Research Article

A Probabilistic Recommendation Method Inspired by Latent Dirichlet Allocation Model

WenBo Xie,¹ Qiang Dong,^{1,2} and Hui Gao^{1,2}

¹ Web Sciences Center, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

² Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Qiang Dong; dongq@uestc.edu.cn

Received 27 May 2014; Accepted 15 August 2014; Published 29 September 2014

Academic Editor: Qinggang Meng

Copyright © 2014 WenBo Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The recent decade has witnessed an increasing popularity of recommendation systems, which help users acquire relevant knowledge, commodities, and services from an overwhelming information ocean on the Internet. Latent Dirichlet Allocation (LDA), originally presented as a graphical model for text topic discovery, now has found its application in many other disciplines. In this paper, we propose an LDA-inspired probabilistic recommendation method by taking the user-item collecting behavior as a two-step process: every user first becomes a member of one latent user-group at a certain probability and each user-group will then collect various items with different probabilities. Gibbs sampling is employed to approximate all the probabilities in the two-step process. The experiment results on three real-world data sets MovieLens, Netflix, and Last.fm show that our method exhibits a competitive performance on precision, coverage, and diversity in comparison with the other four typical recommendation methods. Moreover, we present an approximate strategy to reduce the computing complexity of our method with a slight degradation of the performance.

1. Introduction

The advent of Internet has confronted us with an exploding information era. We find that it is very difficult to select the relevant ones from countless candidates on the e-commerce websites. As an automatic way to help people make right decisions under the information overload, the recommendation system has become a significant issue for both academic and industrial communities.

During the last decade, lots of recommendation methods have been proposed, including collaborative filtering methods [1, 2], content-based methods [3], spectral methods [4, 5], and iterative refinement methods [6, 7]. These methods are all based on the computation of either user similarity or item similarity or both. Recently, some network-based recommendation methods have been proposed to mine the latent relevance of users and items, such as the methods based on mass diffusion or association rules [8, 9].

Latent Dirichlet Allocation (LDA) was first presented as a graphical model for text topic discovery by Blei et al.

in 2003 [10], which can be used to find the inherent relation of words and generate document set through the model. LDA has been widely used in document analysis [11–13], document classification, and document clustering [14–16]. LDA was first introduced into recommender systems for analyzing the context in content-based methods [17]. Now in tag-based recommendation systems, LDA is widely used to find the latent relation between keywords of item description and item tags created by users, such that the items can be recommended based on the tags [18–20]. For instance, Kang et al. [21] proposed an LA-LDA model which considers not only the tags created by the target user but also the tags created by his/her friends in the social network to extend the scope of candidate tags created by the target user.

In this paper, we propose a new content-unaware probabilistic recommendation method inspired by LDA model. Users' collecting behaviors are probabilistic events, in which one user belongs to multiple user-groups and users in each user-group have different collecting preferences. In our method, the collecting process is regarded as two joined

probabilistic processes intermediated by the user-group; that is, every user is a member of one latent user-group at a certain probability, while each user-group will collect various items with different probabilities.

Calculating the probabilities on the entire data set is time-consuming and space-consuming. In order to reduce the computing complexity of our method, we introduce an approximate strategy with a slight degradation of the performance, which samples a part of the data set to build a rough probabilistic recommendation model.

Many products on an e-commerce website are not popular; that is, the sale of every single product lies in the tail of sale curve, but the sales of all these unpopular products constitute a big portion of the whole income. That is the so-called long tail phenomenon. Therefore, a good recommender system must focus on both the accuracy and the diversity. The experiment results on three real-world data sets, MovieLens, Netflix, and Last.fm, show that our method exhibits a competitive performance not only on the precision and the coverage but also on the diversity.

2. Materials and Methods

2.1. Recommendation Model. People have different and multiple inner attributes, including physiological characteristics, preferences, taboos, and religious beliefs. These attributes can be clustered into lots of user-groups which can represent users with similar attributes. Actually, a user does not belong to only one user-group. For example, user A is a male and he is a high school student, minor, Chinese, and a Christian as well. One user belongs to multiple user-groups while users in different user-groups have different habits. For instance, users in the user-groups which contain the attribute of “elder” are more likely to buy health care products and presbyopia glasses than those who belong to the user-groups containing the attribute of “younger.” In our recommendation model, we put forward two assumptions:

- (1) the users’ collecting behaviors are probabilistic events;
- (2) one user belongs to multiple user-groups and users in the same user-group have similar collecting preference.

