Final Report
Independent Study
Fall 2009

Denis Parra, PhD Student

Professor Peter Brusilovsky, Advisor

Topic: Recommender Systems in Social Tagging Systems

Title: “Improving Collaborative Filtering in Social Tagging Systems for the Recommendation of Scientific Articles”

Improving Collaborative Filtering in Social Tagging Systems for the Recommendation of Scientific Articles

Denis Parra-Santander’ PhD Student
Professor Peter Brusilovsky, Advisor

1 Introduction

This report describes our study of different ways to improve existing collaborative filtering techniques in order to recommend scientific articles. Using data crawled from CiteUlike, a collaborative tagging service for academic purposes, we compared the classical user-based collaborative filtering algorithm as described by Schafer et al. [2], with two enhanced variations: 1) using a tag-based similarity calculation, to avoid depending on ratings to find the neighborhood of a user, and 2) incorporate the amount of raters in the final recommendation ranking to decrease the noise of items that have been rated by too few users.

We provide a discussion of our results, describing the dataset and highlighting our findings about applying collaborative filtering on folksonomies instead of the classic bipartite user-item network, and providing guidelines of our future research.

2 Datasets

CiteULike[1] is a social tagging system for storing, organizing and sharing research articles. CiteULike runs since 2004 and provides a daily dump of its dataset to the community. However, it is a restricted dataset. As a result, direct crawling of the site was necessary to gain access to data about users, articles, tags and timestamps of the posting action. Crawling is a time-consuming process. To allow an early assessment of our ideas, we divided the evaluation into two phases: phase 1, a user study based on small dataset, and phase 2, a large scale evaluation with larger dataset.

Phase 1 Dataset. For phase 1 study we chose ten active CiteULike users which had posted at least 50 articles each to be our center users. Four of the subjects are members of the Personalized Adaptive Web Systems (PAWS) lab at the School of Information Sciences at the University of Pittsburgh. Six additional subjects were selected from a list of active CiteULike users and invited to participate on our study. For each one of these *center users*, the crawler collected their posted articles (id, title, authors, post timestamp, and tags associated), the neighborhood of users who posted the same articles, and the neighborhood of users who shared the same tags. To avoid limiting the neighborhood while crawling, we stemmed the tags by using Krovetz algorithm, and we also added compound tags changing hyphens and underscores.

The details of the final dataset for the phase 1 study are described in the second column of Table 1. In total, we crawled 358 users and all their respective neighbors. For each of these neighbors we also crawled all their articles and tags. In Table 1, a *tagging incident* correspond to a triple {user, article, tag}

Table 1. Description of the datasets used.

Item	Phase 1 dataset	Phase 2 dataset
# users	358	5,849
# articles	186,122	574,907

[1] www.citeulike.org

# tags	51,903	139,993
#tagging incidents	902,711	2,337,571

Phase 2 Dataset. For our phase 2 study we used a dataset consisting of crawling CiteULike for 38 days during June and July of 2009. The raw data of this dataset is presented in the third column of Table 1. Following a common evaluation methodology of recommender systems, we filtered out the dataset to keep a p-core [9], with a p of 20 for users and a p of 2 for articles. This means that each retained user appears in at least 20 posts and that each article has been posted for at least 2 different users. We followed the cleaning procedure described and suggested in [10]. The characteristics of this filtered dataset are presented in Table 2:

Table 2. Characteristics of the filtered dataset for the phase 2 study

Item	# unique instances	Item	# unique instances
# users	784	avg # items per user	91
# items	26,599	avg # users per item	2.68
# tags	26,009	avg # tags per user	88.02
# posts	71,413	avg # users per tag	2.65
#tagging incidents	218,930	avg # tags per item	7.07

3. Algorithms

User-based collaborative filtering process consists of two steps. The first step is finding the *neighborhood* of the *center user*, i.e., a set of the most similar users. The second step is to rank the items to be recommended, and recommend the Top N items[2]. These items are taken from the set of items which the neighbors rated positively, and which the center user has not posted on her library. We implemented Classic Collaborative Filtering (CCF) as the baseline method and two enhancements: BM25-based similarity (BM25) as an alternative of the first step, and Neighbor-weighted Collaborative Filtering (NwCF) as an alternative of the second step.

