Temporal Recommender System in online Tourism Website

Qian-Ming Zhang, Xing Yu, and Jingqiu Wang

***Abstract*—Plenty of recommendation methods are put forward to alleviate the information overload problem, but no unique method can satisfy all the e-commerce web sites. One significant reason lies in the different features of data sets, for instance the sparsity of the customer-object relations which would lead to cold start problem. In this study, we obtain a data set from an online tourism website, where the purchasing behaviors are too scarce to support recommender systems well. However, there are abundant of browsing histories which are very related to customers' purchasing behaviors. We then propose a new temporal recommendation method based on the temporal browsing histories. Compared with some benchmark methods, such as collaborative filtering method, popularity-based method, and content-based method, our new method can provide very accurate and novel recommendations. It can also update the recommendation lists in real time with little computational cost. In addition, we infer that customers usually have clear targets when they enter this web site.**

***Index Terms*—Recommender System, Tourism Routes, Data Mining, Temporal, Browsing History**

I. Introduction

RECOMMENDER systems are a subset of information filtering system that provide customers with relevant objects [1], [2]. They are applied in plenty of e-commerce web sites to boost the sales, provide convenient shopping experience, and enhance the customers' loyalty. To date, many different models have been proposed [2], such as content-based method [3], collaborative filtering method [4], [5], hybrid recommender systems [6], [7], diffusion-based method [8], [9] and so on.

Previous related studies usually focus on the data limited to the purchasing or rating histories. However, such records for single customer is relatively scarce in many cases. For instance, we get some purchasing histories from a web site which provides various plans of tourist routes, and find customers only has 1.1 purchased plan on average due to the limited budget. If we only consider the purchasing behavior to build recommendation system, it would probably fall into cold start problem [10]. In order to overcome this disadvantage, recommending popular objects is a good and widely used solution. However, we find it performs badly, because none of the objects is very popular, and the customers' purchased plans are revealed to be quite different. The essential reason is still due to the very few purchasing histories.

Actually, there are indeed much information we can use. In this study, we successfully obtain an amount of browsing histories, which were little studied before. Replacing customer-purchase-object relations with customer-browse-object relations, we firstly implement several classical recommendation method, including popularity-based method, collaborative filtering method, and content-based method. None of them performs well. Nevertheless, we notice that different customers have neither similar purchasing histories, nor browsing histories, while the similarity among the objects within one customer's browsing history are much higher than those among all objects. We also find the customers prefer the objects which they recently browsed. Considering these features comprehensively, we propose a temporal recommender system based on the temporal browsing histories. It can not only provide more accurate and novel recommendations, but also update the recommendation lists in real time with little computational cost. That is to say, the recommendation lists can be adjusted immediately once the customer browses new objects, which might impact customers' decisions.

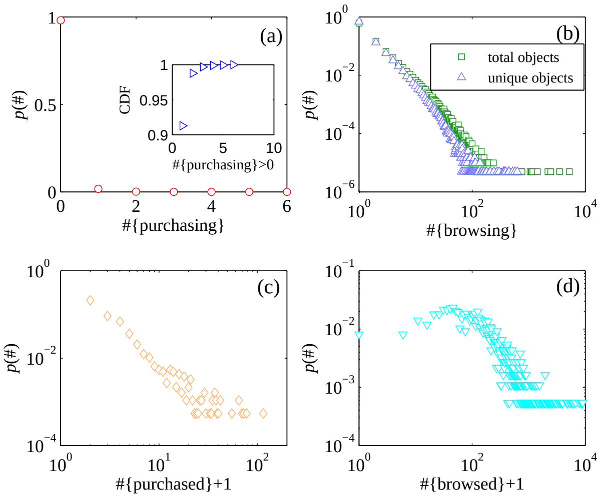

Fig. 1. Distribution of customer's activities and objects' popularities. Subfigure (a) shows the distribution of the amount of objects purchased by per customer, and the embedded figure correspondingly shows the cumulative distribution where only the amounts above 1 are considered. Subfigure (b) shows the distribution of the amount of total/unique objects browsed by per customer. In Subfigure (c), the popularity is measured by the number of purchase. In Subfigure (d), the popularity is measured according to the browsing histories).