The collecting action of users on items is therefore considered as a two-step probabilistic process; that is, users are observed as members of several latent user-groups and users in user-groups will collect items based on the group-item probability distributions. Here we assume that z is a user-group which a user belongs to; θ is the probability vector between users and user-groups. Each column of the vector represents the probability that a user belongs to this user-group. The probability that a user d belongs to user-group z can be expressed as $p(z | d) = \theta_z^d$; φ is the commodity probability vector and each column of vector represents the probability that the users of current user-group will collect this item, while the probability that a user who belongs to user-group z will buy item w can be expressed as $p(w | z, \varphi^z) = \varphi_w^z$. In fact, θ reflects the degree of association between users and user-groups and φ shows which item the

users who belong to this user-group are more likely to buy. For example, a user who loves both basketball and music belongs to two user-groups, but he prefers to play basketball rather than listening to music. The association intensity between user and user-group could be demonstrated by the probability that the user belongs to each user-group. For a student, he may not care about household items but usually buy books for study. Based on the assumption above, if there are T groups, the probability for user d to buy item w can be expressed as

$$p(w | d) = \sum_z p(w | z) p(z | d) = \sum_z \varphi_w^z \cdot \theta_z^d. \quad (1)$$

As long as θ and φ are calculated, the collecting probability vector of the users can be computed. We can get a list ranked in order of the probabilities. Deducing from the list, a proper recommendation can be given to the users. In fact, it is not easy to calculate θ and φ directly.

Considering that, the Latent Dirichlet Allocation (LDA) is a probabilistic model that uses a latent topic to bridge documents and words. Using the latent topic, LDA constructs the documents via two probabilistic processes that chooses a topic after the first probability prediction and then collects words from the attributes of the topic-word according to the topic. Inspired by the LDA, the structure of our recommendation model is designed as a three-layer structure of Bayesian, that is, the user layer, followed by the user-group layer and the item layer. To construct it, parameters are used in pairs. The recommendation model is determined by the hyper parameters α and β , in which α describes the relative intensity between user-groups, $\theta \sim \text{Dirichlet}(\alpha)$, and β reflecting the probability distribution of each user-group, $\varphi \sim \text{Dirichlet}(\beta)$. The complete graphical model representation of LDA for probabilistic recommendation model is as shown in Figure 1. Indeed, we can construct the model without the items or user descriptions. So it is a content-unaware probabilistic recommendation model.

In our method, the probability that a user d has an attribute z is expressed as $p(z | d, \alpha) = \theta_z^d$; the probability that users who have attribute z purchased items w is expressed as $p(w | z, d, \beta) = \varphi_w^z$; and the probability that user d purchases item w is expressed as

$$p(w | d, \alpha, \beta) = \sum_z p(w | z, d, \beta) \cdot p(z | d, \alpha) = \sum_z \varphi_w^z \cdot \theta_z^d. \quad (2)$$

2.2. Parameters Estimation. There are many approximate inference strategies to estimate parameters θ and φ in LDA, such as Laplace approximate, variational inference, Gibbs sampling, and expectation propagation. Griffiths and Steyvers put forward that the perplexity and speed of Gibbs sampling method is better than those of other methods [11]. Since the structure of our method is similar to that of LDA, we chose Gibbs sampling algorithm to estimate parameters θ and φ as well. Gibbs sampling is a simple MCMC (Markov chain Monte Carlo) method. It constructs a Markov chain which converges to the target distribution and samples. Each

FIGURE 1: Graphical model representation of our method.

state of the Markov chain represents the value of the user-group, and the sampling variable is an implicit variable, which is assigned to items collected by users. The transition between states follows a simple rule. By sampling on the current values of all variables and data set of users' purchases, the chain can translate to the next state.

