City Recommender System based on a Latent Topic Model

Thai-Binh NGUYEN[†], Kenro AIHARA[††,†††], and Atsuhiro TAKASU[††,†††]

† SOKENDAI (The Graduate University for Advanced Studies) Shonan Village, Hayama, Kanagawa 240-0193 Japan

†† National Institute of Informatics 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan

E-mail: †{binh,kenro.aihara,takasu}@nii.ac.jp

Abstract Travel recommendation is a very challenging for the application of recommender systems. A travel recommender system helps travelers in decision making processes when they are planning a trip, including the choice of destinations. In this research, our goal is to build algorithm for recommending venues for a user when he visits a city. One of the challenges is the recommendation of venues in a city that the user has never been (no history activities in that city - the cold-start problem). We cope with this challenge by exploiting the location information of users ' history activities as well as users ' behavior when they visit a city. Our experiment is conducted on the dataset of FourSquare, a location-based social network.

Key words Tourism Recommender System, Latent Topic Model, Location-based recommender system

1. Introduction

With the increase in Local-based Social network (LBSN), e.g., FourSquare (foursquare.com), DoubanEvent (douban.com), etc., and the emerging of mobile devices with GPS, travelers can "check-in", share information about the places they have visited, e.g., the venues, the cities, etc. A travel recommender system discovers users' travel patterns by analyzing such history activities, then can help travelers to decide where they should visit in a city. This problem can be formulated as follow (top-k recommendation problem):

Input: a querying user u, a city l

Output: a list of k venues inside l that may interest u.

A majority of algorithms in recommender systems, e.g., collaborative filtering, matrix factorization, etc., are based on analyzing the user-item matrix. However, in travel data, the user-item matrix is often sparse, leading to the low accuracy of these methods (see section 2).

The goal of this research is to propose a method that can cope with the problem of sparseness of the user-item matrix.

Our approach in building a travel recommender system is that we try to capture how travelers decide where to visit in real life. Then we try to find a model that can reflect that decision making process. In other words, our goal is to create a model that fits the decision making process with the history activities of travelers.

The main contribution of this paper is two-fold. First, we defined a new decision making process when a user decides which venues to visit in a city. The goal of this decision making process is to exploit location information of the venues, in order to alleviate the sparseness of the user-item matrix. This is an extension of the decision making process proposed in [1]. We will go into detail of this decision making process later in this section.

Second, we proposed a probabilistic topic model based travel recommender system (LocLDA) that incorporates this decision making process.

The idea of the decision making we mentioned above is due to the fact that when we where to visit in a city, we often refer to (i) Our own personal interests, (ii) Which venues in the city that people living in the city prefer, and (iii) Which venues in the city that people from outside of the city prefer when they travel to the city.

More concretely, the three factors, that are used in recommending, are summarized as following:

Personal interest: It shows which kinds of venues interest the users. Some people like to visit traditional construction buildings, some people like to go local bars, etc.

Local people preference: It shows which kinds of venues that local people of the city like to visit, i.e., which kinds of venues in the city is popular among local people

Non-local people preference: It shows which kinds of venues that people from outside of the city like to visit when they travel to the city, i.e., which kinds of venues in the city is popular among non-local people.

In our model, we represent *personal interest* for each user, *local people preference* and *non-local people preference* for each city using latent topic model. These presentations are learned from the activities in the past of users. This model is an extension of original Latent Dirichlet Allocation (LDA) [2] proposed by Blei et al. We also model the influence of each factor to each individual. For example, some users trend to prefer their own interests, while some trend to prefer the target city's local people preference, some mix three factors but with different weights.

LocLDA is able to link users similar taste (via personal interest) as well as exploit the popularity of each city in making recommendation (via local people preference and non-local people preference), hence it can cope with the problem of sparseness of user-item matrix.

