Personalized Academic Paper Recommendation System

Joonseok Lee
Google Inc.
Mountain View, CA, USA
joonseok@google.com

Kisung Lee
Georgia Institute of Technology
Atlanta, GA, USA
kslee@gatech.edu

Jennifer G. Kim
University of Illinois, Urbana-Champaign
Urbana, IL, USA
jgkim2@illinois.edu

Sookyung Kim
Georgia Institute of Technology
Atlanta, GA, USA
skim722@gatech.edu

ABSTRACT

Recommendation systems can take advantage of social media in various ways. One common example is combining social relationship into neighborhood-based recommendation systems, under the assumption that social relationship affects individuals' interest or preference. Although this assumption may not be always true, this paper presents a realistic application, personalized academic paper recommendation system, which social relationship can be closely related to taste. There is an increasing number of academic papers being published each year, but most researchers rely on keyword-based search or browsing through proceedings of top conferences and journals to find their related work. Personalized academic paper recommendation system is designed to reduce their workload. With a collaborative-filtering-based approach, it recommends potentially preferred articles for each researcher in personalized manner. Both computational evaluation and user study demonstrate that our system recommends a useful set of research papers.

Categories and Subject Descriptors

H.2.8 [**Database applications**]: Data mining

Keywords

recommendation systems, academic papers, user study

1. INTRODUCTION

Recommender systems are widely used these days in e-commerce, for the purpose of targeted advertizement. Based on each user's profile, previous purchase history, and online behaviors, they recommend products which they are likely to prefer. For example, Amazon.com recommends related products such as books, and Netflix recommends movies that each user is interested in.

SRS'15, August 10, 2015, Sydney, NSW, Australia.

Personalized recommendation can be applied to outside of commercial applications. These days, many academic papers are coming out from a lot of conferences and journals. Academic researchers should go through conferences and journals which are related to their field of research and find out if there is any new article that may relate to their current works. Sometimes they search the articles from Google scholars or Citeseer with keywords that might show interesting articles to them. However, this approach requires users to commit their time to search articles, which is labor-intensive, and also do not guarantee that they will find the exact articles related to their field of research.

In order to reduce their workload, we propose a scholarly paper recommendation system for academic researchers, which will automatically detect their research topics they are interested in and recommend related articles they may be interested in, based on similarity of the works. We believe this system will save time to search the articles and reduce a chance to miss a relevant article.

2. RELATED WORK

Recommender systems have concentrated on recommending media items such as movies, but recently they are extending to academia. One of the most popular applications is citation recommendation [1, 9, 13, 11]. Recently, Matsatsinis et al.[8] introduced scientific paper recommendation using decision theory. Sugiyama et al.[12] extended scholarly paper recommendation with citation and reference information.

Collaborative filtering (CF) uses only user-item rating matrix for predicting unseen preferences. Neighborhood-based approaches predict based on similar users or items to the query. Model-based CF builds a model such as matrix factorization, which is known as the most efficient and accurate. [6] Lee et al.[5, 4] introduced local approaches for matrix factorization. Content-based methods, on the other hand, make use of user or item properties. [2] Hybrid approach is a combination of CF and content-based approach. Koren et al.[3] proposed effectively combining rating information and user, item profiles for more accurate recommendation.

3. METHOD

Figure 1 shows the flow of our system. First, our system gathers data and preprocess it, by applying the bag-of-words model to the corpus. In actual learning process, we apply

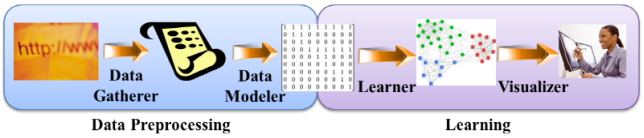

Figure 1: Recommendation Flow

lazy learning method similar to k-Nearest Neighbors (kNN). That is, we estimate preference of a target user and recommend the most preferred papers for each queried user. For this task, we applied clustering and neighbor-based recommendation algorithm. Finally, the result is conveyed to the user by visualizer. In this section, we describe each component in detail.

3.1 Data Model

We model the data with a bag-of-words. In this model, each word appearing in the entire corpora becomes an attribute. Each document is represented by a bit vector, indicating whether each word appears or not. This model is based on two assumptions: 1) the probability for a word to appear is independent of its nearby words (Naive Bayes assumption), and 2) the probability of encountering a specific word is independent of its position. Strictly speaking, these assumptions are obviously incorrect, but it is known that this does not seriously affect classification or learning task. A paper is represented as a set of words from its title, key words, and abstract.