Here, we use the posterior distribution $p(z_j | z, d, w_i, \alpha, \beta)$, which is calculated by counting the user-groups assigned to items, as the transition probability of user-group shifts from z to z_j for item w which is purchased by user d , as shown in

$$p(z_j | z, d, w_i, \alpha, \beta) \propto \frac{n_{-i,j}^{(w_i)} + \beta}{n_{-i,j}^{(*)} + W\beta} \times \frac{n_{-i,j}^{(d)} + \alpha}{n_{-i,*}^{(d)} + T\alpha}. \quad (3)$$

Here, $n_{-i}^{(*)}$ is a count that does not include the assignment of item w ; $n_{-i,j}^{(w_i)}$ is the number of times that item w has been assigned to user-group j ; $n_{-i,j}^{(*)}$ represents the number of times that all items have been assigned to user-group j ; $n_{-i,j}^{(d)}$ is the number of times that items purchased by user d have been assigned to user-group j ; $n_{-i,*}^{(d)}$ represents the quantity of items that user d had purchased.

When the Markov chain is near the target distribution after adequate iterations, we recorded its current values of the implicit variable z and used it to estimate θ and φ as shown in (4) and (5):

$$\hat{\varphi}_j^{(w)} = \frac{n_j^{(w)} + \beta}{n_j^{(*)} + W\beta}, \quad (4)$$

$$\hat{\theta}_j^{(d)} = \frac{n_j^{(d)} + \alpha}{n_*^{(d)} + T\alpha}. \quad (5)$$

Here $n_j^{(w)}$ is the number of times item w has been assigned to user-group j ; $n_j^{(*)}$ represents the number of times that all items have been assigned to user-group j ; $n_j^{(d)}$ is the number of times that items purchased by user d have been assigned to user-group j ; and $n_*^{(d)}$ represents the quantity of items that user d has purchased.

2.3. Approximate Model. Actually, the data set is updated every day. It is not only time-consuming but also space-consuming to use the entire data set to structure the recommendation model. To save time and space, we prefer to model with less data and the recommended items are only listed when required instead of preparing them in advance. In this paper, we present an approximation method to structure the approximate model of the probabilistic recommendation.

In the approximation method, we sample part of the users' collection data from the data set to structure an imprecise probabilistic recommendation model. The imprecise model will serve as a guide to create a recommendation list from two sides. On one hand, the latent user-group vector z will be initialed by using the parameter φ of imprecise model. On the other hand, the transition probability $p(z_j | z, d, w_i, \alpha, \beta)$ is defined as the product of constant φ calculated in the imprecise model and iteration parameter θ , as shown in

$$p(z_j | z, d, w_i, \alpha, \beta) \propto \varphi_j^{(w_i)} \times \frac{n_{-i,j}^{(d)} + \alpha}{n_{-i,*}^{(d)} + T\alpha}. \quad (6)$$

The processes of the approximate method are described as follows.

- (1) Choose part of users from the data set for constructing the approximate model, called approximating data.

TABLE 1: The basic statistical features of the three data sets.

Data set	Users	Items	Links	Sparsity
MovieLens	6,040	3,952	1,000,209	4.19×10^{-2}
Netflix	10,000	6,000	701,947	1.17×10^{-2}
Last.fm	1,882	17,632	186,480	5.62×10^{-3}

- (2) Use the approximating data to initiate the Markov chain: random user-group z from 1 to T is assigned to each item i collected by user d .
- (3) Use (3) to iterate the Markov chain until it is converged. Equation (4) will be used to work out the φ and then used to construct the approximate model.
- (4) When user d needs a recommendation list, user-group z is assigned to each item i collected by user d . The parameter φ , collecting probability of the user-group, which was consequently worked out in Step (3), will be used to initialize z from 1 to T probabilistically. This is the initial state of the Markov chain for user d .
- (5) Use (6) to iterate for appreciable number of times, and we denote the result as b , also called burn-in space. It is thought that the Markov chain is near the target distribution. Then, we record the current values of z .
- (6) Sample once in a certain number of times c which is called thinning space [22]. According to (5), we can estimate θ , the purchasing probability of user d in each user-group.
- (7) By ranking the product of φ and θ , the recommendation list can be provided.