2. Related Works

Traditional recommender system use collaborative filtering [3]–[6] or matrix factorization method [7] in order to make recommendation. Even though these methods have achieved remarkable accuracy in recommender systems, in the travel recommender systems, they are suffering from the problem of the sparseness of user-item matrix. In the travel recommender systems, the sparseness of the user-item matrix is due to the following two observations that have been pointed out in [8]: (1) Most users can only visit a limited number of venues (2) Most users travel within a limited distance from the cities they are living. These two observations imply that the number of users, who have visited city l, and have same travel patterns with the querying user u, is not sufficient so that collaborative filtering and matrix factorization can perform well.

Levandoski et al. [8] proposed a location-aware method for recommender system. Their method is based on the assumption that: user living in the same region likely to have the same taste. In recommending to a user u, they applied collaborative filtering technique that utilizes only ratings of users living in the same region with u. However, the reduction of number of users in collaborative filtering makes the user-item matrix more sparse.

Yin et al. [1] coped with the above challenges by proposing a method that consider both user's personal preference and the popularities of venues in cities. In other words, the recommendation list of venues in city l for user u is generated based on two factors: (1) User u's personal interest, and (2) The popularities of venues in city l. Although this research can reflect both user interest as well as local preference of each city, we can divide the local preference to two part: (i) local preference among local people, and (2) local preference among non-local people, in order to further model the travelers' decision making process more detail.

3. Latent Topic Model based Recommender System

3.1 Preliminary

Notations Notations used in the model are defined in the table 1.

Table 1 Some notations

$\mathcal{U}$,	set of users
$\mathcal{V}$,	set of venues
$\mathcal{L}$,	set of cities
u	user id
v	venue id
l	city id
N	number of users
V	number of venues
L	number of cities
D_u	number of activities of user u
K	number of topics
θ_u	per user-topic distribution of user u
$\theta_l^{'}$	per city-topic distribution of over local people of city l
$\theta_l^{''}$	per city-topic distribution of over non-local people of city l
ϕ_z	per topic-venue distribution of topic z
λ_u	3-dimensional vector, which its elements sum up to 1, showing how *personal interest*, *local people preference*, *non-local people preference* influence to the decision of user u
n	number of observed user activities in the dataset
$\mathbf{v}$	set of observed venues in data set $\mathbf{v} = \{v_1, v_2, ..., v_n\}$
$\mathbf{s}$	set of assigned values for switched variables $\mathbf{s} = \{s_1, s_2, ..., s_n\}$
$\mathbf{z}$	set of assigned values for topic variables $\mathbf{z} = \{z_1, z_2, ..., z_n\}$
$\alpha, \alpha^{'}, \alpha^{''}$	Dirichlet priors to distributions $\theta, \theta^{'}, \theta^{''}$, respectively
β	Dirichlet prior to distribution ϕ
γ	Multinomial prior to distribution λ

User activity: each user activity is a check-in, which is represented by a tuple (u, v, l_u, l_v) . Where $u \in \mathcal{U}$ is the user id, $v \in \mathcal{V}$ is the venue id, $l_u \in \mathcal{L}$ is the id of home city of user, $l_v \in \mathcal{L}$ is the city of venue.

User profile: User profile of user u is the set of user activities, i.e., the tubles $(u, v, l_u, l_v))$ corresponding with user u

Topic: A topic z is a distribution over venues. $P(v|z) = \phi_{z,v}$, $(z = 1...K, v = 1...V)$

A topic distribution can be represented by vector ϕ_z as follow:

$$\phi_z = (\phi_{z,1}, \phi_{z,2}, ..., \phi_{z,V})$$

Intuitively, the distribution of topic z shows how important each venue is in the topic.

