Can't See the Forest for the Trees? A Citation Recommendation System

Cornelia Caragea[1], Adrian Silvescu[2], Prasenjit Mitra[3], C. Lee Giles[3]
[1]Department of Computer Science and Engineering, University of North Texas
[2]Naviance, Inc., Washington DC
[3]School of Information Sciences and Technology, The Pennsylvania State University
ccaragea@unt.edu, silvescu@gmail.com, pmitra@ist.psu.edu, giles@ist.psu.edu

ABSTRACT

Scientists continue to find challenges in the ever increasing amount of information that has been produced on a world wide scale, during the last decades. When writing a paper, an author searches for the most relevant citations that started or were the foundation of a particular topic, which would very likely explain the thinking or algorithms that are employed. The search is usually done using specific keywords submitted to literature search engines such as Google Scholar and CiteSeer. However, finding relevant citations is distinctive from producing articles that are only topically similar to an author's proposal. In this paper, we address the problem of citation recommendation using a singular value decomposition approach. The models are trained and evaluated on the Citeseer digital library. The results of our experiments show that the proposed approach achieves significant success when compared with collaborative filtering methods on the citation recommendation task.

Categories and Subject Descriptors

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

Keywords

citation recommendation; information filtering; collaborative filtering; singular value decomposition.

1. INTRODUCTION

As science advances, scientists around the world continue to produce a large number of research articles. These articles provide the technological basis for worldwide collection, sharing, and dissemination of scientific discoveries. Unfortunately, our ability to manually process and filter this huge amount of information lags far behind the number of research articles available today.

Research ideas are generally developed based on high quality citations. The search for such citations is usually done using specific *keywords* submitted to literature search engines such as Google Scholar [5] and CiteSeer [4]. However, text-based search engines return poor results when there is vocabulary mismatch between a query and the relevant documents. Moreover, finding relevant citations is distinctive from retrieving articles that are only topically similar to an author's proposal. For example, Teufel et al. [16] showed that citations can be of various types, and provided an annotation scheme for the citation function that consists of twelve different categories. Among these categories, some citations are topically similar, others are used as survey articles to provide background information to the reader, while yet others contain tools/algorithms/data that are adapted or modified in the new proposal [16].

What is a good strategy to uncover both topically-related and, at the same time, distant, but highly-relevant citations for a particular query, while filtering out irrelevant information, given today's very large collections of published articles? McNee et al. [10] studied the applicability of collaborative filtering (CF) to recommend citations for papers. However, CF algorithms have several limitations such as data sparsity and scalability [12]. In the citation recommendation task, the underlying citation graph tends to be noisy and sparse (potentially due to errors in citing, missing citations, or space limitation imposed by submission guidelines).

Against this background, in this work, we address the problem of citation recommendation using singular value decomposition (SVD) [3] on the adjacency matrix associated with the citation graph to construct a latent "semantic" space, where citing and cited papers, that are highly correlated, are placed closed to each other. The idea behind SVD is to project the original high-dimensional data into a lower-dimensional space, in which patterns in the data can be more easily identified. We exploit information available in CiteSeer to train and evaluate our models. The assumption is that, when writing a paper, an author has some background knowledge about the topic he writes about and that an initial set of citations (i.e., a "basket" of citations) is provided as input to the system. The system retrieves other relevant works that the author might have missed (works that *should* be cited or the author *should be aware of*).

Contributions. We present an application of SVD to build a reliable citation recommendation system and to address the limitations of memory-based CF algorithms. The results of our experiments show that the SVD-like recommender systems achieve significant success when compared with CF approaches on a subset of the CiteSeer citation graph, a newly constructed data set, made available online from the first author homepage.

JCDL '13, USA

2. RELATED WORK

A variety of approaches to citation recommendation (detailed below) have been recently proposed in the literature.

Collaborative filtering. Using the adjacency matrix associated with a citation network, McNee et al. [10] tested the ability of CF to recommend appropriate additional citations for a target paper, given an initial set of citations (i.e., a "basket" of citations). The analogy with the conventional CF is that citing papers correspond to users and citations correspond to items. Tested in both online and offline settings, CF resulted in high-quality recommendation lists [10].

Citation ranking using content and graph-based information. Strohman et al. [14] presented a graph-based approach in order to generate a references list for a query paper (i.e., a paper with no citation information). The assumption is that the query paper has several pages in length that are written on a specific topic. The approach exploits both the textual similarity and the citation information between the papers in a collection. Bethard and Jurafsky [1] enriched the set of features used in [14] in the same framework, with the exception that the query paper consists only of the abstract.