We additionally apply some preprocessing to this bag-of-words. First, we remove stop words such as *the* or *of*. These words appear in almost every document in English, so they are not useful for classifying or filtering specific documents. We removed about 140 words selected manually.

We also apply stemming. In English, a same word can be used as different parts, usually in slightly different forms. For example, *clear*, *clearly*, and *cleared* have same meaning, but used in different forms depending on its position or role in the sentence. It is much better to deal with these minor changes of forms as same words, as it can dramatically reduce the dimensionality. However, this work is not straightforward. For simplicity, we just removed last *-ed*, *-ly*, and *-ing* from the word, whenever encountered.

3.2 Learner

As a perspective of recommendation system, we can consider authors as users and papers as items. We will use these terms interchangeably henceforth. We can think of recommendation system as a task of filling out missing preference data on a user-item matrix, based on observed values. There can be lots of schemes to predict these missing values. Filling with the user's average or item's average can be a simple baseline. In this section, we discuss fundamental characteristics of our problem, and then describe our algorithm.

3.2.1 Sparsity and One-class Nature

The information we gather contains each paper's title, list of authors, key words, and abstract. In order to build a user-item matrix with this data, we basically assume that users are interested in papers they published or they cited. Thus, we set a high score α to every ⟨researcher, authored paper⟩ pair, and another score $\beta < \alpha$ to ⟨researcher, cited paper⟩ pairs.

We claim that this user-item matrix is extremely sparse, which means most values are missing while only small portion of them are observed. This situation is common in recommendation. According to Netflix Prize data, only 1% of the user-item matrix are observed. Nonetheless, it has been shown that it is possible to accurately estimate missing data only using small portion of observed data. In our situation, however, the sparsity can be more severe. In most cases, one author publishes only one or two papers in one conference proceeding. There are only at most two or three top-level conferences in each field, the average number of papers one author usually publishes a year is very small. For this reason, the matrix should be extremely sparse.

Another issue is one-class nature [10] of this data; that is, we do not have negative feedback. When we request users to explicitly rate items in a common recommendation system (e.g, movie recommendation), we can get both positive and negative feedback from the user. For example, we can get *very like* feedback for the movie *Titanic* as well as *very hate* one for the *Shrek 2*. Based on this variety, we can infer that the user may prefer romance movies to animations. In our data, however, we do not have negative feedback. This problem makes difficult for us to use many collaborative filtering algorithms.

3.2.2 Recommender

We basically assume that authors like papers similar to ones they published before. As similar papers convey similar topic, we regard papers with similar set of words as similar ones. Our algorithm directly applies this idea as follows. Consider a simple case of a user with only one publication. To generate a list of recommendations to this user, we first calculate similarity between his own paper and all other candidate papers. These similarity values are considered as the score of each candidate. We retrieve the most similar k items to the target user's previous papers, similar to the K-Nearest Neighbors (KNN) algorithm. For similarity measures, any vector distance measure such as vector cosine or Pearson correlation can be used.

If the target user has published more than one papers previously, we first cluster the set of candidate papers. With this step, all candidate papers are assigned to only one of the most similar paper published by the target researcher. We used a simple K-means clustering. After assigned to a cluster, the score is calculated based on the distance between the candidate paper and its centroid. For example, as shown in Figure 2, each big circle represents a centroid of a cluster, and small circles connected to the centroid are members of its cluster. Using the calculated score as a distance metric for KNN, we select K papers for recommendation to the target user.

To illustrate, assume that the user u published two papers (x_1 and x_2). We would like to estimate how much the user u is likely to see a recent paper y. First, we calculate the similarity between y and x_1, denoted as s_1. We also calculate similarity s_2 between y and x_2. Then, we compare s_1 and s_2. We set estimated preference of y by user u as $\max(x_1, x_2)$. This is because the user may like a paper when it is related to his one of the favorable topics. Although it is not related to other papers, the author may still like it if it is related to at least one of the topics in which he is interested. More