Classic Collaborative Filtering (CCF). This approach is described in detail in [2]. In the classic CF model, the similarity between two users is calculated using the Pearson correlation over the ratings of their common items. The formula for the Pearson correlation, as stated in [2], is:

$$userSim(u,n) = \frac{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)(r_{ni} - \bar{r}_n)}{\sqrt{\sum_{i \subset CR_{u,n}} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \subset CR_{u,n}} (r_{ni} - \bar{r}_n)^2}} \qquad \mathbf{(1)}$$

In the formula, r stands for rating, u denotes the center user and n a neighbor. r_{xi} represents the rating given by the user x to the item i, and $\bar{r}_x$ is the average rating of the user x over all her items. $CR_{u,n}$ denotes the set of co-rated items between u and n, being i an element in that set. Next, we rank the articles of these users to recommend to the center user, using the formula of predicted rating for user u with average adjusts described in [2]

[2] That is why this method is called Top N recommendation.

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} userSim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} userSim(u,n)} \qquad \textbf{(2)}$$

BM25-based Similarity (BM25). BM25, also known as Okapi BM25, is a non-binary probabilistic model used in information retrieval [7]. It calculates the relevance that the documents of one collection have given a query. As we try to take advantage of the set of tags of each user, we made two analogies, comparing the tags of the center user with a query, and the set of tags of each neighbor with a document. Based on this, we use BM25 to calculate similarity and thus we obtain her neighborhood. Our proposed BM25-based similarity model is taken from the calculation of the Retrieval Status Value of a document (RSVd) of a collection given a query [7]:

$$RSV_d = \sum_{t \in q} IDF \cdot \frac{(k_1+1)tf_{td}}{k_1((1-b)+b \times (L_d / L_{ave}))+tf_{td}} \cdot \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}} \qquad \textbf{(3)}$$

In our model RSV_d represents the similarity score between the center user (the terms of the query q) and one neighbor (the terms of the document *d*). This similarity is calculated as a sum over every tag *t* posted by the center user. The neighbor *d* is represented as her set of tags with their respective frequencies. L_d is the document length, in our case is the sum of the frequencies of each tag of the neighbor *d*. L_{ave} is the average of the L_d of every neighbor. The term tf_{td} is the frequency of the tag t into the set of tags of the neighbor d. tf_{tq} represents the frequency of the tag t into the query, i.e., the set of tags of the center user. Finally, k_1, k_3 and *b* are parameters that we set to 1.2, 1.2 and 0.8 respectively, values slightly different from those suggested by default in [7]. After calculating this similarity measure, we choose the top N most similar neighbors, and then we calculate the ranking of the recommended articles using the formula (2) or (4).

Neighbor-weighted Collaborative Filtering (NwCF). This method enhances the ranking step by taking into account the number of raters. It is usueful to filter out potentially noisy items, which have been rated by only one or at most two users. In this way, we push up in the recommendation list the items rated by a larger number of neighbors. The new predicted rating is given by