The time consumption of the approximate model depends on the size of approximating data. Indeed, the performance of the approximate model depends on the size as well. In the experiment, we use different percentages of data as approximation data to find the optimal size. Approximate model is, however, imprecise owing to the use of data locality. Meanwhile, the performance oscillates when different data is chosen to do approximate modeling. Different strategies (random: randomly choose users within the entire data; item degree: according to each user's average degree of items sampled proportionally; user degree: according to the average degree of user sampled proportionally; quick classification: use a quick classification method to classify the users and then sample proportionally) are compared to find out the user distribution offered by which strategy is most similar to that of the entire data and has the same tendency of performance. In the experiment, we use the average value and the upper bound value to represent the performance of the approximate model.

3. Experiments and Results

3.1. Data Description. Three benchmark data sets (Table 1) are used to evaluate the performance of the proposed LDA-based recommender method. The first data set *MovieLens* is

provided by GroupLens Project at the University of Minnesota. The second data set *Netflix* is a randomly selected subset of the huge data set released by the DVD rental company Netflix for its Netflix Prize. The third data set *Last.fm* was released in the framework of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). According to the chronological order of the data in MovieLens, we chose the early 80 percent of the data as the training set and the later 20 percent as the probe set. For the Netflix and Last.fm date sets, the data is randomly selected into two parts: the training set contains 90% of the data and the remaining 10% of the data constitutes the probe set in the experiment.

3.2. Evaluation Metrics and Comparison Methods. Three evaluation metrics were used to assess the recommendation's effect in the experiment: precision, coverage, and diversity. Precision is a basic evaluating metric. It is defined as the proportion of users that accept the recommended items:

$$P = \frac{R \cap U}{U}. \quad (7)$$

Here, R represents the list of items recommend to users; U represents the set of items that the user has bought.

Different algorithm will provide different recommendation list to users. The union set of recommendation lists R_i can be used to work out the proportion of recommended items in the entire item set. We use coverage to define this proportion, as shown in

$$\text{Cov} = \frac{\bigcup R_i}{A}. \quad (8)$$

Here, R_i represents the list of items recommended to user i and A represents the quantity of all items. Diversity is an important metric for personalized recommender systems. It is used to evaluate the difference between users' recommendation lists, and we use the average hamming distance of recommendation list to define diversity as follows:

$$H_{ij}(L) = 1 - \frac{R_i(L) \cap R_j(L)}{L}, \quad (9a)$$

$$\text{Div} = \frac{2 \sum_{i \neq j} H_{ij}}{I(I-1)}. \quad (9b)$$

$H_{ij}(L)$ is the hamming distance of recommendation list between user i and user j ; L is the length of the recommendation list; and I represents the quantity of users.

For comparison, we present the results of the four recommendation methods which are the probabilistic spreading (ProbS), heat spreading (HeatS), user-based collaborative filtering (UserCF), and the association rule algorithm (ARule). User-based collaborative filtering algorithm is one of the most classic collaborative filtering methods. Based on the similarity of purchased items between users, it recommends the items that similar users have bought but not yet bought by the user himself. The association rule method is also widely used in recommender systems. This method concentrates on

FIGURE 2: The recommendation performances on MovieLens; our method uses exponential constant parameters: $t = 100$, $\alpha = 0.5$, and $\beta = 0.1$.

the latent relationship between items. To find these relationships, every user's item list is analyzed to create a list of the most related items called association rule. Heat spreading method, a variant of probabilistic spreading method, has the highest rate of coverage and diversity in current recommendation algorithms, but it ignores accuracy. In the experiment, we use accuracy of recommendation as the lower bound of precision and use its coverage and diversity rate as the upper bound. Therefore, we use the enhance metrics to evaluate the performance, as shown in (10a)–(10c). Heat spreading and probabilistic spreading are integrated methods which do well on precision but are not so good on coverage and diversity:

$$E\text{-Precision} = \frac{P}{P_{\text{HeatS}}}, \quad (10a)$$

$$E\text{-Coverage} = \frac{\text{Cov}}{\text{Cov}_{\text{HeatS}}}, \quad (10b)$$

$$E\text{-Diversity} = \frac{\text{Div}}{\text{Div}_{\text{HeatS}}}. \quad (10c)$$