Personal interest: Personal interest of a user u is denoted by θ_u, showing personal reference of u. It is a distribution over topics.

$$\theta_u = (\theta_{u,1}, \theta_{u,2}, ..., \theta_{u,K})$$

Local people preference: The local people preference of a city l is denoted by $\theta_l^{'}$. It shows the preference of local people to venues inside l and is a distribution over topics.

$$\theta_l^{'} = \left(\theta_{l,1}^{'}, \theta_{l,2}^{'}, ..., \theta_{l,K}^{'}\right)$$

Non-local people preference: The non-local preference of a city l is denoted by $\theta_l^{''}$. It shows the preference of non-local people when they visit city l and is a distribution over topics.

$$\theta_l^{''} = \left(\theta_{l,1}^{''}, \theta_{l,2}^{''}, ..., \theta_{l,K}^{''}\right)$$

3.2 LocLDA model

According to the assumption of the decision process that user u may choose the venues to visit based on a mixture of her personal interest, local people preference and non-local preference of the city, the probability that user u prefers venue v is calculated by Equation 1.

$$\begin{aligned} P(v|u, l, \phi) = & \lambda_{u,1} P(v|\theta_u, \phi) + \\ & \lambda_{u,2} P\left(v|\theta_l^{'}, \phi\right) + \\ & \lambda_{u,3} P\left(v|\theta_l^{''}, \phi\right) \end{aligned} \tag{1}$$

Where

- $\lambda_{u,1}, \lambda_{u,2}, \lambda_{u,3}$: mixing weights that show how the decision of u is influenced by her personal interest, local people preference and non-local preference. These mixing weights sum up to 1 and vary by users. They reflect the user 's characteristic
- $P(v|\theta_u, \phi)$: the probability that u prefers v according to her personal interest
- $P\left(v|\theta_l^{'}, \phi\right)$: the probability that u prefers v according to the local people preference
- $P\left(v|\theta_l^{''}, \phi\right)$: the probability that u prefers v according to the non-local user preference

$P(v|\theta_u, \phi), P\left(v|\theta_l^{'}, \phi\right), P\left(v|\theta_l^{''}, \phi\right)$ are calculated by Equation 2.

$$\begin{aligned} P(v|\theta_u, \phi) &= \sum_z P(v|z, \phi) P(z|\theta_u) \\ P\left(v|\theta_l^{'}, \phi\right) &= \sum_z P(v|z, \phi) P\left(z|\theta_l^{'}\right) \\ P\left(v|\theta_l^{''}, \phi\right) &= \sum_z P(v|z, \phi) P\left(z|\theta_l^{''}\right) \end{aligned} \tag{2}$$

That decision process is represented by the following generative process in Algorithm 1.

```
for each activity $(u, v, l_u, l_v)$ of user $u$ do
    Draw $s \sim Categorical(\lambda_u)$;
    if $(s == 1)$ then
        Draw $z \sim Multinomial(\theta_u)$;
    else if $(s == 2)$ then
        Draw $z \sim Multinomial\left(\theta^{'}_{l_v}\right)$;
    else
        Draw $z \sim Multinomial\left(\theta^{''}_{l_v}\right)$;
    end
    Draw $v \sim Multinomial(\phi_z)$;
end
```

Algorithm 1: Generative process

where the distributions of the parameters are given as in Equation 3.

$$\begin{array}{lll} \theta & \propto & Dirichlet(\alpha) \\ \theta^{'} & \propto & Dirichlet(\alpha^{'}) \\ \theta^{''} & \propto & Dirichlet(\alpha^{''}) \\ \phi & \propto & Multinomial(\beta) \\ \lambda & \propto & Dirichlet(\gamma) \end{array} \tag{3}$$

Here, $\alpha, \alpha^{'}, \alpha^{''}, \beta, \gamma$ are hyper parameters, being used as priors for the parameters. They specify the nature of the parameters. Although these hyper parameters are vector-valued, for simplifying, we assume they are symmetric. In other words, elements of each hyper parameters have same value. Hereafter, we use each of $\alpha, \alpha^{'}, \alpha^{''}, \beta, \gamma$ to refer to both the vector as well as its element's value. Which is referred depends on the context.

The graphical representation of the model is described in Figure 4.