The rest of this paper is organized as follows: Section II briefly describe the related works; Section III presents the Data, the benchmark recommendation methods, and the evaluation metrics; we compare the experimental results and propose our new method in Section IV; Section V shows the conclusion and discussion.

Manuscript received December 23, 2015; revised January 15, 2016. QMZ thanks the support from the Program of Outstanding PhD Candidate in Academic Research by University of Electronic Science and Technology of China (No. YBXSZC20131034) and China Scholarship Council (No. 201306070064).

Q.-M. Zhang is with the CompleX Lab, Web Science Center, University of Electronic Science and Technology of China, Chengdu, Sichuan, 611731, P. R. China. Corresponding author e-mail: qmzhangpa@gmail.com.

X. Yu and J. Wang are with the Tours4Fun in Chengdu, Sichuan, 610041, P. R. China. E-mail: star.yu@toursforfun.cn, lucy.wang@toursforfun.com

Q.-M. Zhang and X. Yu contributed equally to this work.

II. Related Works

Considering the features of our data, we apply one content-based recommendation method [3] and one memory-based collaborative filtering recommendation method [11] as two benchmark methods. The crucial part of the content-based method is to characterize the items and then measure the similarity between them. The mainstream method contains Vector Space Model such as Term Frequency Inverse Document Frequency [12], and Probabilistic models such as Naïve Bayes Classifier [13] and Decision Trees [14]. As to the collaborative filtering methods, there are a mass of method measuring the similarities between customers [2], including Cosine index [5], Pearson coefficient [5], node-dependent similarity [15], path-dependent similarity [2] and so on.

Browsing histories were studied by Lieberman [16] early in 1995, in order to assist users in browsing the Internet, and still attract researchers' much attention to learn users' behaviors in the area of web personalization [17], [18]. Browsing histories are well utilized in web mining. However they are rarely studied in the recommendations for e-commerce web sites, because such information is very confidential. One related study is implemented in Amazon, where the recommender system records the users' browsing histories and purchasing histories to create a personalized shopping experience for each customer [19].

As to the time information, many temporal recommender systems are proposed [20]. Ding and Li [21] applied an exponential decay rate on old ratings to give more importance to recent ratings. They argued that the recent ratings should better reflect the users' current tastes. More evidence was found by Zhang et. al. [22], where they removed some old ratings and extracted the information backbone of online systems. In 2010, Koren [23] presented a model tracking the time changing behaviors in order to distill the longer-term trends from the noisy patterns. Xiong et. al. [24] proposed a continuous time-based model based on Bayesian probabilistic tensor factorization. And then, Guo et. al. discussed the impact of the time window on the recommender systems [25]. In addition, the influences of time dimension, such as the time of the day, day of the week, and season of the year, were also proved to be a valuable input for improving recommendation quality [26], [27].

III. Data and Method

A. Data

The data set is collected from a web site which offers various plans of tourist routes (named as objects in this paper). Due to the confidentiality agreement, neither the information about the company's name nor any personal information about the customers is public to us. Our data set is only consisted of the browsing and purchasing histories of anonymous customers with the time stamps from May to August in 2015. After removing the invalid accounts, including the testing accounts and proxy accounts, we get $201,466$ customers and $1,844$ objects. Therein, $3,893$ customers placed orders, and purchased $4,297$ objects in total. The majority of customers did not purchase any objects, while 91.29% customers out of who placed orders purchased only 1 object, shown in Figure 1(a). It therefore seems like to be very difficult to learn the customers' interests.

However, we find many customers have an abundant of browsing histories as shown in Figure 1(b), which also shows that a customer might browse an object many times. Then we check the frequencies of objects in each customer's history. As shown in 2(a), a customer will browse the purchased object 3.76 times on average, much higher than the number of times, 2.4, averaged by all the objects he browsed. When we consider all the customers, the average number of times is only 1.29. Many customers did not focus their attention on any specific object, which maybe because they did not have enough desire to travel, or just were surfing in the Internet. Therefore the pattern of browsing one object multiple times may be the significant feature of the customer who are eager to travel. And the object which was browsed multiple times are very probable to be the final purchased object (the evidence will be shown in the Section Experiments).