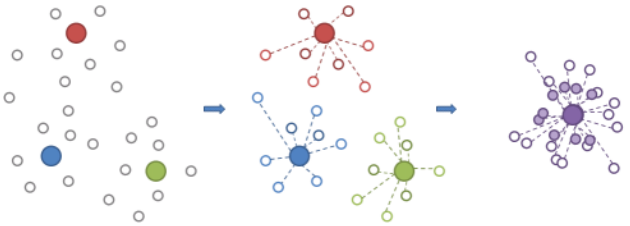

Figure 2: Visualization of Clustering and kNN

Table 1: Crawled Data

Area	Paper	Author	Conferences (Years)
ML	3,644	5,786	ICML(04-09), KDD(04-10), COLING(04-10), UAI(04-09), SIGIR(04-10), JMLR(04-11)
HCI	2,557	4,728	CHI(03-09), ASSETS(02-09), CSCW(04-11), UIST(04-10) Ubicomp(07-10)
DB	4,156	7,213	ICDE(06-10), SIGMOD(06-10), VLDB(06-10), EDBT(08-11), PODS(06-10), CIKM(06-10)

formal formula for estimating preference of user u for item i is given as

$$\max_{i,j}\left(\frac{C\cdot s(x_i,y_j)}{\max_{a,b}\left[s(a,b)\right]}\right), \quad (1)$$

where i is the index of a paper that the user u has published, j is the index of candidate papers, and $s(a,b)$ is a similarity function between item a and b. C is a constant, which may have different value for authored papers and referenced papers.

4. EVALUATION

We evaluate our system in two ways: measuring a classification accuracy (computational experiment) and a user study with real researchers. For our data corpora, we implemented a web data crawler from both IEEE Xplore and ACM Digital Library. We crawled papers from three different areas in computer science: machine learning (ML), human computer interaction (HCI), and database (DB). The set of users consists of all authors of this corpora. Table 1 summarizes the nature of crawled data. For similarity measure, we used vector cosine.

4.1 Classification Accuracy

We formulate our recommendation task as a classification problem with users and papers from three different areas. Specifically, we observe how many papers are recommended from the researcher's own area when we recommend papers from mixed corpora, assuming that each user is interested mainly on his own area.

We recommended 10 papers out of 10,386 candidates in Table 1 to 10 researchers in each area. The result is shown in Table 2. In overall, our system recommended papers from correct area with accuracy of 89%. One thing to note is that, however, a recommendation from different areas may not be an incorrect result, as some researchers actually do research cross over different areas. For example, data mining research is highly related both to ML and DB.

Table 2: Classification Accuracy

Researchers	ML Paper	HCI Paper	DB Paper	Accuracy
ML	84	0	16	84%
HCI	3	88	9	88%
DB	4	1	95	95%

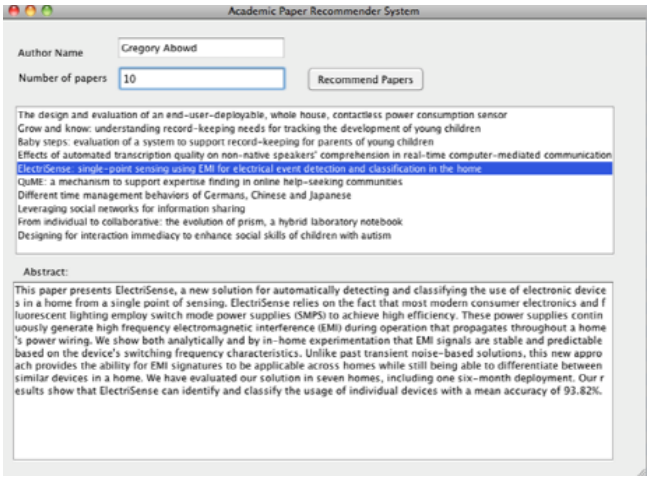

Figure 3: Graphical User Interface Prototype

4.2 User Study

In order to verify our system, we conducted a focus group user study by interviewing three professors, each from ML, DB, and HCI. One professor is a junior professor, and the other two are senior professors. Figure 3 shows our system prototype. When an author name is entered in the top text box, it retrieves a set of recommended papers.

We provided a list of 10 recommendations to each participant with title, authors, proceedings name, and abstract of those papers. The participants are asked to read and answer to our questionnaire described below. We used a Likert-scale between 1 (not relevant at all) and 6 (perfectly relevant), in order to prevent voting to middle-way.

First, we asked how much the recommendations are relevant to their research topics in general. As shown in Table 3, the three subjects indicated that recommended papers are related to their research. Subsequently, we asked how much the list was relevant to the researcher's previous research and current research separately. When we asked how the recommended paper list is related to their previous research, all gave higher than 5 point. (5.5, 5.0, 5.5) However, for the relevance of their current research topics, even though two professors gave 5.0 and 5.5, respectively, the other gave 2.0. In this case, she has worked on various topics before, so our system recommended papers that are relevant to topics she is not currently working on. Among 10 recommended papers, there were only two papers related to current research topics. In overall, however, all the professors were satisfied with the results of the recommended papers in respect of the topic relevance to their research.