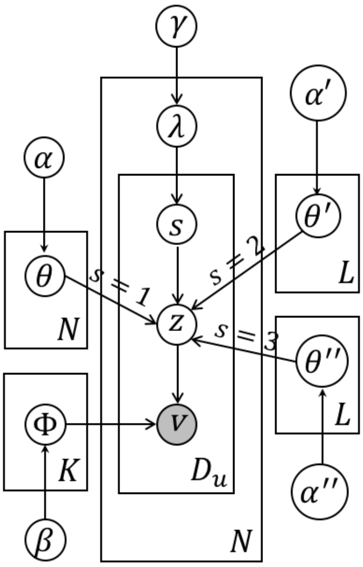

Fig. 1 LocLDA model

3.3 Model inference by collapsed Gibbs sampling method

Model inference is to infer the latent variables, i.e. **z**, **s**. First, we represent the joint distribution of the model is calculated via the formulas in Equation 4.

$$\begin{aligned} & P\left(\mathbf{v}, \mathbf{z}, \mathbf{s} | \alpha, \alpha^{'}, \alpha^{''}, \beta, \gamma\right) \\ & = \int \cdots \int P(\mathbf{v}|\phi, \mathbf{z}) P(\phi|\beta) \\ & \times P(\mathbf{z}|\theta, \theta^{'}, \theta^{''}, \mathbf{s}) P(\theta|\alpha) P(\theta^{'}|\alpha^{'}) \\ & \times P(\theta^{''}|\alpha^{''}) P(\mathbf{s}|\lambda) P(\lambda|\gamma) d\phi d\theta d\theta^{'} d\theta^{''} d\lambda \end{aligned} \tag{4}$$

The joint distribution in Equation 4 is intractable. Therefore, we use the collapsed Gibbs sampling method [9] to approximate the hidden variables assignment for the training data and to estimate the parameters $\left(\theta, \theta^{'}, \theta^{''}, \phi, \lambda\right)$.

Slightly different from Gibbs sampling procedure in [9], our model requires two-fold one: one fold to sample s and one fold to sample z.

In order to apply collapsed Gibbs sampling method as described above, we have to calculate $P\left(s_i | \mathbf{s}^{-i}, \mathbf{z}, \mathbf{v}, .\right)$ and $P\left(z_i | \mathbf{z}^{-i}, \mathbf{s}, \mathbf{v}, .\right)$. By doing the same procedure as described in [9], these probabilities can be finally represented as in Equation 5 and Equation 6.

Sample s

$$p(s_i | \mathbf{s}_{-i}, \mathbf{z}, \mathbf{v}, \mathbf{.}) \quad \propto \quad A \times B \tag{5}$$

where,

$A = \frac{n^{-i}_{u_i, s_i} + \gamma}{\sum_s (n^{-i}_{u_i, s} + \gamma)}$, and,

$$B = \begin{cases} \frac{n^{-i}_{u_i, z_i} + \alpha}{\sum_z (n^{-i}_{u_i, z} + \alpha)} & \text{if } s_i = 1 \\ \frac{n^{'-i}_{l_i, z_i} + \alpha^{'}}{\sum_z (n^{'-i}_{l_i, z} + \alpha^{'})} & \text{if } s_i = 2 \\ \frac{n^{''-i}_{l_i, z_i} + \alpha^{''}}{\sum_z (n^{''-i}_{l_i, z} + \alpha^{''})} & \text{if } s_i = 3 \end{cases}$$

Sample z

$$p(z_i | \mathbf{z}_{-i}, \mathbf{s}, \mathbf{v}, \mathbf{.}) \quad \propto \quad C \times D \tag{6}$$

where,

$C = \frac{n^{-i}_{z_i, v_i} + \beta}{\sum_{v \in V} (n^{-i}_{z_i, v} + \beta)}$, and,

$$D = \begin{cases} \frac{n^{-i}_{u_i, z_i} + \alpha}{\sum_z (n^{-i}_{u_i, z} + \alpha)} & \text{if } s_i = 1 \\ \frac{n^{'-i}_{l_i, z_i} + \alpha^{'}}{\sum_z (n^{'-i}_{l_i, z} + \alpha^{'})} & \text{if } s_i = 2 \\ \frac{n^{''-i}_{l_i, z_i} + \alpha^{''}}{\sum_z (n^{''-i}_{l_i, z} + \alpha^{''})} & \text{if } s_i = 3 \end{cases}$$

