Research Paper Recommendation with Topic Analysis

Chenguang Pan, Wenxin Li
Computer Science Department
Peking University
Beijing, China
{cgp, lwx}@pku.edu.cn

***Abstract*—With the collaborative filtering techniques becoming more and more mature, recommender systems are widely used nowadays, especially in electronic commerce and social networks. However, the utilization of recommender system in academic research itself has not received enough attention. A research paper recommender system would greatly help researchers to find the most desirable papers in their fields of endeavor. Due to the textual nature of papers, content information could be integrated into existed recommendation methods. In this paper, we proposed that by using topic model techniques to make topic analysis on research papers, we could introduce a thematic similarity measurement into a modified version of item-based recommendation approach. This novel recommendation method could considerable alleviate the cold start problem in research paper recommendation. Our experiment result shows that our approach could recommend highly relevant research papers.**

Keywords: collaborative filtering, research paper recommendation, latent dirichlet allocation, topic model, cold start

I. INTRODUCTION

Recommender system appeared in the early 1990s [1][2][3][4], and now they are playing a significant role in people's daily lives. Online shopping site Amazon.com and online movie rental company Netflix recommend commodities and services by analyzing customers' preferences and online behaviors. Social networking services like [14] may suggest that you may know some people or would possibly like someone to become your friend. There are also interests-based social networking site like [15] provides users with recommendations of books, music CDs, movies, and articles, and recommendations of people who might share the similar tastes based on users' ratings for the mentioned items.

In colleges, universities and research institutions, graduate students, professors, and other researchers need to find the papers most relevant to their research projects. Thus, looking for the right papers to read becomes a very important part of their academic lives. A research paper recommender system will benefit these people in helping the most relevant papers and saving their precious time. Unluckily, paper research recommender systems have not received enough attention. The currently rare available work includes [16], [17], [18], [19] etc. All these work have not taken thematic information of research papers into recommendation. And there is even no appropriate dataset available for the overall evaluation of research paper recommendation algorithms.

A recommender system primarily exploits two kinds of information: users' ratings for items, profiles of users and/or items. Memory-based recommendation makes use of only the users' ratings for items, say, the user-item matrix, with each entry the rating of a user for a certain item. There are basically two kinds of memory-based recommendations: user-based recommendation and item-based recommendation [6] [7]. User-based recommendation attaches higher weights for users who share the similar rating patterns with the active user and calculate the utilities of new items from these weighted users. Item-based recommendation evaluate an item's utility for a user by first selecting its most similar items that have been rated by the user, then compute the utility as the weighted average of the rating of these similar items. For an online recommender system, when the number of the users grows to a considerable amount, the computation of similarity of every user pair's rating patterns would be very time consuming, thus item-based recommendation could provide a feasible approach for it computes the similarities between items offline. Reference [8] gives a nice comprehensive survey on recommender systems.

Research paper recommendation would not only utilize the user-item matrix but also the content information of the research papers. We use topic model techniques to make topic analysis of research papers and introduced the similarity over topics between research papers as the thematic similarity. By incorporating the thematic similarity and a modified version of item-based method, we could successfully generate highly relevant recommendations and greatly relieve the cold start problem.

II. COLLABORATIVE FILTERING

References [5][6][7][8][9] describe the details of the two approaches of memory-based recommendations: user-based recommendation and item-based recommendation, very well. For later convenience we will still elaborate on these two approaches.

In a typical recommender system, we have a user set U consists of N users, an item set I consists of M items and a rating set R in which r_{ui} stands for user u's rating for item i. $S_u \subseteq I$ stands for the set of items user u has rated. Collaborative filtering algorithms attempt to compute the expected ratings or say, utilities of items that has not been rated by the active user, then present the active user with the

items with the highest utilities. We will describe below the details of relevant memory-based recommendations: user-based recommendation, item-based recommendation and our modified version of item-based recommendation.

A. User-based Recommendation

In a collaborative filtering algorithm we are to compute the utility p_{ai} of the active user a on a given item i that has not been rated by this user. Since the item i is not rated by user a, thus we have $i \notin S_a$. For user-based recommendation, the utility of an unrated item for a user is based on the ratings of that item of the nearest neighbors of the user. A similarity measure between users need to be defined and a set of nearest neighbors to be selected. Then we combine the ratings of these neighbors on the item in some way.