Topic models-based link prediction. Nallapati et al. [11] extended topic models [2] to discover clusters of topical words as well as clusters of "topical citations", while exploiting the information flow from the citing to the cited documents.

Context-aware citation recommendation. Tang and Zhang [15] considered a topic-based approach to context-aware citation recommendation and proposed to match citation contexts with recommended papers, using Restricted Boltzmann Machines [13]. The context-aware approach to citation recommendation is defined as follows: given a query paper, for each citation placeholder, the task is to recommend a set of citations based on the context of the placeholder, also known as the *citation context* (i.e., a window of n words around the placeholder). Rather than generating a global references list for a query paper, local references lists are generated for each placeholder, based on the keywords in the citation contexts.

He et al. [6] proposed non-parametric probabilistic models to citation recommendation for placeholders in query manuscripts, which measure the context-based relevance between a citation context and a candidate citation for ranking a candidate set. Kataria et al. [8] extended the approach of Nallapati et al. [11] to jointly model content and citations by explicitly incorporating citation context information into the model. Lu et al. [9] and Huang et al. [7] proposed to use translation models to address the differences in vocabularies between the content of papers and the citation contexts, and show improvement in performance over the context-aware relevance model [6].

In contrast to the approaches reviewed above, we address citation recommendation using SVD. Rather than providing as input to the recommender system several pages of text, with no citation information, our system requires the user to input an initial set of citations (i.e., some background information about a research area). In formulation, our work is most similar to the work by McNee et al. [10]. For this reason, we compare SVD with collaborative filtering [10] only.

3. RECOMMENDER ALGORITHMS

Before reviewing the recommender algorithms compared in this study, we introduce some notations used in the paper.

Notations: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a directed citation graph, where the vertices p_i in $\mathcal{V}$, $i = 1, \cdots, |\mathcal{V}|$ represent papers, and the edges $[p_i \rightarrow p_j]$ in $\mathcal{E}$ represent the citations between papers. Let $C = \{c_{ij}\}$ denote the adjacency matrix associated with $\mathcal{G}$, such that $c_{ij} = 1$ if there is an edge from p_i to p_j, and $c_{ij} = 0$, otherwise. Note that the matrix C, also referred to as the *link matrix*, is asymmetric.

Similar to McNee et al. [10], we distinguish between *citing* and *cited* papers. Specifically, we define a *citing paper* (denoted by p) as a paper for which we have access to its content and the reference list, and a *cited paper* (or a *citation*, denoted by q) as a paper that occurs in the references list of at least one citing paper in the corpus, and for which we have access to its content (title and/or abstract), but may or may not have access to its references list. Because standard CF algorithms require the transformation of a dataset into a matrix of ratings, i.e., the columns of the matrix represent "items", the rows represent "users", and the entries in the matrix represent users' ratings of particular items, we adopted the matrix representation introduced by McNee et al. [10]. That is, the "users" from standard CF are replaced by *citing papers* (i.e., the matrix rows), and the "items" are replaced by *cited papers* (i.e., the matrix columns). Each citing paper would then "vote" for (or rate) the cited papers in its references list, and in contrast to standard CF, citing papers do not add more "votes" over time, after a citing paper is entered into the system. We denote by $\mathcal{P}$ the set of citing papers, and by $\mathcal{Q}$ the set of cited papers (or citations).

3.1 User-Based Collaborative Filtering

The user-based collaborative filtering algorithm first compares citing papers in $\mathcal{P}$ (rows) to determine a neighborhood $\mathcal{N}$ of the most similar n papers to the target paper (i.e., the size of $\mathcal{N}$ is n). The algorithm then computes a score for each citation in $\mathcal{Q}$ by counting the number of occurrences of the citation in the neighborhood $\mathcal{N}$, with each occurrence weighted by the similarity of the neighbor to the target paper. Finally, the algorithm recommends the top N citations with the highest scores. That is, for the j^{th} citation in $\mathcal{Q}$:

$$score_j = \sum_{i \in \mathcal{N}} c_{ij} \cdot w_i, \tag{1}$$

where w_i represents the cosine similarity of neighbor i with the target paper, and c_{ij} is the "vote" of neighbor i on the citation j. We call this method user-based simple-weighted-sum recommendation, denoted by CF User (SWS). We also experimented with a naïve user-based approach where the summation in Eq. 1 is not weighted by the neighbors similarities. That is, for the j^{th} citation in $\mathcal{Q}$:

$$score_j = \sum_{i \in \mathcal{N}} c_{ij}. \tag{2}$$

We call this method user-based most-frequent-item recommendation, denoted by CF User (F).