In Figure 1 (c) and (d), we respectively represent the distributions of the frequencies that each object was purchased and browsed. Again we can see the small number of customers' purchasing behaviors: nearly half of the objects were not purchased by anyone; even the most popular object was purchased by only 114 customers (less than 3% customers).

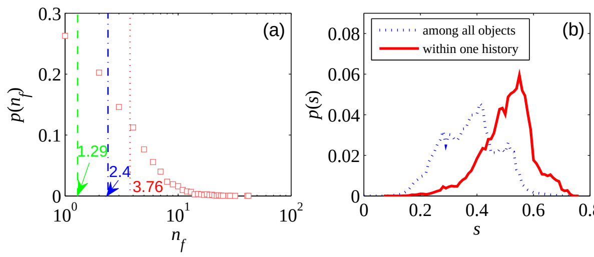

Fig. 2. The features of customers' browsing histories. Subfigure (a) represents the distribution of the frequency that a user browsed the object which he purchased. The decimal 3.76 is the average value of the frequencies. The decimal 2.4 is the average frequency of all the objects in each customer's (who purchased at least 1 object) history. The decimal 1.29 is also the average frequency, but for all the customers, no matter if he purchased any objects or not. Subfigure (b) shows the distribution of the similarities among objects. Blue dash curve represents the distribution of similarities among all the objects. Red solid curve corresponds to the similarities among the objects within one customer's browsing history.

B. Method

To compare the performances of different recommendation methods, only the customers who purchased objects are taken into consideration. We define the set of customers as C, and the set of objects as O. If a customer $c_u \in C$ browsed an object $o_i \in O$, we mark this relationship via $b_{ui} = 1$; else $b_{ui} = 0$. Similarly, if c_u purchased o_i, we denote it by $p_{ui} = 1$, else $p_{ui} = 0$. We introduce four benchmark recommendation methods.

Popularity based Algorithm (PbA for short). Popular items are favoured by many well performed recommender systems [2]. It can be defined as

$$s(o_i) = \sum_{c_u \in C} p_{ui}. \tag{1}$$

where $s(o_i)$ is the score of object o_i. The disadvantage is very obvious that every customer will be provided with the exactly same recommendation lists.

Collaborative Filtering based Algorithm (CFA for short). The term "collaborative filtering" was introduced

ISBN: 978-988-19253-8-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

by creators of the first commercial recommender system, Tapestry [28], and now has been successfully and widely applied. It filters information based on the common wisdom of multiple people, which means that people would like the object recommended by the people with similar interest. Here, we assume two customers are similar if they have similar browsing histories. The similarity can be defined based on Jaccard Index [29],

$$s(c_u, c_v) = \frac{\sum_{o_i} b_{ui} * b_{vi}}{\sum_{o_i} b_{ui} + \sum_{o_i} b_{vi} - \sum_{o_i} b_{ui} * b_{vi}}.$$

where $o_i \in O$. Then the score of recommending o_i to c_u can be calculated through

$$s(o_i|c_u) = \sum_{c_v \neq c_u} s(c_u, c_v) * b(v, i). \quad (2)$$

Similar Item based Algorithm (SIA for short). It is a content-based method. Here we create the objects' profiles using the cities contained in the tour routes, duration, quality (i.e., economy, normal and deluxe), and some characteristics (i.e., pick-up service, local tours, package tours, etc.). Therein, cities and characteristics are textual descriptions while the other two are numerical; the duration is continuous while the quantity is discrete. We therefore measure the similarity through four aspects: the similarity of the cities, the proximity of duration, the proximity of quality and the similarity of characteristic of different objects.

