TechLens+: A Hybrid Approach to Recommending Research Papers

Roberto Torres[1, 2], Mara Abel[1], John Riedl[2]

[1]Instituto de Informática - Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre, RS - Brazil

[2]GroupLens Research Project - Department of Computer Science and Engineering
University of Minnesota - MN 55455 – United States

{rtorres, marabel}@inf.ufrgs.br, riedl@cs.umn.edu

Abstract. *In this paper, we present and test hybrid recommender algorithms that combine Collaborative Filtering and Content-based Filtering for recommending research papers. The hybrid approaches combine the strengths of each algorithm to address their individual weaknesses. We evaluated our algorithms through both offline experiments on a database of 102,000 research papers, and an online experiment with 110 users. Our results show that users value recommendations of papers, that the hybrid algorithms can be successfully combined, that different algorithms are more suitable for recommending different kinds of papers, and that users with different levels of experience perceive recommendations differently. Our results lead us to a completely tailored research paper recommender system.*

1. Introduction

According to the United States' National Science Foundation, more than 530,000 research papers are published every year. In order to deal with this overwhelming number of published papers, many digital libraries and information retrieval systems have been built. Many of these systems respond to user queries, usually based on a set of keywords. Systems like CiteSeer and the ACM Digital Library store papers in a centralized repository, making them available through parametric searches. However, these solutions do not guide users to papers of their interests, and commonly users struggle to find relevant papers to read.

In the past few decades, researchers have turned to recommender systems to further help reduce information overload. These systems recommend items to users that might be valuable to them, according to their preferences. Our goal is to reduce this problem by using recommender systems on the task of finding relevant papers. Thus, we combine the two most known recommender systems' techniques, namely Collaborative Filtering and Content-Based Filtering to recommend research papers.

Collaborative Filtering (CF) is one of the most successful techniques used in recommender systems. It has been used to recommend Usenet news (Resnick et al. 1994), audio CDs (Shardanand;Maes 1995), and research papers (McNee et al. 2002), among others. CF works by recommending items to people based on what other similar people have liked. CF creates neighborhoods of "similar" users (neighbors) for each user in the system and recommends an item to one user if his neighbors have rated it highly. CF is "domain independent" in that it performs no analysis of the items in the domain. Rather, it relies on user opinions about the items to generate recommendations. Despite being a successful technique in many domains, CF has the first-rater and sparsity problem. The former is because and item cannot be recommend until one neighbor rates it and the latter is because one user is very likely to consume only a few percentage of the items in a domain, becoming hard to find agreements among users.

Content-Based Filtering (CBF) is also commonly used in recommender systems. Applied mostly in textual domains, such as news (Kamba et al. 1995), CBF recommends items to a user if these items are similar in content to items the user has liked in the past. CBF has many strengths, including the ability to generate recommendations over all items in the domain. However, CBF cannot successfully analyze items in domains like audio and video, it does not consider aspects like the writing style and it has the over-specialization problem, because it recommends only similar items to the ones the user rated in the past.

This research builds upon previous work by exploring how to combine the complementary advantages of CF and CBF, generating recommendations of research papers (McNee;Albert et al. 2002). We propose a set of new hybrid algorithms that combine a TF-IDF CBF algorithm, with a k-nearest-neighbor User-user CF algorithm. The full research is described in (Torres et al. 2004).

2. CF-CBF Hybrid Algorithms

Ten different algorithms were developed and tested; each algorithm is seeded by one paper and generates a list of papers as recommendations. Two algorithms were used as baseline comparison: Pure-CF and CBF-Separated. Pure-CF is the standard knn-User-user CF algorithm (Herlocker et al. 1999). It takes the citations of the selected ("active") paper as input and gives a list of recommended papers as output. CBF-separated search for similar papers based on TF-IDF similarity and recommends the most similar papers as follows: for every citation of the active paper one recommendation list is generated. In the end, all lists are merged and sorted using its similarity with the active paper. The most similar are recommended.

Each hybrid algorithm is composed of two engines: A CF algorithm and a CBF algorithm. They are combined in two ways: either they run in sequence or in parallel. In sequence, the output of the first engine (up to 20) is used as input to the second. For instance, the algorithm *CF-CBF Separated* is seeded by one paper and the CF engine recommends up to 20 papers. Thus, for every paper given by CF it is generated a recommendation list using the algorithm CBF-Separated. On the other hand, *CBF Combined-CF* runs CBF with a modification: instead of generating a list of recommendations for every citation, it merges all full text of the citations into one single text and uses this to seed the CBF engine (CBF-Combined). The recommendations of CBF are used as input to the CF engine.