To evaluate the performance of the approximate model, missing rate M and comprehensive Comp are defined in the metric, as shown in (11). Here, M_* presents the missing rate of performance such as precision, coverage, and diversity; PN_* presents the performances of normal model and PA_* is the approximate performance of the model; per denotes the percentage of approximate data in the entire data set; Δ is the controlling parameter that accommodates the floating of M ; a , b , and c are the weight of metrics (precision, coverage, and diversity) for comprehensive rate:

$$M_* = \frac{PN_* - PA_*}{PN_*},$$

FIGURE 3: The recommendation performances on Netflix; our method uses experiential constant parameters: $t = 100$, $\alpha = 0.5$, and $\beta = 0.1$.

$$\text{Comp} = \frac{(a \times M_p + b \times M_{\text{Cov}} + c \times M_{\text{Div}}) \times (\text{per} + \Delta)}{\Delta}. \quad (11)$$

3.3. Results. The recommendation performances of different methods on the MovieLens, Netflix, and Last.fm data sets are shown in Figures 2, 3, and 4, respectively. The parameters in our method will affect the result. Here we use experiential constant parameters: $T = 100$, $\alpha = 0.5$, and $\beta = 0.1$.

The performances of our method are far better than those of the other methods on MovieLens data set which has the most links in the experiment, with ProbS running a close second, while both UserCF and ARule performed significantly worse. When the length of recommendation is lower than 20, the performance of our method is at least twice as well as the other two methods. On the Netflix data set, our

method consistently performs very well in terms of precision, coverage, and diversity. The precision of ProbS goes near to that of our method while its other performances are much worse. In addition, our method gets good comprehensive performance on Last.fm which is the sparsest data set in the experiment. When the recommendation list length is over 50, the precision of our method is lower than that of ARule, and coverage runs a close second. Furthermore, the consistency of its diversity could be rated as the best. The performances of the approximate model are shown in Figures 5 and 6.

Different metrics are drawn on different maps to show their tendency of coverage, as shown in Figure 5. For precision, with the increasing of data size, the missing rate declines more slowly and is leveled out in the end. The missing rate is controlled between 0 and 20%. The tendency of coverage is different from that of precision. The missing rate will increase,

FIGURE 4: The recommendation performances on Last.fm; our method uses exponential constant parameters: $t = 100$, $\alpha = 0.5$, and $\beta = 0.1$.

at first, while the transition occurs in the range of 5 to 10 (percentage). On the contrary, the missing rate of diversity is very high in the heat of the line. The lowest point is located in the range of 5 to 10 as well, and the missing rate is leveled out at the tail.

Figure 6 shows the comprehensive performances of approximate model. Different comprehensive curve will be drawn depending on the parameters, and we kept the parameters fixed as $\Delta = 10$, $a = 0.50$, $b = 0.15$, and $c = 0.35$ in the experiment. There are some differences between the two data sets on their comprehensive curve. The comprehensive curve of MovieLens data set is flat, and less lower values lay on the heat of the curve while slight oscillation comes on the tail. In contrast, the comprehensive curve of Netflix data set is like a hook. Considering the above-mentioned factors, the optimal value occurs near 10 (percentage).

4. Conclusions

In this paper, we proposed a method which makes use of users' behaviors to give recommendation. Instead of modeling with tags or contexts, our method takes the collecting lists to construct a recommendation model without the contents of items. As shown in the experiment, our method exhibits an all-round competitive performance on precision, coverage, and diversity, in comparison with four typical classes of recommendation algorithms. To reduce the computing complexity of our method, approximate model is also proposed in this paper, where the adjusting parameters are the determinant of performance curve of approximate model. As shown in the experiment, the approximate method is feasible since the optimal value is under 20%. When precision is considered to be the most important metric, it is

FIGURE 5: The performances of approximate model.

more appropriate to use less than 10% of the data to construct the approximate model.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61300018 and 61103109), Research Fund for the Doctoral Program of Higher Education of China (no. 20120185120017), China Postdoctoral Science Foundation (nos. 2013M531951 and 2014T70860),

FIGURE 6: The missing rates and comprehensive rates on MovieLens and Netflix in which $\Delta = 10$, $a = 0.50$, $b = 0.15$, and $c = 0.35$.