3.2 Item-Based Collaborative Filtering

The item-based collaborative filtering compares citations in $\mathcal{Q}$ (columns) to determine a neighborhood $\mathcal{N}$ of the most similar citations to each *known* citation (i.e., each citation in the "basket") of a target citing paper. Specifically, for a target citing paper p, a candidate set of citations is first identified by taking the union of the n most similar papers (denoted by $\mathcal{N}(j)$) for each known citation j of p, and excluding any of the known citations of p (i.e., $\mathcal{N} = \cup_j \mathcal{N}(j)$).

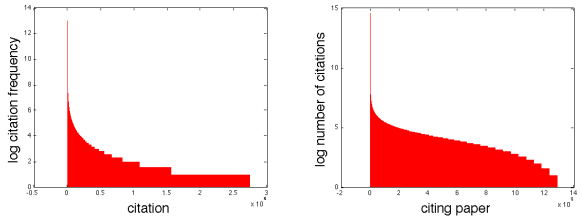

Figure 1: The CiteSeer citation graph information: (a) Frequency of citations, shown on a logarithmic scale; (b) Number of citations per citing paper.

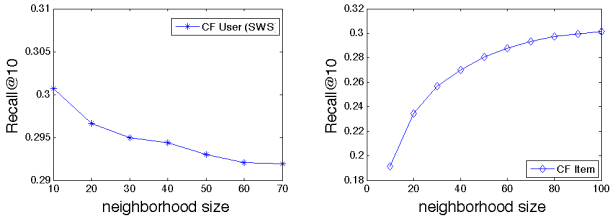

Figure 2: Neighborhood size estimation for: (a) CF User (SWS) and (b) CF Item.

For each citation c in the candidate set, the algorithms computes its similarity to the citations j in p, as the sum of similarities between all citations $j \in p$ and c, given only the n most similar papers of j. The algorithm recommends the top N citations from the candidate set with the highest scores. That is, for the c citation in the candidate set:

$$score_c = \sum_{j \in p} w_{jc}, \tag{3}$$

where $w_{jc} = 0$ if $c \notin \mathcal{N}(j)$ and w_{jc} is the cosine similarity between j and c, otherwise.

3.3 Regularized Singular Value Decomposition

SVD is a popular technique for identifying latent semantic factors, where association patterns in the data can be more easily identified, compared with the original space [3]. In our setting, using SVD, both citing and cited papers are mapped into a joint k-dimensional latent factor space, and the citing-cited correlations are modeled as inner products in this space. That is, each paper is represented as a vector in $\mathbf{R}^k$. Let $\mathbf{q}_i \in \mathbf{R}^k$ and $\mathbf{p}_u \in \mathbf{R}^k$ denote the vectors associated with the cited paper q_i and the citing paper p_u, respectively. The inner product $\mathbf{q}_i^T \mathbf{p}_u$ reflects the correlation between p_u and q_i, and approximates the "vote" of p_u on q_i, i.e.,

$$\hat{c_{ui}} = \mathbf{q}_i^T \mathbf{p}_u. \tag{4}$$

To avoid overfitting, we adopted a regularized formulation of SVD [17]. Regularized SVD learns the factor vectors by minimizing the regularized squared error on the training set (i.e., the link matrix C, or the matrix of "votes"):

$$\min \sum_{(u,i)} (c_{ui} - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda(||\mathbf{q}_i||^2 + ||\mathbf{p}_u||^2), \tag{5}$$

where (u, i) denote pairs of citing-cited papers with non-zero entries in C. In experiments, we used *stochastic gradient descent* to minimize Eq. 5. The algorithm iterates through all "votes" in the training, predicts c_{ui}, and computes the error between the actual and predicted "votes". That is,

$$err_{ui} = c_{ui} - \mathbf{q}_i^T \mathbf{p}_u. \tag{6}$$

The parameters are then updated, using the following updating rules:

$$\mathbf{q}_i \leftarrow \mathbf{q}_i + \alpha \cdot (err_{ui} \cdot \mathbf{p}_u - \lambda \cdot \mathbf{q}_i) \tag{7}$$

$$\mathbf{p}_u \leftarrow \mathbf{p}_u + \alpha \cdot (err_{ui} \cdot \mathbf{q}_i - \lambda \cdot \mathbf{p}_u) \tag{8}$$

We use regularized SVD to get predictions on individual "votes" and return top N recommendations for citing papers.