Here,

$n_{u,z}$: the number of times a history activity of user u has been assigned to topic z

$n^{'}_{l,z}$: the number of times a history activity at city l of local people has been assigned to topic z

$n^{''}_{l,z}$: the number of times a history activity at city l of non-local people has been assigned to topic z

$n_{u,s}$: the number of times a history activity at city l of user u has been assigned to switch variable s

$n_{z,v}$: number of times venue v has been assigned to topic z

Superscript $^{-i}$ or subscript $_{-i}$ is a count that does not include the current assignment of s_i, z_i

The s_i variables are initialized randomly to values in $\{1, 2, 3\}$ and z_i variables are initialized randomly to values in $\{1, 2, ..., K\}$.

3.4 Parameter estimation

After a sufficient number of iterations, the posterior distributions converge to the target distribution. And the parameters can be computed as in Equation 7.

$$
\begin{aligned}
\theta_{u,z} &= \frac{n_{u,z} + \alpha}{\sum\limits_{z}(n_{u,z} + \alpha)}, & u \in \mathcal{U}, z \in \{1, ..., K\} \\
\theta^{'}_{l,z} &= \frac{n^{'}_{l,z} + \alpha^{'}}{\sum\limits_{z}(n'_{l,z} + \alpha')}, & l \in \mathcal{L}, z \in \{1, ..., K\} \\
\theta^{''}_{l,z} &= \frac{n^{''}_{l,z} + \alpha^{''}}{\sum\limits_{z}(n''_{l,z} + \alpha'')}, & l \in \mathcal{L}, z \in \{1, ..., K\} \\
\phi_{z,v} &= \frac{n_{z,v} + \beta}{\sum\limits_{v}(n_{z,v} + \beta)}, & z \in \{1, ..., K\}, v \in \mathcal{V} \\
\lambda_{u,j} &= \frac{n_{u,j} + \gamma}{\sum\limits_{j^{'}}(n_{u,j'} + \gamma)}, & j, j^{'} \in \{1, 2, 3\}
\end{aligned}
\tag{7}
$$

3.5 Ranking score

The ranking scores show how users may interest venues. They are used as the criterion to form the top-k venues, in a city, that might interest a user. They are computed by $\theta, \theta^{'}, \theta^{''}, \phi, \lambda$ which are learned in the model.

The ranking score of user u to location v in city l is calculated as in equation 8.

$$S(u, v, l) = \sum_{z} T(u, l, z)\phi_{z,v} \tag{8}$$

Where, $T(u, l, z)$ shows how user u prefers topic z in city l, and is calculated by equation 9.

$$T(u, l, z) = \lambda_{u,1}\theta_{u,z} + \lambda_{u,2}\theta^{'}_{l,z} + \lambda_{u,3}\theta^{''}_{l,z} \tag{9}$$

In the equation 8, the $T(u, l, z)$ is calculated each time a recommendation task is required. While, the $\phi_{z,v}$ is calculated offline in advance, in the model training phase.

4. Experiments

4.1 Goal of the experiment

Goal of the experiment is to evaluate the accuracy of the recommendation lists in recommending venues of the city that is new with the target user.

4.2 Dataset

4.2.1 Dataset introduction

In this paper, we use data from FourSquare, a Location-based Social Network (LBSN), for the performance evaluation. FourSquare allows users to mark ”check-in” at places they visited (e.g., restaurants, museums). The resident locations of users as well as the location of the venues are also included in the dataset by mean of $(latitude, longitude)$ pairs.