Fundamental Research Funds for the Central Universities (no. ZYGX2012J071), and Special Project of Sichuan Youth Science and Technology Innovation Research Team (no. 2013TD0006).

References

- [1] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," *Lecture Notes in Computer Science*, vol. 4321, pp. 291–324, 2007.
- [2] B. Sarwar, G. Karypis, and J. Konstan, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web (WWW '01)*, 2001, Item-Based Collaborative Filtering Recommendation Algorithms.
- [3] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The Adaptive Web*, vol. 4321 of *Lecture Notes in Computer Science*, pp. 325–341, Springer, 2007.
- [4] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: a constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.
- [5] S. Maslov and Y.-C. Zhang, "Extracting hidden information from knowledge networks," *Physical Review Letters*, vol. 87, no. 24, Article ID 248701, 2001.
- [6] P. Laureti, L. Moret, Y. Zhang, and Y. Yu, "Information filtering via iterative refinement," *Europhysics Letters*, vol. 75, no. 6, pp. 1006–1012, 2006.
- [7] J. Ren, T. Zhou, and Y.-C. Zhang, "Information filtering via self-consistent refinement," *Europhysics Letters*, vol. 82, Article ID 58007, 2008.
- [8] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data (SIGMOD '93)*, vol. 22, no. 2, pp. 207–216.
- [9] W. Lin, S. A. Alvarez, and C. Ruiz, "Efficient adaptive-support association rule mining for recommender systems," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 83–105, 2002.
- [10] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4–5, pp. 993–1022, 2003.
- [11] T. L. Griffiths and M. Steyvers, "Finding scientific topic," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 1, 2004.
- [12] D. M. Blei and J. D. Lafferty, "Correlated topicmodels," in *Advances in Neural Information Processing Systems*, vol. 18, 2005.
- [13] F.-F. Li and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 2, pp. 524–531, June 2005.
- [14] X. Wei and W. B. Croft, "LDA-based document models for ad-hoc retrieval," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '06)*, pp. 178–185, August 2006.
- [15] Z. Wang and X. Qian, "Text categorization based on LDA and SVM," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE '08)*, vol. 1, pp. 674–677, December 2008.
- [16] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia-Molina, "Clustering the tagged web," in *Proceeding of the 2nd ACM International Conference on Web Search and Data Mining (WSDM '09)*, pp. 54–63, New York, NY, USA, February 2009.
- [17] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," *Lecture Notes in Computer Science*, vol. 7301, no. 1, pp. 431–443, 2012.
- [18] R. Krestel, P. Fankhauser, and W. Nejdl, "Latent Dirichlet allocation for tag recommendation," in *Proceedings of the 3rd ACM Conference on Recommender Systems*, pp. 61–68, October 2009.
- [19] Y. Song, L. Zhang, and C. L. Giles, "Automatic tag recommendation algorithms for social recommender systems," *ACM Transactions on the Web*, vol. 5, article 4, no. 1, 2011.
- [20] X. Si and M. Sun, "Tag-LDA for scalable real-time tag recommendation," *Journal of Computational Information Systems*, vol. 6, no. 1, pp. 23–31, 2009.

- [21] J.-H. Kang, K. Lerman, and L. Getoor, “LA-LDA: a limited attention topic model for social recommendation,” in *Social Computing, Behavioral-Cultural Modeling and Prediction*, vol. 7812 of *Lecture Notes in Computer Science*, pp. 211–220, Springer, Berlin, Germany, 2013.
- [22] T. Hofmann, “Unsupervised learning by probabilistic Latent Semantic Analysis,” *Machine Learning*, vol. 42, no. 1-2, pp. 177–196, 2001.