$$pred'(u,i) = \log_{10}(1+nbr(i)) \cdot pred(u,i) \qquad \textbf{(4)}$$

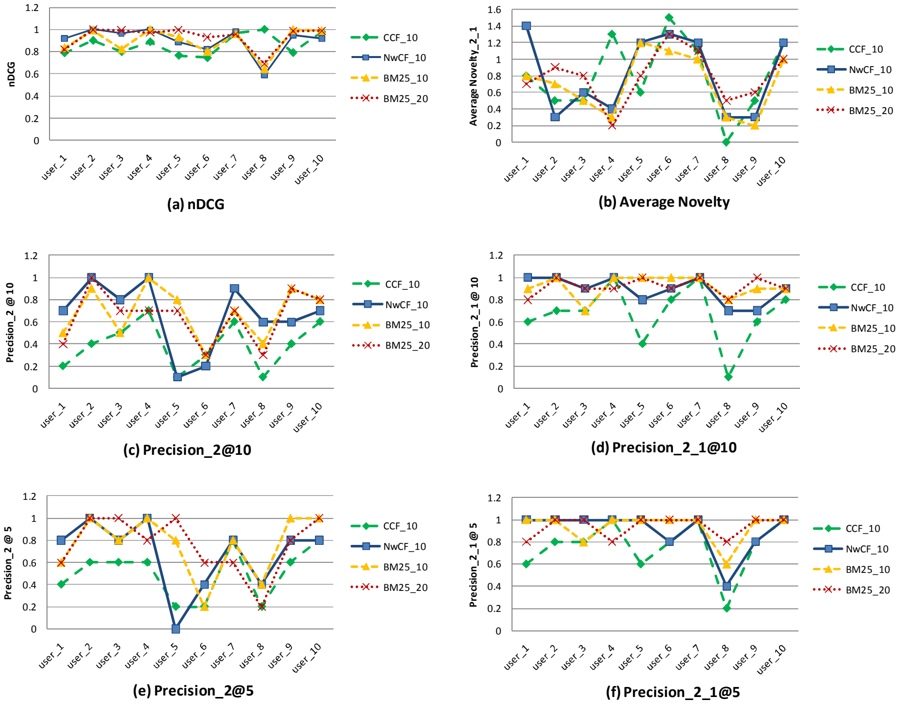
4. Experiments

To test our approaches we did two different experiments: a pilot user study (phase 1 study), and an objective evaluation by a 10-fold cross-validation experiment (phase 2 study).

Phase 1 study. We conducted this study in order to assess the feasibility of our approaches before committing to large-scale crawling. For each subject we generated 4 sets with 10 ranked articles each one. The first two lists were generated using the CCF and NwCF. The other two were generated by BM25+NwCF, using 10 and 20 neighbors for each center user correspondingly. This deign of the pilot study was driven by the goal to assess the cumulative effect of both innovations. To avoid bias in the evaluation [8], for each subject we combined the 4 sets of recommendations into one set, we changed the order of the articles randomly and we ask them to evaluate each article relevancy (relevant, somewhat relevant, and not relevant), and novelty (novel, somewhat novel, and not novel) using a 3-point scale.

To make the comparison more reliable we controlled the amount of information about an article used to make relevance judgment. For each article, we provided a URL to its CiteULike record, which provides basic bibliographic information and frequently an abstract for each article. We requested each subject to evaluate the articles based on that information or, if the abstract was not available, looking for the abstract in the paper source, but do not beyond that.

For each subject, we calculated normalized Discounted Cumulative Gain (nDCG) [7], Precision_2@5, Precision_2@10, Precision_2_1@5 and Precision_2_1@10 over the different initial four lists of recommendations. In Precision_2_1, we consider relevant those articles evaluated as Relevant and Somewhat Relevant. In Precision_2, we only consider relevant the articles evaluated as Relevant. Besides, we calculated the average novelty for each user on each method. To calculate the average novelty, we considered only items evaluated as relevant or somewhat relevant, disregarding novelty of not relevant items.



(a) nDCG

(b) Average Novelty

(c) Precision_2@10

(d) Precision_2_1@10

(e) Precision_2@5

(f) Precision_2_1@5

Figure 1. Metrics showing the results of each user on each method of the user study (a) nDCG, (b) Average Novelty, (c) Precision_2 @ 5, (d) Precision_2 @ 10, (e) Precision_2_1 @ 5, (f) Precision_2_1 @ 10.

The analysis of the data, in particular the results on Figure 1a), shows rather smooth results on different subjects and not so different results on the values of nDCG between different algorithms. Most importantly, we can see that CCF was the worst, while it is not so clear which one, BM25_10, BM25_20 or NwCF is better than others. This result suggests us that the ranking order is very close to the optimal one, having the most relevant articles at the top and the less ones at the bottom of the list of recommendations. In figure 1 b) is not possible to see any clear trend about which algorithm performs the best on novelty.

The results on Precision_2 and Precision_2_1 encourage us to continue our study. CCF performs noticeable worse than the rest, suggesting that including the amount of raters in the ranking formula and considering an alternative measure of similarity are feasible factors to the success of the recommendations.

Phase 2 study. This study was conducted to make a reliable assessment of our innovation. To assess both separate and cumulative impact of the suggested approaches we compared four conditions (CCF, NwCF, CCF+BM25 and NwCF+BM25), not just three as in phase 1. As stated in section 3, the CCF recommendation process has 2 steps: 1) calculation of user-similarity, and 2) raking the items to be recommended. While NwCF is an alternative for the second step, BM25 is an alternative for the first one, so they could be both used separately (retaining the classic CCF implementation of the other step) or combined, replacing both steps of CCF.