The way we can define the similarity between users will be described in A.1). Let $sim(a, u)$ be similarity between users a and u. The number of neighbors to be considered is often set by a system parameter that we denote by K, K is often set to N, the total number of users in the system. So the set of neighbors of a give user a, denoted by T_a, is made of the K users that maximize their similarity to user a.

We define the utility of item i for user a as the weighted sum of the ratings of the nearest neighbors $u \in T_a$ that have already rated item i:

$$p_{ai} = \frac{\sum_{\{u \in T_a | i \in S_u\}} sim(a,u) \cdot r_{ui}}{\sum_{\{u \in T_a | i \in S_u\}} |sim(a,u)|} \tag{1}$$

In order to take into account the difference in user of rating scale by different users, predictions based on deviations from the mean ratings have been proposed. In that case, p_{ai} is computed using the sum of the user mean rating and the weighted sum of deviations from their mean rating of the neighbors that have rated item i:

$$p_{ai} = \overline{r_a} + \frac{\sum_{\{u \in T_a | i \in S_u\}} sim(a,u) \cdot (r_{ui} - \overline{r_u})}{\sum_{\{u \in T_a | i \in S_u\}} |sim(a,u)|} \tag{2}$$

$\overline{r_u}$ represents the mean rating of user u:

$$\overline{r_u} = \frac{\sum_{\{i \in S_u\}} r_{ui}}{|S_u|} \tag{3}$$

1) Similarity Measurements

The similarity defined between users or items include but are not limited to the following ways:

Pearson correlation:

$$pearson(a, u) = \frac{\sum_{\{i \in S_a \cap S_u\}} (r_{ai} - \overline{r_a}) \times (r_{ui} - \overline{r_u})}{\sqrt{\sum_{\{i \in S_a \cap S_u\}} (r_{ai} - \overline{r_a})^2 \sum_{\{i \in S_a \cap S_u\}} (r_{ui} - \overline{r_u})^2}} \tag{4}$$

Simple cosine:

$$cosine(a, u) = \frac{\sum_{\{i \in S_a \cap S_u\}} r_{ai} \times r_{ui}}{\sqrt{\sum_{\{i \in S_a \cap S_u\}} r_{ai}^2 \sum_{\{i \in S_a \cap S_u\}} r_{ui}^2}} \tag{5}$$

There are also other similarity definitions as Manhattan similarity, Jaccard similarity, Tanimoto score, but we will not describe them here.

B. Item-based Recommendation

In the item-based approach, p_{ai}, the utility of an unrated item i for the user a is computed as follows:

We choose the most similar rated K items by user a for item i. We denote the set of these K items as I_K. r_k stands for rating of item $i_k \in I_K$. So the utility of the unrated item i is:

$$p_{ai} = \frac{\sum r_k \times \mathrm{sim}(i, i_k)}{\sum \mathrm{sim}(i, i_k)} \tag{6}$$

Like the user-based approach, we also have a method to compute utility of i using deviation from the mean rating given by user a.

If we use the recommendation provided by formula (6), we could easily find out that when the user has just rated one item or has given a few items the same rating, utilities of all the unrated items will be the same as the given rating. In that case no unrated item could be given preference in recommendation.

Since we hope our recommendation could give reliable recommendations even when the user has just rated very few items or even one item. We will use a modified version of item-based recommendation:

$$p_{ai} = \frac{\sum r_k \times \mathrm{sim}(i, i_k)}{K} \tag{7}$$

This formula simply weights the ratings of items according to their respective similarity with the current item being evaluated, and computes an average.

III. TOPIC ANALYSIS

Topic models [10][11] are based upon the idea that with in text corpora, a document is a mixture of topics, where a topic is a multinomial distribution over words. The latent dirichlet allocation (LDA) model (or "topic model") is a generative model for documents: it specifies a simple probabilistic procedure by which documents can be generated. The mixing coefficients for each document and

the word-topic distributions are unobserved and are learned from data using unsupervised learning methods.

LDA can be described as finding a mixture of topics for each document, i.e., P(z | d), with each topic described by words following another probability distribution, i.e., P(t | z). This can be formalized as

$$P(t_i \mid d) = \sum_{j=1}^{z} P(t_i \mid z_i = j) P(z_i = j \mid d) \quad (8)$$

where $P(t_i \mid d)$ is the probability of the ith word for a given document d and z_i is the latent topic. $P(t_i \mid z_i = j)$ is the probability of t_i within topic j. $P(z_i = j \mid d)$ is the probability of generating a word from topic j in the document d. The number of latent topics Z is defined in advance and controls the granularity of differences among latent topics. LDA estimates the topic-word distribution P(t | z) and the document-topic distribution P(z | d) from an unlabeled corpus of documents using Dirichlet priors for the distributions and a fixed number of topics.