- Similarity of cities s_1. Different routes usually contain different numbers of cities, so the similarity between two routes with many cities would be underestimated by the number of common cities or Jaccard Index. For example, we have $s_1(o_i, o_j) = \frac{3}{6}$, $s_1(o_a, o_b) = \frac{4}{12}$ and $s_1(o_x, o_y) = \frac{9}{18}$, which are calculated via the Jaccard Index, where the numerators are the numbers of common cities. If we use the numbers of common cities, we get $s_1(o_x, o_y) > s_1(o_a, o_b) > s_1(o_i, o_j)$; if we use Jaccard Index, we have $s_1(o_i, o_j) = s_1(o_x, o_y) > s_1(o_a, o_b)$. But empirically such order $s_1(o_x, o_y) > s_1(o_i, o_j) > s_1(o_a, o_b)$ would be more rational, because two sets with large size are more difficult to share so many elements. So we introduce a new index $s_1(o_i, o_j) = \frac{n}{\sqrt{m}} * (\sqrt{m} - 1)$, where n is the number of common cities shared by o_i and o_j, m is the total number of cities in o_i and o_j.
- Proximity of duration s_2. It can be measured by the difference between two objects' durations, written as $s_2(o_i, o_j) = \frac{|d_i - d_j|}{|d_{\max} - d_{\min}|}$, where d_i is the duration of o_i, and $d_{\max}$ ($d_{\min}$) is the longest (shortest) duration. Smaller difference means higher similarity.
- Proximity of quality s_3. We define s_3 via a similar way to s_2, written as $s_3(o_i, o_j) = \frac{|q_i - q_j|}{2}$, where q_i represents the quantity of o_i. We use 1 to indicate economy tours, 2 to normal tours, and 3 to deluxe tours.
- Similarity of characteristic s_4. Because of the quite uneven distribution of different characteristics, we can not directly use the number of common characteristics to define s_4. We borrow the idea from Adamic and Adar [30], and get the definition $s_4(o_i, o_j) = \sum_{k=1}^{4} \frac{\delta(\mathrm{ch}_{ik}, \mathrm{ch}_{jk})}{\log N(\mathrm{ch}_{ik})}$, where ch_{ik} is the k-th characteristic of o_i, and $N(\mathrm{ch}_{ik})$ is the number of objects having this characteristic ch_{ik}.

Mixing the normalized s_1, s_2, s_3 and s_4 together, we obtain the similarity via

$$s = 0.4637s_1 + 0.1912s_2 + 0.0353s_3 + 0.3098s_4. \quad (3)$$

The parameters used here are determined by the stand score z-score, which is defined as $z = \frac{x-\mu}{\sigma}$, where x is raw value of one sample, μ is the mean value of the population and σ is the standard deviation of the population. The absolute value of z represents the distance between the raw value and the population mean in units of the standard deviation. The higher z is, the more significant x is. In this study, the z-scores of s_1, s_2, s_3, and s_4 are 1.5190, 0.6262, 0.1157 and 1.0146 respectively, which clearly show the significance of cities. Then we get the definition of s by normalizing z-score, shown as Eq. (3), and define the score of o_i as

$$s(o_i|c_u) = \sum_{o_j \in \{B(c_u)\}} s(o_i, o_j),$$

where $\{B(c_u)\}$ is the set of c_u's browsing history.

Intra-popularity based Algorithm (IPA for short). From Figure 2(a) we know that a customer tends to browse the object over and over again before he makes the final choice. Then we introduce a simple recommendation strategy that the objects which are browsed with higher frequency are assigned with higher score. The definition is written as

$$s(o_i|c_u) = \sum_{o_j \in B(c_u)} \delta(o_i, o_j), \quad (4)$$

where, $B(c_u)$ is the browsing history of customer c_u.

C. Evaluation Metrics

To evaluate the recommendation results, we firstly need to separate a testing set from the whole data set. Because of the few purchasing behaviors of each customer, we treat each trade as a single record, and randomly choose 10% records to form the testing set.

We first choose two standard evaluation metrics, precision and diversity. Precision is a very practical metric since the customers are usually concerned only with the top part of the recommendation list. The top-L precision of the recommendation lists for customer c_u is defined as

$$P_u(L) = \frac{h_u(L)}{L}, \quad (5)$$

where $h_u(L)$ is the number of relevant objects (purchased by c_u) ranked in the top-L places. Averaging the individual precision over all the customers, we obtain the mean precision $P(L)$.

Diversity in recommender systems refers to how different the recommended objects are with respect to each other, which is defined by considering the variety of customers' recommendation lists [31]. Given customers c_u and c_v, the difference between the top L places of their recommendation lists can be measured by the Hamming distance

$$H_{uv}(L) = 1 - \frac{Q_{uv}(L)}{L}, \quad (6)$$

where $Q_{uv}(L)$ is the number of common objects in the top-L places of the lists. If the lists are identical, $H_{uv}(L) = 0$, while if their lists are completely different, $H_{uv}(L) = 1$.