On this study we used the dataset described in 2.2, and performed the evaluation using an IR perspective, comparing MAP@10, a modified version of Mean Average Precision (MAP) [7] and User Coverage (UCov) after a 10-fold cross-validation evaluation.

As accuracy metric, we initially considered MAP, which would be calculated by averaging over the average (AP) precision of every user. However, a recommender system rarely displays the complete list of possible recommended items to the center user, which can be thousands. It typically displays the top N items, so we decided to calculate the AP for each user and cutting at retrieval point 10 (AP@10). The formula can be expressed as:

$$AP@10 = \frac{\sum_{\mathrm{r}=1}^{10}\left(\mathrm{P}(\mathrm{r}) \times rel(r)\right)}{number \text{ of } relevant \text{ items @ retrieval point 10}} \qquad (\mathbf{5})$$

In the formula *r* is the rank, *rel()* a binary function on the relevance of a given rank, and *P()* precision at a given cut-off rank. MAP@10 corresponds to averaging AP@10 over a set of users. On the other side, UCov quantifies the percentage of users for whom the system could generate recommendations.

The evaluation is done by a 10-fold cross-validation process, which is visualized in figure 2. First, the dataset is divided into 10 folds, where 10% of the users are randomly assigned to each fold (A). The process follows by selecting one fold as the testing set and the remaining 9 folds as the training set (B).

The training set is then used to optimize the main parameter in a collaborative filtering algorithm: *k*, the size of the neighborhood. To calculate the optimal size of the neighborhood, for each user in the training set we withhold 10 articles to be predicted, and we measure the quality of our prediction by calculating its Avg@10 (C). Then, we average over the AP@10 of the whole set of users in the training set obtaining the MAP@10 (D). The neighborhood size with the highest MAP@10 in the training step is used to calculate the MAP@10 of the fold withhold for the testing step (E). We repeat the steps (B), (C), (D) and (E) for every fold, hence, 10 times. Finally, we average over the MAP@10 of each fold to calculate the final MAP@10.

To compare the accuracy of the baseline method CCF against NwCF, BM25, and their combination, we followed the process described in the two previous paragraphs, testing statistical significance of our results with the Wilcoxon paired test of each method compared to the baseline method, CCF. We didn't use a paired t-test since the distribution of the average precision over the users is not normal.

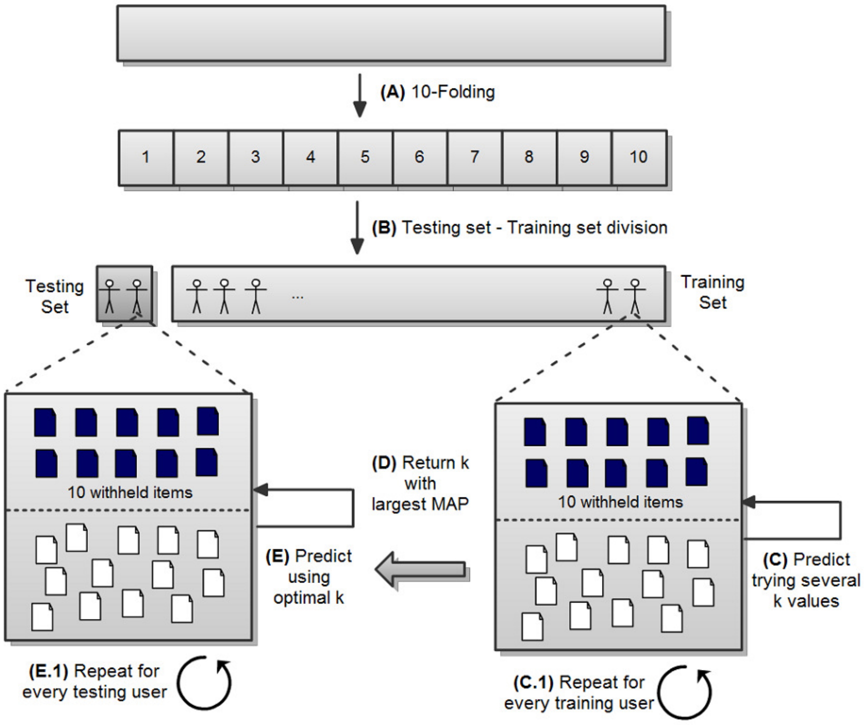

(A) 10-Folding
1
2
3
4
5
6
7
8
9
10
(B) Testing set - Training set division
Testing Set
...
Training Set
10 withheld items
(D) Return k with largest MAP
10 withheld items
(E) Predict using optimal k
(C) Predict trying several k values
(E.1) Repeat for every testing user
(C.1) Repeat for every training user