In the parallel, the final recommendation list has items coming from both techniques. In this algorithm, it runs the two engines separately and combines their recommendations as follows: every recommendation that is present in both modules' result lists is added to the final list with a rank score. This score is the summation of the ranks of the recommendation in their original lists. The final recommendation list is sorted based on these scores. Therefore, a paper that was ranked 3rd from the CF module and 2nd from the CBF module would receive a score of 5. The lower the score, the closer to the top an item goes. The other recommendations that don't appear in both lists are alternatively added in the final list, coming either from the CF or the CBF recommendation list.

3. Experiments

To validate the algorithms two kinds of experiments were performed: one **offline** (without users) and one **online** (with users). These experiments have different goals: the offline tries to measure the ability of the algorithms to recommend papers. The online experiments assess users' satisfaction and perceptions about the recommendations.

In order to test our algorithms we created a dataset with papers extracted from CiteSeer. This dataset has 102.295 papers. All of them cite at least three other papers and all the citations are papers for which we have the full text. In average, every paper cites and is cited by seven other papers.

In the offline experiment, we randomly removed one citation from the active paper and then checked whether our algorithms could recommend that removed citation. This "leave one out" methodology has been frequently used in other recommender systems' offline experiments (Breese et al. 1998; McNee;Albert et al. 2002). We also divided the dataset into training and test datasets at a 90% to 10% ratio. Ten different training and testing dataset were created for 10-fold cross validation.
We define "hit-percentage" (HP) as a metric to measure the percentage of the time the recommender algorithm correctly recommended the removed citation. We also measured the rank where the removed citation was found. Figure 1 presents the results by algorithms and layers. For instance, Fusion had 28% of top-1 recommendations, which means that in 28% of the cases, the removed citation was the first to be recommended.

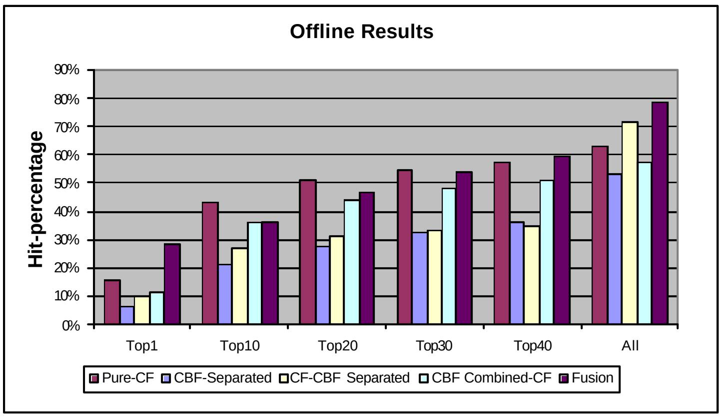

Figure 1. Offline Experiments' Results

We developed an online experimental system, called TechLens^{+}, consisting of a six-page Web-based experiment where users evaluated recommendations of papers by answering questions for each single recommendation and for the whole set of recommendations. Users ranged from masters and PhD students to professor and researchers and were invited to anonymously participate in the experiment through links in related websites, posts in universities' internal lists and interest lists, like the User Modeling Interest List and interest lists of the Brazilian Computer Society.
In order to gather user opinion about the recommendation received, the user was asked to answer questions regarding satisfaction, classification of the recommendations

(novel, authoritative, introductory, specialized, survey/over view), and the familiarity with them. During the 32-day experimental run, 110 subjects participated in the experiment. Users were randomly assigned to each algorithm. On average, subjects spent 20 minutes answering our questions. The results show that the user satisfaction for the individual recommendation was 46% and for the overall set of recommendations was 62%. Otherwise, the user dissatisfaction for the individual recommendation was of 21% and for the overall set of recommendations was 19%.