To evaluate the usefulness of the recommended papers, we asked them to indicate the number of papers they would take time to read among the recommendations. Realistically, they replied that they are willing to read only about 2 out of 10 recommended papers that are highly related to

their current research. This number seems promising, as those papers may not have been known without our recommendations.

Lastly, we asked how much they are satisfied with the system in overall and how much they are willing to use the system. All of the subjects marked 6.0 point out of 6.0 to use this recommendation system, indicating that our research is valuable for real users.

One thing to note is that our system recommended to senior professors four papers that their previous students have published. As their previous students graduated, they did not publish those recommended papers together with the professor, but the topic of those papers is relevant to what they have done with the professor. This observation implies that collaborative filtering based paper recommendation system is likely to discover social relationships between users (researchers), and therefore we can further improve recommendation quality if we directly incorporate social relationship data. We leave this as a future work.

Table 3: User Satisfaction about Relevance

Subjects	Average	Standard Deviation
Subject 1	4.40	1.26
Subject 2	4.00	1.56
Subject 3	3.25	1.72
Total	3.88	1.52

5. DISCUSSION AND FUTURE WORK

Even though our system showed promising performance for real application, it can be improved in several ways. First, we can apply more complex language models. As we count only the frequency of words without any weight, it may miss some rare but important words representing a specific research area.

Another limitation is that we do not take advantage of temporal information of papers. In many cases, research topic changes as time goes. Many researchers, therefore, are no longer interested in topics they have worked in the past. Because our system does not have any information about whether the user is still interested in the paper or not, it is hard to distinguish topics to be recommend or not. One way to solve this issue is applying publication year with decay, as we can naturally expect that recently-published papers would have higher probability to be ongoing research topics.

Our current system takes no user information into account. During the focus group interview, researchers showed great interest to their peers' papers, although the topics are not that relevant. Making use of social information such as co-authorship graph can improve recommendation quality, especially in terms of user satisfaction.

Although we used a static dataset, in that preference data is extracted only from authorship and citations, this can be dramatically improved by incorporating interactions with users. If the system can collect feedback from users about recommended papers, we will have negative feedbacks as well as positive ones. This enables use of many collaborative filtering techniques, in addition to one-class ones.

Scalability can be another important issue. Although our current system runs quickly on small-scale crawled dataset, it may take longer time on larger dataset for real-world service. For faster computation, dimensionality reduction of the contents matrix may be applied. More sophisticated preprocessing of words will be also useful to reduce dimensionality.

6. SUMMARY

In this paper, we presented a personalized academic paper recommendation system, which recommends related articles for each researcher. With our system, researchers can get notified their related papers without searching keywords on web search engines or browsing from conferences proceedings. Our contribution is not limited to effective recommendation engine, but also to evaluation methodology covering both accuracy of recommendation and user satisfaction. In particular, we showed promising experimental results that we can take advantage of social relationship between researchers such as co-authorship. Both offline and online evaluation indicates that this is a potentially useful application to researchers, opening a way to incorporate more social relationship data with more sophisticated methods.

7. REFERENCES

[1] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proc. of the International Conference on World Wide Web*, 2010.

[2] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. *Recommender Systems - An Introduction*. Cambridge, 2011.

[3] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 426–434, 2008.

[4] J. Lee, S. Bengio, S. Kim, G. Lebanon, and Y. Singer. Local collaborative ranking. In *Proc. of the International Conference on World wide web*, 2014.

[5] J. Lee, S. Kim, G. Lebanon, and Y. Singer. Local low-rank matrix approximation. In *Proc. of the International Conference on Machine Learning*, 2013.

[6] J. Lee, M. Sun, and G. Lebanon. A comparative study of collaborative filtering algorithms. *ArXiv Report 1205.3193*, 2012.

[7] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[8] N. F. Matsatsinis, K. Lakiotaki, and P. Delias. A system based on multiple criteria analysis for scientific paper recommendation. In *Proc. of the Panhellenic Conference in Informatics*, 2007.

[9] 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 *Proc. of the ACM conference on Computer Supported Cooperative Work*, 2002.

[10] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Proc. of the IEEE International Conference on Data Mining*, 2008.

[11] T. Strohman, W. B. Croft, and D. Jensen. Recommending citations for academic papers. In *Proc. of the International ACM SIGIR Conference*, 2007.

[12] K. Sugiyama and M.-Y. Kan. Scholarly paper recommendation via user's recent research interests. In *Proc. of the Joint Conference on Digital Libraries*, 2010.

[13] J. Tang and J. Zhang. A discriminative approach to topic-based citation recommendation. In *Advances in Knowledge Discovery and Data Mining*, 2009.