Figure 2. Description of the 10-fold cross-validation process.

On this phase, we tested the condition missed in the pilot study: a combination of BM25 with CCF. As we expected, BM25+NwCF is still the leader, in table 3 a MAP@10 of 0.1942 compared to a MAP@10 of 0.0876, significant with p<0.001. Now, comparing simple CCF with BM25+CCF, one may argue that the actual improvement is done by NwCF rather than BM25. However, the importance of BM25 stems not from MAP@10 results, but from its coverage (99.23% compared to 81.12% over 784 users), i.e., more users can receive recommendations. It makes BM25 a sound alternative for Pearson correlation (especially taking into account that many social bookmarking systems have no ratings). We extend the explanation of this result in the discussion section.

Table 3. Results of the phase 2 study

	CCF	NwCF	BM25+CCF	BM25+NwCF
MAP@10	0.12875	0.1432*	0.0876**	0.1942***
N	20	22	21	29
Ucov	81.12%	81.12%	99.23%	99.23%

Significance over the baseline: *p < 0.236, ** p < 0.033, *** p < 0.001

In the case of NwCF, the results lead us to the same conclusions that our pilot study, as seen in table 3 and 4. They show an improvement of NwCF (MAP@10 0.1432) over CCF (MAP@10 0.12875), with p<0.236. The statistics of table 2 can give us an explanation. With an average of just 2.68 users per item, it is difficult to assure that the ratings given to one item reflect an objective evaluation. By improving the recommendation ranking to the items rated by a larger amount of users, we decrease the noise of that evaluation. A simple and clear example is the online store Amazon.com: we tend to trust more in the aggregated rating of one item when it has been evaluated by more people.

5. Related Work

A few pioneer projects explored different ways to integrate social links or social tags. In [3], the authors incorporate social tags and also the concept of web of trust for the issue of quality assessment into a collaborative recommendation approach. The study in [4] investigates the effect of incorporating tags to different CF algorithms, testing their algorithms on last.fm, a musical social tagging system, obtaining promising results. The approach presented in [5] compared a pure content-based with a tag-enhanced recommender, showing an improvement in predicted accuracy in the context of cultural heritage personalization. The study presented in [6] describes the use of CiteULike for recommending articles to users. They compared three different collaborative filtering algorithms, two item-based and one user-based, and they found that the latter performed the best. They evaluated their algorithms using accuracy metrics as MAP, MMR and Precision@10, with low accuracy levels, in the range 0.1-0.3.

In [8] McNee et al. developed three algorithms to recommend articles to users, and they assessed them with a detailed survey on real users. In some algorithms, the subjects provided strong negative results, and the authors concluded that when evaluating a recommender system "the evaluation must be done with real users, as current accuracy metrics cannot detect these problems". Based on this study we decided to ask the subjects to evaluate the novelty in addition to the relevance. Five of our ten subjects commented at the end of the survey that they found very interesting articles in their recommendation list.

6. Discussion

The results of our user study, as well as the experience of other teams, show that one must approach the problem of recommendation in social tagging systems with an open mind. Pragmatic implementation of traditional approaches may deliver relatively poor results in this new context. Our work shows that both steps of a traditional collaborative filtering approach behave sub-optimally in CiteULike. First, the use of Pearson correlation to form user neighborhood delivers poor results. While CiteULike with its 5-star-plus-one categorization of bookmarked papers appears, at first sight, to be a good case for using Pearson formula, we found that in a bookmarking context this rating is not reliable. We started our user study using Pearson over five star-based rating, but we were puzzled with low quality of recommendations in the pilot study. To address it, we moved from 6-point to 3-point rating. Since many users post articles without taking care of the ratings (by default it is 2 stars), and their evaluation criteria can vary, we decided to treat default 2-star rating as considerable interest (1 point), explicit change to one star as low interest (0 points), and explicit change to 3-5 stars as high interest (2 points). Afterwards, the results showed a significant improvement. This highlights the importance of the rating scale used in recommender algorithms for social bookmarking systems, where the meaning of "stars" could be different, since tags, not stars are the primary product of bookmarking.