4.2.2 Data preprocessing

We used GeoHash reversion algorithm to retrieve city name and nation name from $(latitude, longitude)$. Each city and each nation is assign an id.

In other words, after the preprocessing process, we get dataset which is a set of tuples (u, v, l_u, l_v)

4.3 Data splitting

The dataset is splitted into two parts: *test data* and *train data*. Because the goal of the experiment is to evaluate the accuracy of the algorithm in recommending venues at a city that a user has never been, the way we splitted data is described as in the table 2.

Table 2 Data splitting

Test data	All venues visited by each user in a non-home city
Train data	The rest of each user 's visited venues in other cities

4.4 Evaluation method

We adopt the *Recall*@k metric [10]. For each test case in S_{test} (S_{test}: set of all test cases):

(1) Randomly select 50 additional venues located in that user has not visited
(2) Calculate the ranking scores 51 venues including test item v and the above 50 venues
(3) Form a ranked list of the 51 venues by ordering there ranking scores descending. Let p denote the rank of v in the list
(4) Create a top-k recommendation list by picking the top-k ranked venues from the list formed in the previous step. If $(p < k)$ we have a hit (i.e., the test venue v appears in the top-k list that will be recommended to the user). Otherwise, we have a miss.

The recall@k metric is calculated as the following formula:

$$Recall@k = \frac{hit@k}{|S_{test}|} \tag{10}$$

Where is the number of hit over the test set, is the size of the test set.

4.5 Experimental settings

Values of hyper parameters used in the experiment are fixed as in the table 3.

Table 3 Hyper parameters settings

K (number of topics)	20
$\alpha, \alpha^{'}, \alpha^{''}$	$50/K$
β	0.01
γ	0.33

4.6 Experimental results

The recall@k is calculated for k run from 1 to 20.

We compared our method with previous methods:

LDA: in this model, each user is considered as a document, each venue that the user has visited is considered as a word of the document. The parameters for LDA is set as follow: $\alpha = 50/K$, $\beta = 0.01$,

The ranking score in LDA based method is calculated by equation 11.

$$S_{\text{LDA}}(u, v, l) = \sum_{z} \theta_{u,z}\phi_{z,v} \tag{11}$$

Where, $\theta_{u,z}$ shows the personal interest of user u (per user topic distribution), and $\phi_{z,v}$ shows the per topic venue distribution.

LA-LDA [1]: parameters for LA-LDA are set as follow: $\alpha = \alpha\prime = 50/K$, $\beta = 0.01$, $\gamma = 0.5$. The ranking score in LA-LDA based method is calculated by equation 12.

$$S_{\text{LA-LDA}}(u,v,l) = \sum_{z} T_{\text{LA-LDA}}(u,l,z)\phi_{z,v} \quad (12)$$

Where, $T_{\text{LA-LDA}}(u,l,z)$ shows how user u prefers topic z in city l, and is calculated by equation 13 (refer [1]).

$$T_{\text{LA-LDA}}(u,l,z) = \lambda_u \theta_{u,z} + (1-\lambda_u)\theta^{'}_{l,z} \quad (13)$$

where $\theta_{u,z}$ shows how user u prefers topic z, and $\theta^{'}_{l,z}$ shows the preference of topic z in city l (including both local people and non-local people).

Here, the parameter λ_u is use to represent how use u is influenced by her personal interest and the target city's local preference when u decides where to visit in city l.

Figure 2 and Figure 3 show the *recall@k* when we evaluate users who have visited at least 10 and 15 venues respectively.