Reference Blei et al [11] introduced the LDA model within a general Bayesian framework and developed a variational algorithm for learning the model from data. Reference Griffiths and Steyvers [12] subsequently proposed a learning algorithm based on Gibbs sampling. The Gibbs sampling approach iterates multiple times over each word t_i in document d_i, and samples a new topic j for the word based on Equation (8), until the LDA model parameters converge.

$$P(z_i = j \mid t_i, d_i, z_{-i}) \propto \frac{C^{TZ}_{t_i j} + \beta}{\sum_t C^{TZ}_{tj} + T\beta} \frac{C^{DZ}_{d_i j} + \alpha}{\sum_z C^{DZ}_{d_i z} + Z\alpha} \quad (9)$$

C^{TZ} maintains a count of all topic-word assignments, C^{DZ} counts the document-topic assignments, z_{-i} represents all topic-word and document-topic assignments except the current assignment z_i for word t_i, α and β are the hyperparameters for the Dirichlet priors, serving as smoothing parameters for the counts. Based on the counts the posterior probabilities in Equation (8) can be estimated as follows:

$$P(t_i \mid z_i = j) = \frac{C^{TZ}_{t_i j} + \beta}{\sum_t C^{TZ}_{tj} + T\beta} \quad (10)$$

$$P(z_i = j \mid d_i) = \frac{C^{DZ}_{d_i j} + \alpha}{\sum_z C^{DZ}_{d_i z} + Z\alpha} \quad (11)$$

Simply put, we are going to use Gibbs sampling methods to compute the mixture coefficients of each topic in a document, and the coefficients of a topic's multinomial distribution over words.

Say we have D documents in the corpora, W unique words appeared within, N words in total, and we suppose there are K topics in the corpora. Using Gibbs sampling for LDA we could obtain the results as follows:

θ [j, k], the mixture coefficient for topic k in document j.

φ [k, w], the multinomial coefficient for word w in topic k.

$j \in [1, D]$, $w \in [1, W]$, $k \in [1, K]$.

We define the thematic similarity between two research papers p and q as the cosine similarity defined over their topic mixture coefficient vector $\overrightarrow{\theta[p]}$ and $\overrightarrow{\theta[q]}$:

$$Thematic\_Sim(p, q) = \frac{\overrightarrow{\theta[p]} \cdot \overrightarrow{\theta[q]}}{\left\|\overrightarrow{\theta[p]}\right\| \left\|\overrightarrow{\theta[q]}\right\|} \quad (12)$$

As we know, there are a variety of ways to define the similarity between vectors, but in our experiment we simply choose simple cosine similarity for the computation of recommendations.

In our recommendation method we use thematic similarity to replace the original similarity get from user-item matrix, combined with modified item-based method, we could make research paper recommendations for users. This recommendation method could considerably alleviate cold start problem of recommender systems. In a typical recomender system, cold start problem has two situations. One is that when a user has just ranked very few items, it is very difficult to generate recommendations accurately due to the little information of the user we have. The other situation is that when a new item has just been registered in the system, no user has yet got time to rate the item, so that this item will not get the slightest chance to be recommended. For most recommender system the first situation is more severe and more common than the second situation. Our method could relieve both the two situations greatly, especially the first situation.

IV. EXPERIMENT

Our experiment is carried in the following procedure and we will demonstrate the result all along, and we will make data and result available online for the convenience of other researchers.

We take 122 research papers from the research paper collection of our laboratory, mainly on biometrics research, as the research papers that are going to be used in our recomender system. They are all in Portable Document Format (PDF).

Then all these PDF files are converted into text files using a Java library called Apache PDFBox [22]. These text can be download at [20].

We preprocessed these text files and use a perl script by David Newman to make these research papers into UCI Bag-Of-Words Dataset format [21]. The dataset generated by us is available here [20].