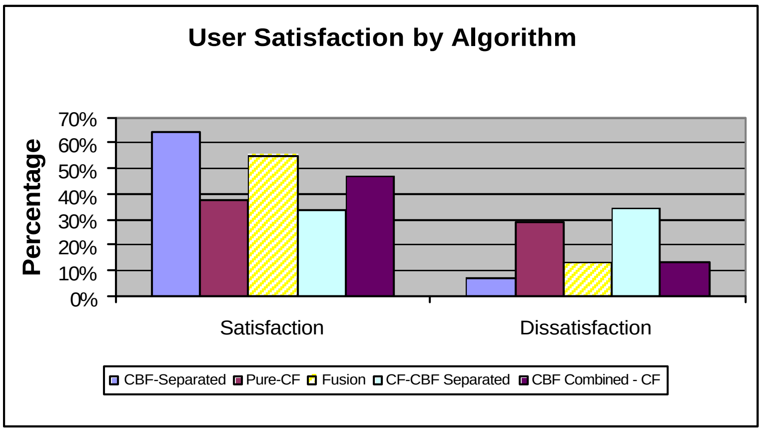

Figure 2. Users' Satisfaction by Algorithm

Recommendation satisfaction varied by user type with 75% of the masters students, 61% of the PhD students, 67% of the researchers, and only 52% of the professors saying they were satisfied with their recommendations.

5. Online results

Our results showed that students (masters and PhD) were happier with the recommendations than professionals (professors and researchers). Further, users were more satisfied with CBF-Separated recommendations. We have two hypotheses for this: first, because CBF-Separated tends to give recommendations on the author's specialty, they might be happier about it. Second, we ran the experiment only for 32 days, which might not be enough time to say which one is really the better. In the long run, Fusion might perform best. Figure 2 shows user's satisfaction by algorithm. Moreover, according to our statistical analysis [1] on the users' answers, table 1 summarizes the best algorithms to recommend each kind of paper.

6. Conclusions and Future Work

In this research we described, implemented and tested different techniques for combining content-based and collaborative filtering recommender algorithms for recommending research papers. We verified our results through both offline and online experiments. Although we tested in Computer Science research papers, the algorithms can be used in any scientific domain. Returning to our hypotheses, we found that the hybrid recommender algorithms can generate paper recommendations that users were very happy to receive, since 85% of the users said they received at least one good recommendation. Of particular note is the Fusion, which can promptly incorporate any evolution to each single technique.

---

[1] Unless noted otherwise, significance tests are at $p < 0.05$

Class of Papers	Best Algorithms	Worst Algorithms
Novel	Pure-CF and Fusion	CBF-Separated
Authoritative	Pure-CF and Fusion	CF-CBF Separated
Introductory	CBF-Separated and CF-CBF Separated	Pure-CF
Survey/Overview	CBF-Separated	Pure-CF

Table 1: Recommended Algorithms by Paper Class

One other advantage of building this hybrid system is the increase of coverage, which is the percentage of items for which the system can generate a recommendation. All of the hybrid algorithms had 100% coverage due to the CBF approach. Moreover, our online results showed that different algorithms should be used for recommending different kinds of papers. In addition, our results showed that users with different levels of experience perceive recommendations differently.

This leads us to a vision of a completely personalized or 'tailored' recommender system. Our results suggest that such a system can be tailored in two ways. First, it can tailor recommender algorithms for particular user tasks using Table 1 as a guide. Second, a system can be tailored based on the user's level of experience.

References

BREESE, J., et al. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. Conference on Uncertainty in Artificial Intelligence, 1998, Madison, Wisconsin. **Proceedings.** 1998.

HERLOCKER, J., et al. An Algorithmic Framework for Performing Collaborative Filtering. ACM SIGIR Conference on Research and Development in Information Retrieval, 1999, Berkeley, CA. **Proceedings.** 1999.

KAMBA, T., et al. The Krakatoa Chronicle: An Interactive Personalized Newspaper on the Web. International World Wide Web Conference, 4, 1995, Boston. **Proceedings.** O'Reilly & Associates, 1995.

MCNEE, S., et al. On the Recommending of Citations for Research Papers. Computer Supported Cooperative Work Conference, 2002, New Orleans, Louisiana - USA. **Proceedings.** ACM, 2002.

RESNICK, P., et al. GroupLens: An open architecture for collaborative filtering of Netnews. Computer Supported Collaborative Work Conference, 1994, Chapel Hill, North Carolina - USA. **Proceedings.** 1994.

SHARDANAND, U.; MAES, P. Social information Filtering: Algorithms for automating "word of mouth". Conference on Human Factors in Computing Systems, 1995, New York. **Proceedings.** 1995.

TORRES, R., et al. Enhancing Digital Libraries with TechLens+. Joint Conference on Digital Libraries (JCDL), 2004, Tucson, Arizona - USA. **Proceedings.** 2004.