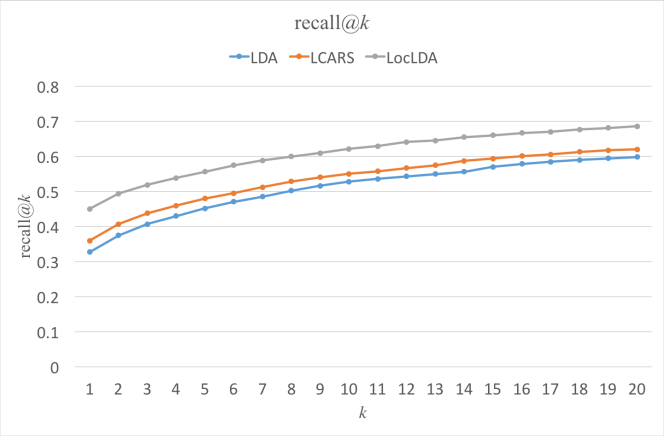

Fig. 2 Recall@k for users who have visited at least 10 venues

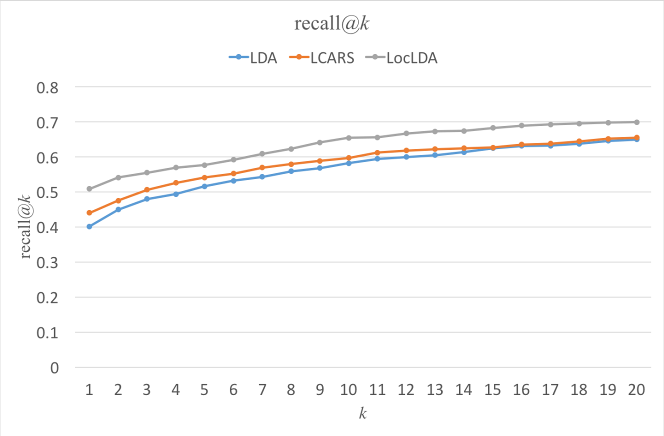

Fig. 3 Recall@k for users who have visited at least 15 venues

Graphs in figures 2 and 3 show that the accuracy of LocLDA super-passes that of both LA-LDA and LDA. It means that when the accuracy trends to rise if the amount of data is sufficient enough. When we increase the number of visit per user, the *recall@k* rises significantly for small k.

5. Conclusion and Future Work

We proposed a new decision process when travelers choose venues to visit and proposed a latent topic model based travel recommender system. The experimental results show that the accuracy of this model is better than previous methods when recommending venues in cities that is new for users. However, this research has not still care about the ratings given by users to venues. In the future, we want to incorporate these ratings into the model in order to increase the performance.

References

[1] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "Lcars: A location-content-aware recommender system," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, New York, NY, USA, pp.221–229, ACM, 2013.

[2] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent dirichlet allocation," J. Mach. Learn. Res., vol.3, pp.993–1022, March 2003.

[3] M.D. Ekstrand, J.T. Riedl, and J.A. Konstan, "Collaborative filtering recommender systems," Found. Trends Hum.-Comput. Interact., vol.4, no.2, pp.81–173, Feb. 2011.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," IEEE Internet Computing, vol.7, no.1, pp.76–80, Jan. 2003.

[5] J. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in The Adaptive Web, Lecture Notes in Computer Science, vol.4321, pp.291–324, Springer Berlin Heidelberg, 2007.

[6] F. Ricci, L. Rokach, B. Shapira, and P.B. Kantor, Recommender Systems Handbook, 1st ed., Springer-Verlag New York, Inc., New York, NY, USA, 2010.

[7] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," Computer, vol.42, no.8, pp.30–37, Aug. 2009.

[8] J.J. Levandoski, M. Sarwat, A. Eldawy, and M.F. Mokbel, "Lars: A location-aware recommender system," Proceedings of the 2012 IEEE 28th International Conference on Data Engineering, ICDE '12, Washington, DC, USA, pp.450–461, IEEE Computer Society, 2012.

[9] T.L. Griffiths and M. Steyvers, "Finding scientific topics," Proceedings of the National Academy of Sciences of the United States of America, vol.101, no.Suppl 1, pp.5228–5235, April 2004.

[10] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, New York, NY, USA, pp.39–46, ACM, 2010.