We implemented the Gibbs sampling algorithm to process the dataset we generated. We took 40 topics, and α =2.0/40, β =0.01 as the parameters in our inference. We successfully computed the coefficient of topic mixture in each research paper and the multinomial coefficient for each

TABLE I. TOPICS MIXTURE IN DOC. 59

TOPIC ID	TOPIC WEIGHT
32	0.70430677
27	0.12807462
26	0.06082450
5	0.04884846
38	0.02535698
8	0.01476278
35	0.00831414
23	0.00370797
6	0.00324735
10	0.00140488
33	0.00048365
11	0.00002303

word to be generated in each topic. Results are available at [20]. Table 1. shows the coefficients of topic mixture in Doc. 59: "A Novel Scheme for Fingerprint Identification" by Tsong-Liang Huang et al. Only 12 topics with highest mixture weight are show in Table 1. It is very obvious that this research paper is mainly about Topic 32 and Topic 27. Topics 32 and 27's probability to generate each word is shown in Table 3 and 4. Only 20 words with the highest probability are shown.

Then we computed the thematic similarities between all possible pairs of research papers. Results are available here [20]. We can easily find that the thematic similarity between Doc. 59 and Doc.61 is 0.99821355. The topic weights of Doc. 61 are shown in Table. 2. Doc. 61 is "A Pixel-Level Automatic Calibration Circuit Scheme for Capacitive Fingerprint Sensor LSIs" by Hiroki Morimura et al. We can easily find out that these two papers are quite similar in their main topics.

In the recommendation scenario we consider a user that has only rated 3 papers: Doc. 14, 31, 59, with respective rating 1, -2, 2. We take rating –2, -1, 0, 1, 2 to stand for "awful", "bad", "so-so", "good", "awesome".

We successfully use our recommendation algorithm to generate the recommendations with the highest predicted utility for this user. The recommendation result is available here [20] and partly shown in Table 5 on next page.

Relevant Doc. 14, 31, 58, 55, 49, 48 are "Estimation and sample size calculations for correlated binary error rates of biometric identification devices" by Michael E. Schuckers, "Validating a Biometric Authentication System: Sample Size Requirements" by

TABLE II. TOPICS MIXTURE IN DOC. 61

TOPIC ID	TOPIC WEIGHT
32	0.65057699
27	0.13236567
26	0.06168410
38	0.03968626
5	0.03247386
24	0.02490083
35	0.01660656
21	0.01011540
28	0.00903354
8	0.00867292
16	0.00578796
23	0.00398485

TABLE III. WORDS GENERATION PROBABILITY IN TOPIC 32

WORD ID	PROBABILITY	WORD
3240	0.08928722	sensor
745	0.05155874	circuit
731	0.04073623	chip
1565	0.03562560	fingerprint
2704	0.03487404	pixel
3237	0.02705778	sensing
1870	0.02014340	image
2056	0.01773840	japan
1545	0.01668621	fig
633	0.01398058	calibration
3301	0.01232714	signal
1562	0.01067370	finger
3541	0.01067370	surface
2712	0.01007245	plate
404	0.00992214	area
781	0.00992214	cmos
1323	0.00977183	element
412	0.00886995	array
401	0.00871964	architecture
1551	0.00856932	film

TABLE IV. WORDS GENERATION PROBABILITY IN TOPIC 27

WORD ID	PROBABILITY	WORD
1857	0.05555788	ieee
2844	0.04356642	processing
3622	0.02695439	test
3301	0.02618429	signal
3085	0.02508416	research
2969	0.02255385	received
2841	0.02211380	process
1355	0.02035358	engineering
3597	0.01760325	technology
3795	0.01617307	university
2311	0.01320271	member
1317	0.01155251	electrical
1948	0.01133248	information
1047	0.01100244	data
1094	0.01089243	degree
1135	0.01056239	design
1189	0.00946225	digital
1992	0.00946225	interest
1111	0.00935224	department
872	0.00891219	computer

Sarat C. Dass et al., "A Multichannel Approach to Fingerprint Classification" by Anil K. Jain et al., "A Fingerprint Verification System Based on Triangular Matching and Dynamic Time Warping" by Zsolt Mikos Kovacs-Vajna, "A 600-dpi Capacitive Fingerprint Sensor Chip and Image-Synthesis Technique" by Jeong-Woo Lee et al., "Theoretical statistical correlation for biometric identification performance" respectively.

The recommendation list in Table 5 shows us that under the influence of Doc. 59, which got the highest rating 2, the Doc.61 which is highly thematically similar to Doc. 59 as demonstrated above, rank the first in the recommendation list. The utility of Doc. 61 is not as high as close to Doc. 59's rating is due to the influence of the worst rated item: Doc. 31.