Yet, we believe that even the reduced-scale rating is not reliable enough to use Pearson correlation due to nature of bookmarking systems. While traditional recommender systems use a fair mixture of positive and negative ratings, a presence of a bookmark is mostly a positive sign. In this context, any additional star ratings as used in CiteULike represent different shades of positive and become less reliable. While some users may do their best by distinguishing "I want to read it" and "I really want to read it", a good fraction simply gives up and become single-value raters. In our case, in the unfiltered dataset of the phase 2, 21% of the users had rated all their articles with 2 stars (the default rating), and 34% have used the same rating (either with 1, 2, 3, 4, or 5 stars) over all their articles. In this context, Pearson correlation becomes too noisy. In the case of the phase 2 study, the filtered dataset presented a 10% of users as single-value raters. A small fix is to move from Pearson to some different options as those explored in [10] and [11], such as Jaccard, tf-idf or cosine similarity. We moved to a more radical option, switching from an item-based to a tag-based approach using a state-of-the art information retrieval model to calculate similarity between users. In our case it paid off: BM25 based similarity, combined with NwCF, performed better than CCF in both phases of the study.

BM25 has shown its potential as an alternative to Pearson correlation as a similarity measure, which can be especially useful in social bookmarking systems since they do not commonly provide ratings. Though showing worse results than CCF when is not combined with NwCF, its user coverage percentage highlights the importance of this method. Furthermore, when combined with NwCF, it provides better results than Pearson combined with NwCF. A natural question is: How can be explained the large difference in MAP between BM25+CCF and BM25+NwCF? We think that, as shown by the user coverage, BM25 is able to produce a larger but also "noisier" neighborhood than CCF. It allows finding more truly relevant users, while also mixing them with superficially relevant users. This additional noise is however, reduced by NwCF. Consequently, the combination of BM25+NwCF has the ability to bring more neighbors and also to re-enforce the signal of the most relevant neighbors. In comparison, CCF, which relies on item co-tagging, carries less noise, so its improvement with NwCF is not as large as combining BM25 with NwCF.

Yet, there is a trade-off on using BM25 between its accuracy and its scalability. Despite the good results of BM25+NwCF, BM25 by itself is more computationally-expensive than CCF. BM25 can be sped up by storing users and tags by a batch process using an indexer such as Lucene or Lemur, so at the moment of the recommendation the information of users is quickly retrieved. However, this indexing process must be executed off-line: running it each time a new user is added or a new article is posted would be too computationally expensive, at the expense of lacking the most updated information when creating the recommendations. A very active user or a popular article can produce similar scalability problems on CCF and NwCF. On this line of research, we have already started to work on a family of Spreading Activation Methods, which are less computationally expensive and more easily scalable for large datasets, though more difficult to tune, as we show in [12].

Our experience with NwCF demonstrates that the inclusion of the amount of raters in the ranking formula is an important contribution. In the user study, the analysis of the result shows that both nDCG and precision metrics have better results for NwCF than for CCF. This result hints that the number of raters is a part of the "social knowledge", which can increase the quality of outcome that CCF ignores. NwCF helps to reduce the noise of items rated by too few users, so it can be considered as an important tool given the sparsity of the dataset: much more items than users, as shown in the tables 1 and 2. This can be also seen as an option to alleviate the cold-start problem, common to new users and items in recommender systems. The new users still have not added enough articles to their library, and new items have not been shared by enough people to be suggested by the collaborative filtering algorithm. However, we cannot claim NwCF as an ultimate solution, since we still need users sharing at least one item and items posted by at least two users to provide to have a chance of recommendation. One option for those cases is using a content-based approach, or a hybrid approach that combines a collaborative filtering solution with a content-based one, as described in [10] and [13], such as computing cosine similarity or similarity based on available metadata.