4. EXPERIMENTS AND RESULTS

Here, we describe our compiled CiteSeer data set [4] used in our experiments, and present the results of the comparison of SVD with the collaborative filtering approaches.

4.1 Dataset

The citation recommendation data set used in our experiments is compiled from the CiteSeer citation graph and the metadata available for each paper indexed in CiteSeer [4], as of December 2011. As already mentioned, we define a *citing paper* as a paper for which we have access to its content and the reference list, and a *cited paper* or a *citation* as a paper that occurs in the reference list of at least one citing paper in the corpus, and for which we have access to its content, but may or may not have access to its reference list. In the CiteSeer citegraph, there are $1,345,249$ unique citing papers and $9,150,279$ unique citations. The total number of links in the graph, i.e., [citing paper $\rightarrow$ citation], is $25,526,384$.

Figure 1(a) shows the frequency of citations in CiteSeer (on a logarithmic scale). As can be seen, the citations in CiteSeer typically follow a Zipf distribution, i.e., only a few citations are cited by very many citing papers, whereas the majority of them are cited rarely. Figure 1(b) shows the number of citations per citing paper, i.e., the size of the reference list. As shown, very few citing papers have a large number of citations, whereas for most of the citing papers the number of citations ranges between 8 and 32. From the citation graph and the available metadata, we constructed a smaller data set as follows: we filtered out papers that do not have title and abstract, as well as papers that are cited by other papers in the corpus less than 10 times and more than 100 times. In addition, we filtered out papers that cite less than 15 or more than 50 other papers. In the resulting citegraph, there are $81,508$ unique cited papers, $16,394$ unique citing papers, and $341,191$ links.

4.2 Experimental Design

Our experiments are designed to explore the following question: How does SVD compare with CF on the citation recommendation task for the returned top N citation recommendation lists?

To answer this question, we split the data set into training and test sets by randomly selecting one non-zero entry from each citing paper, to be part of the test set, whereas the remaining non-zero entries are considered part of the training set (i.e., the "basket items" for each citing paper). Thus, we sampled $16,394$ non-zero entries from the $s = 16,394$ citing papers and used them to test the models. Furthermore, in a similar manner, we sampled $16,394$ non-zero entries from

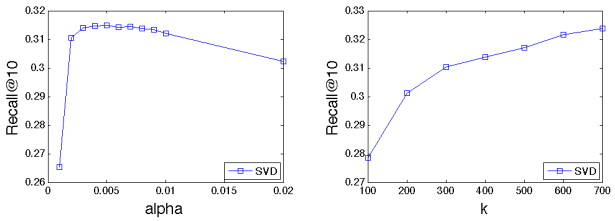

Figure 3: The estimation of: (a) the learning parameter α, and (b) the dimension of the latent space k.

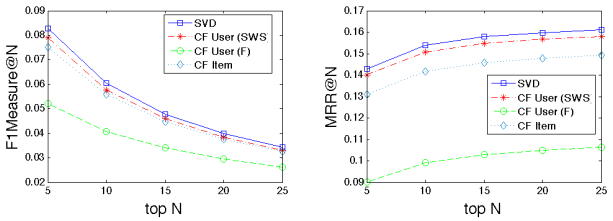

Figure 4: Comparison of SVD with CF, in terms of F1 Measure, and MRR, for various values of N.

the training set to be part of a validation set, which was used to estimate model parameters (detailed below).

In our experiments, all four algorithms return a list of top N recommendations for each citing paper. If a hidden citation (in the test set) is part of the top N recommendation list returned by an algorithm, the algorithm was considered accurate for the citing paper. We repeated each experiment 5 times to ensure the results are not sensitive to a particular train-test split. The results are averaged across the five runs.

Performance Measures. We used Recall, F1 Measure and the Mean Reciprocal Rank (MRR). The higher these measures are, the more accurate the results returned.

Parameter Tuning. To select a set of "good" parameters, i.e., a set of parameters that result in high-accuracy models, we trained the models on the training set (from which we removed the entries in the validation set), and selected the parameters on the validation set, as follows: we fixed N to 10 (i.e., the size of the recommendation list made by any algorithm). For user- and item-based CF, we chose the neighborhood size n that resulted in the highest recall on the validation set (see Figure 2). The selected values for n are: $n = 10$ for both CF User (SWS) and (F), and $n = 100$ for CF Item models (data not shown for CF User (F)).