TABLE V. TOP 5 RECOMMENDATIONS

DOC ID	PREDICTED UTILITY
61	0.66960195
58	0.66002356
55	0.49133007
49	0.37691359
48	0.37416252

The thematic similarity between Doc. 61 and lowest rated Doc. 31 reduced the utility of Doc. 61 to below 1 but still remains the highest. We can also see that utility of research papers with highest thematic similarity to Doc. 31 have very low or even negative utilities. For example, Doc. 21 named "Effects of User Correlation on Sample Size Requirements" by Sarat C. Dass and Anil K. Jain, is predicted a utility of -0.55714472, because this paper has a high thematic similarity at 0.99470842 with Doc. 31.

V. CONCLUSION

Recommender systems are a powerful technology for electronic commerce and social networks. They can help a business to increase sales revenue and help customers to find product to their liking, or help people to find friends who share the same taste for things. At the same time their potential impact on scientific research is largely unexplored. We proposed a novel method for research paper recommendation. We proved by our experiment that by making topic analysis on research papers and introducing thematic similarity we could recommend highly relevant papers and considerably alleviate the cold start problem. Even when the user has just rated very few papers, we could still generate satisfactory recommendations.

ACKNOWLEDGMENT

We would like to thank the developers of Apache PDFBox, and Dave Newman for his nicely working perl script. Thank you for your work, so that we don't need to do everything from scratch.

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki and D. Terry, "Using collaborative filtering to weave an information tapestry", Communication of the ACM 35 (1992), 61-70

[2] P. Resnick et al., "GroupLens: an open architecture for collaborative filtering of netnews", Proc. ACM 1994 Conf. Computer Supported Cooperative Work, ACM Press, 1994, 175-186

[3] U.Shardanand, P. Maes, "Social information filtering: algorithms for automating 'word of mouth'", ACM 1995 Conf. Human Factors in Computing Systems, Vol1, 210-217

[4] J.A. Konstan et al. "GroupLens: applying collaborative filtering to Usenet news", Comm. ACM, Vol40,no.3,77-87,1997

[5] J.S. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering", Proc. 14th Conf. Uncertainty in Artificial Intellligence, July 1998

[6] B.Sarwar, G. Karypis, J.Konstan and J. Riedl, "Item-Based collaborative filtering recommendation algorithms", Proc. 10th Int'l WWW Conf. 2001

[7] G. Linden, B. Smith and J. York, "Amazon.com recommendations: item-to-item collaborative filtering", IEEE Internet Computing, Jan./Feb. 2003

[8] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible Extensions", IEEE Trans. on Knowledge and Data Engineering 17, (2005), 634-749

[9] L. Candillier, F. Meyer and F. Fessant, "Designing specific weighted similarity measures to improve collaborative filtering systems", ICDM 2008, LNAI 5077, 242-255

[10] T. Hoffman, "Probabilistic latent semantic analysis", UAI'99

[11] D. Blei, A. Ng and M. Jordan, "Latent dirichlet allocation", Journal of Machine Learning Research, 3:993-1022, Jan. 2003

[12] T. Griffiths and M. Steyvers, "Finding scientific topics", PNAS/ USA, 101 Supp l1:5228-5235, Apr. 2004

[13] I. Porteous et al. "Fast collapsed Gibbs sampling for latent dirichlet allocation", KDD'08 Aug. 2008

[14] www.facebook.com

[15] www.douban.com

[16] C. Basu, H. Hirsh, W.W. Cohen and C. Nevill-Manning, "Technical paper recommendation: a study in combining multiple information sources", Journal of Machine Learning Research 1 (2001) 231-252

[17] T. Y. Tang and G. McCalla, "A multidimensional paper recommender – experiments and evaluations", 2009 IEEE Internet Computing

[18] M. Gori and A. Pucci, "Research paper recommender systems: a random-walk based approach", Proc. of 2006 IEEE/WIC/ACM Int'l Conf. on Web Intelligence.

[19] T. Bogers and A. van den Bosch, "Recommending scientific articles using CiteULike", ACM Recsys'08

[20] https://docs.google.com/uc?id=0Bw3EfwDJYdmXZDg3NDViYzctN2E3Mi00ODJkLWI2YTMtZjYxNDY3NDNkOTQx&export=download&hl=en

[21] http://archive.ics.uci.edu/ml/datasets/Bag+of+Words

[22] http://pdfbox.apache.org/