Averaging the diversity over all customers pairs, we obtain the mean precision $H(L)$.

Besides, we introduce a metric named novelty which refers to how different the recommended objects are with respect to what the users have already seen before. The initial way to quantify the novelty is to measure the average popularity of the recommended objects [2]. In this study, we re-define the novelty as

$$N_u(L) = 1 - \frac{d_u(L)}{L}, \tag{7}$$

where $d_u(L)$ is the number of objects browsed by c_u ranked in the top-L places. The high value of novelty means that the algorithm can recommend more novel objects that have not been noticed by this customer before.

IV. Experiments and Results

A. Comparisons among the benchmark methods

We consider only the top-k objects in each customer's browsing history to examine the top-L precision ($P(L)$), diversity ($H(L)$) and novelty ($N(L)$). It is because the recommendation lists would be worthless if we employ the whole history. The reason/evidence will be presented in the following results, that is the most frequently browsed objects have higher probability to be the final purchase objects. In the implementations, the values of k are selected from 1 to 20, and the values of L range from 1 to 10. Notice that, we usually can not get L recommended objects via CFA and IPA if L is large and k is small. For instance, if $k = 1$, only 1 object can be recommended by IPA. So we introduce the randomly filling policy that is randomly selecting some objects to fill the recommendation list.

Apparently, the tendencies of $P(L)$ when L changes from 1 to 10 should be descending, because each customer has at most one testing object and larger L will lead to recommending more unrelated objects. The evidences are shown in Figure 3 (P-1), (P-2), (P-3) and (P-4). We can see that IPA can provide the most accurate recommendations, better than SIA. It is a strong evidence for that customers would pay more attention to the object he wants to buy, rather than the most similar one to all the objects he browsed. In comparison, the two classical methods, PbA and CFA, have poor performances. It hints that the customers' interests, i.e., purchased objects, are not similar with each other. We can also get the evidence from the high $P(L)$ and $H(L)$ of IPA especially when $k = 20$ and $L = 1$, which means that the most favoured objects for different customers are quite different. The values of $H(L)$ are always very large (except PBA), because random filling policy would introduce various objects. As to the $N(L)$, there is a dilemma between precision and novelty. It makes sense because the recommendations are provided based on the browsing histories. As above, high $N(L)$ but low $P(L)$ of PbA support that popular objects are not so popular, which is in accordance with the results in Figure 1(c).

From Figure 3, we notice that different k will significantly impact the recommendations. Thus we present the influence of k in Figure 4. For precision, the IPA is most sensitive to the increasing k. The reasons are two folds. Firstly, the intra-similarity among the objects in each customer's history is high (see Figure 2(b)) while the inter-similarity among the histories of different customers is low (see the low precision and high diversity of CFA). Then more records will not largely influence the precisions of CFA and SIA. Secondly, the customers usually browsed one certain object multiple times before he/she placed the order (see Figure 2(a)). So the IPA can identify the related object more precisely if it knows more records. Novelty is also largely influenced by different k. At this aspect, CFA is more preferred.

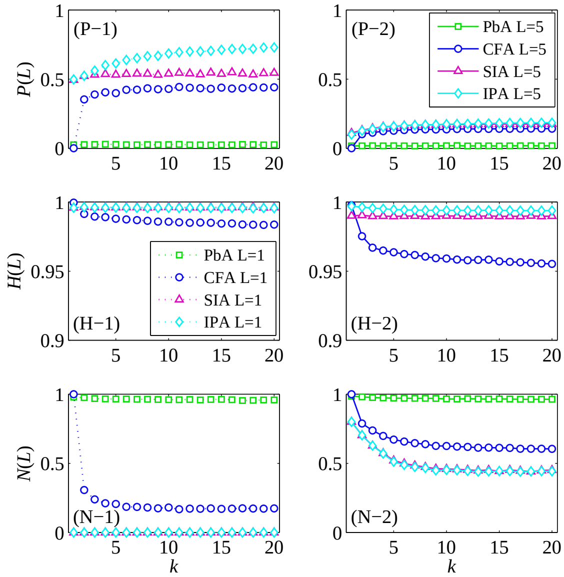

Fig. 4. The tendencies of $P(L)$, $H(L)$ and $N(L)$ when k changes from 1 to 20. The subfigures in each row correspond to one type of evaluation metric. The subfigures in each column are the results with same L, and have the same legend. We do not show the results of $H(L)$ for PbA, because the values are always 0. The results are averaged by 100 independent experiments.

