF-IDF L_2	0.29	0.05	0.07
TF L_2	0.34‡	0.07‡	0.10‡

(d) Normalization s_{11}

Method	nDCG	MAP	P@10
BM25 L_1	0.21	0.02	0.02†
TF-IDF L_1	0.01	0.00	0.00
TF L_1	0.00	0.00	0.00
BM25 L_2	0.06	0.00	0.00
TF-IDF L_2	0.05	0.00	0.00
TF L_2	0.27†	0.03†	0.00

the rest for testing. The item ranking task was performed by predicting a rating score for all the items contained in the test set for the current user.

We evaluated our approach in the user- and item-based versions, although due to space constraints we only show here results for the item-based algorithms. We used four different normalisation techniques and two norms: L_1 and L_2. These techniques are denoted as s_{QD}, where Q and D are 0 or 1, depending on which vectors (query or document) are used for normalization. For example, Eq. 3 is the same as s_{01} when the L_1 norm is used.

In Table 3 we present results from the item ranking task in the *Movielens 1M* dataset. For the experiments, we assume vectors $w(\boldsymbol{q})$ and $w(\boldsymbol{d})$ are defined as in Section 2.1, where values w_j^d and w_j^q are given by any retrieval method we want to use (see Table 1). Besides that, the interpretation of qf and tf is taken as in Section 2.3. We can see that for each normalization strategy there is at least one method outperforming the baseline. Moreover, BM25 obtains very good results in some of these situations, while the TF method performs better with the L_2 norm. Note that neither of these two methods matches the standard one used in the CF literature. We have obtained very similar results with the *Movielens 100K* dataset, including the best performing algorithms in each situation.

From these experiments, we can see that there can be better IR-based formulations than classic ones used in CF, providing a positive answer to **RQ2** in those cases. Besides, since other norms, as well as different weighting functions, appear naturally in our framework, they have also been tested with good results.

4 Conclusion and Future Work

We have proposed a generalised model for ranking items in rating-based CF which can fit many different algorithms, including different normalisation techniques and weighting functions commonly used in IR models. An analogy between IR and rating-based CF has been found: in item-based CF, terms are seen as the items, while the term frequencies are the user ratings (in the query representation) or the item similarity (in the document representation). As a result, it is possible to directly apply IR models in our framework with comparable or better results than classic CF formulations. Besides that, different normalisation techniques can fit into our framework and also lead to good results; furthermore, as in IR, different frequency normalisation techniques can also be used, which in CF can be translated into z-scores [5] instead of ratings, for example. These techniques have not been studied in this work, but are envisioned as future work. We also plan to extend our study to further IR models such as language models or probabilistic models.

Finally, although our results are consistent across two different datasets, we plan to test our framework on further publicly available datasets, such as Netflix or *Movielens 10M*. Besides, comparison with other state-of-the-art techniques should be performed, such as SVD, for a more detailed answer to **RQ2**.

Acknowledgments. This work was supported by the Spanish Ministry of Science and Innovation (TIN2008-06566-C04-02), and the Regional Government of Madrid (S2009TIC-1542).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE TKDE 17(6), 734–749 (2005)
2. Belkin, N.J., Croft, W.B.: Information filtering and information retrieval: two sides of the same coin? Commun. ACM 35(12), 29–38 (1992)
3. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: 14th Conference on UAI. pp. 43–52 (1998)
4. Cöster, R., Svensson, M.: Inverted file search algorithms for collaborative filtering. In: 25th ACM SIGIR conference. pp. 246–252. ACM (2002)
5. Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: 22nd ACM SIGIR conference. pp. 230–237. ACM (1999)
6. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, 1 edn. (2008)
7. Metzler, D., Zaragoza, H.: Semi-parametric and non-parametric term weighting for information retrieval. LNCS, vol. 5766, pp. 42–53. Springer (2009)
8. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
9. Wang, J., Robertson, S., de Vries, A., Reinders, M.: Probabilistic relevance ranking for collaborative filtering. Information Retrieval 11(6), 477–497 (2008)