For SVD, we fixed $N = 10$, and chose the learning rate α, the regularization factor λ, and the dimension of the latent space k that resulted in the highest recall on the validation set (see Figure 3). These values are: $\alpha = 0.005$, $\lambda = 0.001$, $k = 600$ (data not shown for the estimation of λ).

4.3 Results

Figure 4 show the results of the comparison of SVD with CF, i.e., user-based simple-weighted-sum, CF User (SWS), user-based most-frequent-item, CF User (F), and item-based, CF Item, in terms of F1 Measure, and MRR, for various values of N of the size of the top N recommendation lists, with N ranging from 5 to 25 in steps of 5. As can be seen in the figure, SVD outperforms the CF models in terms of both performance measure reported, for all values of N. This suggests that the citation graph data contain patterns of association of citations that SVD is able to find in the latent low-dimensional factor space.

Furthermore, the fact that SVD outperforms CF in terms of MRR, for all values of N, suggests that the original citations (in the references list of a citing paper) are higher ranked in the top N recommendation lists returned by SVD, compared to the lists returned by CF.

Again, as can be seen in Figure 4, the performance of CF User (F), is much worse than that of CF User (SWS) (as expected). On the other hand, the performance of item-based, CF Item, is similar to that of CF User (SWS), in terms of F1 Measure for all values of N, but is worse than that of CF User (SWS) in terms of MRR, suggesting that the original citations are ranked higher in the top N recommendation lists by CF User (SWS), compared with CF Item.

5. CONCLUSION

In this paper, we studied the applicability of SVD to citation recommendation and found that SVD-based recommender systems perform better compared to standard CF in a recommendation experiment. We also introduced a new citation graph data set, compiled from the CiteSeer digital library [4], that consists of multiple types of information such as textual information, author and venue information, citation context information, in addition to the citation graph data, which is made available to the research community.

In future, because SVD allows for easy incorporation of additional information, we plan to integrate other types of information (e.g., textual information, author, venue) into our models. The compiled data set facilitates further experimentation with more complex models that are able to exploit such information. It would also be interesting to see how other recommendation algorithms perform on the compiled CiteSeer data set, and design new ones for this task.

6. REFERENCES

[1] S. Bethard and D. Jurafsky. Who should i cite? learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management, CIKM '10*, 2010.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[4] C. L. Giles, K. Bollacker, and S. Lawrence. Citeseer: An automatic citation indexing system. In *Digital Libraries '98*, pages 89–98, 1998.

[5] Google. Google scholar. In *http://scholar.google.com*.

[6] Q. He, J. Pei, D. Kifer, P. Mitra, and C. L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World Wide Web '10*, pages 421–430, 2010.

[7] W. Huang, S. Kataria, C. Caragea, P. Mitra, C. L. Giles, and L. Rokach. Recommending citations: Translating papers into references. In *Proceedings of the 21st ACM CIKM '12*, 2012.

[8] S. Kataria, P. Mitra, and S. Bhatia. Utilizing context in generative bayesian models for linked corpus. In *Proceeding of AAAI*, 2010.

[9] Y. Lu, J. He, D. Shan, and H. Yan. Recommending citations with translation model. In *Proceedings of CIKM '11*, pages 2017–2020, 2011.

[10] S. M. McNee, I. Albert, D. Cosley, P. Gopalkrishnan, S. K. Lam, A. M. Rashid, J. A. Konstan, and J. Riedl. On the recommending of citations for research papers. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work '02*, pages 116–125, 2002.

[11] R. M. Nallapati, A. Ahmed, E. P. Xing, and W. W. Cohen. Joint latent topic models for text and citations. In *Proc. of KDD*, pages 542–550, 2008.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system a case study. In *WebKDD-2000 Workshop*, 2000.

[13] P. Smolensky. Parallel distributed processing: explorations in the microstructure of cognition. volume 1, pages 194–281. 1986.

[14] T. Strohman, W. B. Croft, and D. Jensen. Recommending citations for academic papers. IR 466, 2006.

[15] J. Tang and J. Zhang. A discriminative approach to topic-based citation recommendation. In *Proceedings of the 13th PAKDD '09*, pages 572–579, 2009.

[16] S. Teufel, A. Siddharthan, and D. Tidhar. Automatic classification of citation function. In *Proceedings of EMNLP-06*, 2006.

[17] B. Webb. Netflix update: Try this at home. In *http://sifter.org/∼simon/journal/20061211.html*, 2006.