Overall speaking, the benchmark method PbA is not good at either precision or diversity. Although the recommended objects are novel for each user, the lists are totally the same. For the rest three algorithms, there is a dilemma between precision and novelty. In the following, we proposed a new algorithm based on the temporal browsing histories, which can provide both precise and novel recommendations.

B. Temporal Recommender System

Previous studies revealed that old information might be redundant or even misleading [21], [22]. We are motivated to check whether the newly browsed objects are more preferred by the customer. We calculate the probability that the t-th browsed object was purchased, which is defined as

$$p(1|t) = \frac{1}{|C_t|} \sum_{c_u \in C_t} p_{ui}, \tag{8}$$

where C_t is the set of customers whose browsing histories contain no less than t records. $p_{ui} = 1$ if customer c_u purchased object o_i, and o_i is the t-th object in $B(c_u)$. The results are shown in Figure 5, where we compare the two cases of "forward" and "backward". The "forward" means the numbering (i.e., $t = 1, 2, ...$) starts from the earliest record, while "backward" means the numbering starts from the latest record. The latest records are observed to be more consistent with the purchased objects. And we are inspired to propose a new method based on the temporal browsing history, named by TBH (temporal browsing history).

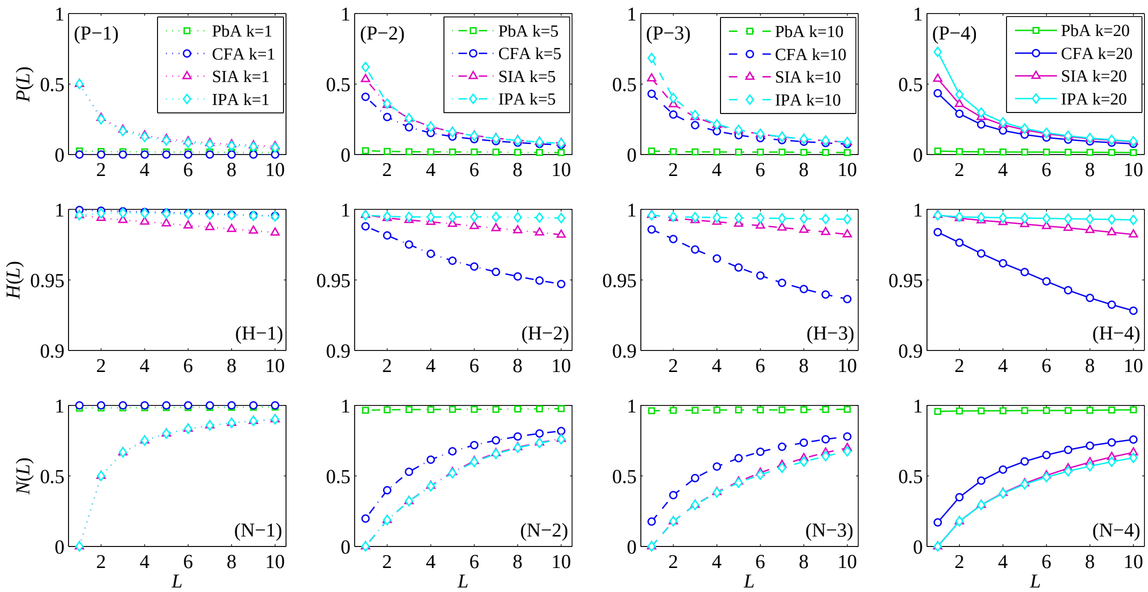

Fig. 3. The tendencies of $P(L)$, $H(L)$ and $N(L)$ when L changes from 1 to 10. The subfigures in each row correspond to one type of evaluation metric. The subfigures in each column are the results with same k, and have the same legend. k is the number of records learned by recommender systems. We do not show the results of $H(L)$ for PbA, because the values are always 0. The results are averaged by 100 independent experiments.