7. Conclusions and Future Work

We explored two variations of classic user-based collaborative filtering algorithms in the context of a social tagging system for scientific articles, CiteULike. We can summarize the results of our study in two observations. First, incorporating the amount of raters in the ranking algorithm can help to decrease the uncertainty produced by items with too few ratings. Second, a tag-based approach to obtain the user neighborhood in social tagging systems can be a suitable alternative to CCF. In addition, this method ensures better user coverage, since is not affected by the problems arisen by single-value raters when using Pearson correlation to calculate user-similarity. Another important point that we highlight, though not arising directly from the results but rather from our work, is that classic rating-based collaborative filtering algorithms implemented on social tagging systems must carefully consider the rating scale used, to avoid noise in the recommendation lists.

Some open issues regarding this research remain open. One of them is how to beat the cold-start problem when providing recommendations in social tagging systems. Using a content-based or a hybrid-based approach can help to mitigate this problem. Scalability is also an area of concern and we are already working in developing novel spreading activation algorithms to calculate local ranking of users and items around her network.

9. References

[1] Pazzani, M. and Billsus, D.: Content-Based Recommendation Systems. The Adaptive Web. Springer, Berlin, 325-341 (2007)

[2] Schafer, J., Frankowski, D., Herlocker, J. and Sen, S.: Collaborative Filtering Recommender Systems. The Adaptive Web, Springer, Berlin, 291-324 (2007)

[3] Massa, P. and Avesani, P.: Trust-Aware Collaborative Filtering for Recommender Systems. In Proceedings of OTM Confederated International Conferences, CoopIS, DOA, and ODBASE. 492-508 (2004)

[4] Tso-Sutter, K. H., Marinho, L. B., and Schmidt-Thieme, L.: Tag-aware recommender systems by fusion of collaborative filtering algorithms. In Proceedings of the 2008 ACM Symposium on Applied Computing (Fortaleza, Brazil, March 16 - 20, 2008). SAC '08. 1995-1999 (2008)

[5] de Gemmis, M., Lops, P., Semeraro, G., and Basile, P.: Integrating tags in a semantic content-based recommender. In Proceedings of the 2008 ACM Conference on Recommender Systems (Lausanne, Switzerland, October 23 - 25, 2008). 163-170 (2008)

[6] Bogers, T. and van den Bosch, A.: Recommending scientific articles using citeulike. In Proceedings of the 2008 ACM Conference on Recommender Systems (Lausanne, Switzerland, October 23 - 25, 2008). 287-290. (2008)

[7] Manning, C., Raghavan, P. and Schutze, H. Introduction to Information Retrieval. Cambridge University Press (2008)

[8] McNee, S. M., Kapoor, N., and Konstan, J. A.: Don't look stupid: avoiding pitfalls when recommending research papers. In Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work (Banff, Alberta, Canada, November 04 - 08, 2006). 171-180.

[9] Batagelj, V. and Zaveršnik, M. 2002. Generalized cores. Arxiv preprint cs/0202039 (2006)

[10] Bogers, T. and van den Bosch A.: Collaborative and Content-based Filtering for Item Recommendation on Social Bookmarking Websites. Proceedings of the Workshop on Recommender Systems and the Social Web, collocated with the 3rd ACM Conference on Recommender Systems, RecSys'09 (2009)

[11] Markines, B., Cattuto, C., Menczer, F., Benz, D., Hotho, A., Stumme, G.: Evaluating similarity measures for emergent semantics of social tagging. In: WWW '09: Proceedings of the 18th international conference on World Wide Web. pp. 641-650 (2009)

[12] Troussov, A., Parra, D. Brusilovsky, P. 2009. Spreading Activation Approach to Tag-aware Recommenders: Modeling Similarity on Multidimensional Networks. Proceedings of the Workshop on Recommender Systems and the Social Web, collocated with the 3rd ACM Conference on Recommender Systems, RecSys'09 (2009)

[13] Gemmell, J., Schimoler, Th., Ramezani, M., Christiansen, L., Mobasher, B.: Improving Folkrank with Item-Based Collaborative Filtering. Proceedings of the Workshop on Recommender Systems and the Social Web, collocated with the 3rd ACM Conference on Recommender Systems, RecSys'09 (2009)