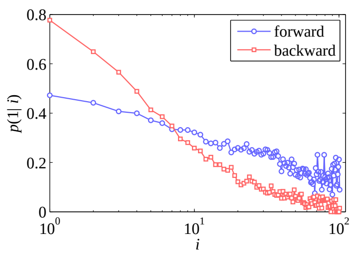

Fig. 5. The probability that the t-th browsed object was purchased. Because the number of customers whose $|B(c_u)| > 100$ is very small, we treated all the records whose $t > 100$ as a whole group (marked by $t = 100$) to avoid the effect of small sample size.

This method considers two factors. One is the similarity between the objects, defined in Eq. (3). The other is the weight of the objects in each history. By introducing the impact of time decay, the weight of an object o_j for customer c_u, denoted by $w(o_j|c_u)$, is defined as

$$s(o_i|c_u) = \sum_{o_j \in \{B(c_u)\}} s(o_i, o_j) \cdot w(o_j|c_u),$$

where $w(o_j|c_u)$ is defined following an iterative way. It is

$$w(o_j|c_u)|_t = [w(o_j|c_u)|_{t-1} + \delta(o_j, o_k)] \times f(\Delta T_t|_{c_u,o_j}),$$

where o_k is the t-th object browsed by c_u and $t \geq 1$. The function $f(\Delta T_t|_{c_u,o_j})$ is introduced to decay the weight of the object browsed at early time, defined by an exponential function $f(\Delta T_t|_{c_u,o_j}) = e^{-\alpha * \Delta T_t|_{c_u,o_j}}$, where α is to control the rate of decay. The $\Delta T_t|_{c_u,o_j}$ record the time difference between the current time t and the time when o_j was last browsed. It is set to 0 if c_u browsed o_j at time t. So we have $\Delta T_t|_{c_u,o_j} = (\Delta T_{t-1}|_{c_u,o_j} + 1) \times (1 - \delta(o_j, o_k))$. All the values of $w(o_j|c_u)|_0$ and $\Delta T_0|_{c_u,o_j}$ are initialized as 0.

When $\alpha = 0$, the first object recommended by TBH is exactly the same to that of IPA. When $\alpha = 1$, TBH is meaningless because it will recommend the latest browsed object. To find the optimal α, we compare the $P(L)$, $H(L)$ and $N(L)$ in Figure 6 (a), (b) and (c) respectively, using the most significant values, i.e., $P(1)$, $H(10)$ and $N(10)$ (shown in Figure 3 and 4). The results show that a small value of α would enhance the precision, while the larger value of α would provide more novel recommendations. However, the advantage of novelty is not significant. In Figure 6 (d)-(f), we compare TBH ($\alpha = 0.2$) with the best performances of the four basic algorithms, i.e., IPA which has the highest $P(L)$ and $H(L)$, and CFA which has the highest $N(L)$. We can see the TBH can provide more precise recommendation. Although the diversity decreases, it is still very high (larger than 0.98). As to the novelty, TBH is competitive to CFA, or even better when L and k is large. Moreover, the recommendation of CFA is not accurate enough. Overall speaking, the new method performs the best.

V. Conclusion and Discussion

Recommender systems are widely employed by online retailers, while different retailers usually needs different recommendation strategies. In this study, we find the classical methods, like collaborative filtering method and popularity-based method, can not be directly applied to recommend the plans of tourist routes. The reasons are not only due to the very few purchased behaviors per customer, but also because people does not choose very similar routes. To provide effective recommendations, we learn the customers' browsing histories and put forward a temporal recommender system. Compared with the classical method, it can learn the customers' interests at a very early stage, and then provide more accurate recommendations.

People's interest may change over time, so the old information may do harm to the precise commendations [22]. Our method provide an automatic way of weakening the role of old information. When to calculate the weight of the items for a customer, we only need to update the score based on the latest values and the time gap. Such method is much more efficient than the collaborative filtering based model. Need

ISBN: 978-988-19253-8-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

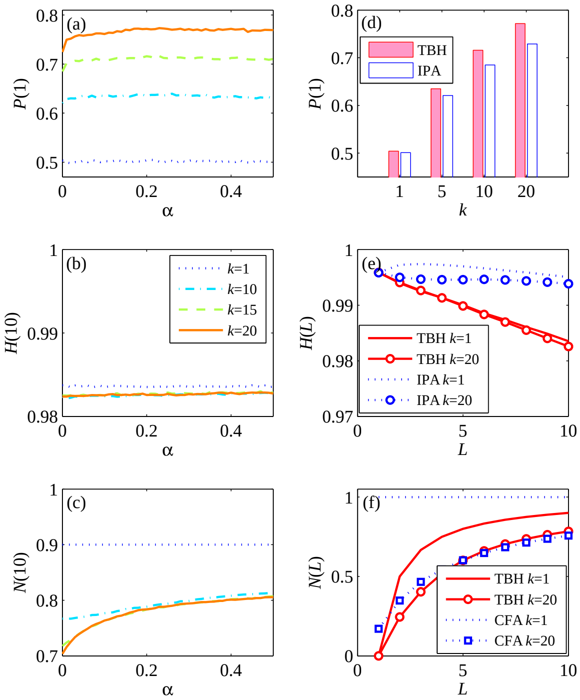

Fig. 6. Experimental results of the new algorithm TBH. Subfigure (a), (b) and (c) are to find the optimal α for TBH, and presents the values of $P(1)$, $H(10)$ and $N(10)$ respectively. Subfigure (d), (e) and (f) compares the optimal TBH with the best benchmark methods, and respectively present the comparisons for $P(1)$, $H(L)$ and $N(L)$ with different k.

to notice that, the items in one customer's recommendation list provided by our method, must have similar properties. It might be a bad recommendation in many cases, but not here. As we showed in our statistic results, the objects browsed by one customer are very similar, especially on the travelling routes. Thus we infer that customers usually have some specific travelling targets before he enters this web site. As a result, recommending similar objects in such a business system might be a better strategy of enhancing the purchase rate.

References

[1] F. Ricci, L. Rokach, and B. Shapira, *Introduction to Recommender Systems Handbook*. Springer, 2011, pp. 1–35.

[2] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, Jul. 2012.

[3] P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds., *The Adaptive Web: Methods and Strategies of Web Personalization*. Berlin, Heidelberg: Springer-Verlag, 2007.

[4] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl, "Grouplens: Applying collaborative filtering to usenet news," *Communications of the ACM*, vol. 40, no. 3, pp. 77–87, Mar. 1997.

[5] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, Jan. 2004.

[6] C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," in *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 714–720.

[7] C. Christakou, S. Vrettos, and A. Stafylopatis, "A hybrid movie recommender systems based on neural networks," *International Journal on Artificial Intelligence Tools*, vol. 16, no. 5, pp. 771–792, Oct. 2007.

[8] Y.-C. Zhang, M. Blattner, and Y.-K. Yu, "Heat conduction process on community networks as a recommendation model," *Physical Review Letters*, vol. 99, no. 15, p. 154301, Oct. 2007.

[9] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, Feb. 2010.

[10] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock, "Methods and metrics for cold-start recommendations," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 2002, pp. 253–260.

[11] L. Terveen and W. Hill, *Beyond Recommender Systems: Helping People Help Each Other*. Addison-Wesley, 2001, pp. 487–509.

[12] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[13] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, Nov. 1997.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[15] Q.-M. Zhang, M.-S. Shang, W. Zeng, Y. Chen, and L. Lü, "Empirical comparison of local structural similarity indices for collaborative-filtering-based recommender systems," *Physics Procedia*, vol. 3, no. 5, pp. 1887–1896, Jul. 2010.

[16] H. Lieberman, "Letizia: An agent that assists web browsing," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 924–929.

[17] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C. D. Spyropoulos, "Web usage mining as a tool for personalization: A survey," *User Modeling and User-Adapted Interaction*, vol. 13, no. 4, pp. 311–372, Nov. 2003.

[18] R. Suguna and D. Sharmila, "An efficient web recommendation system using collaborative filtering and pattern discovery algorithms," *International Journal of Computer Applications*, vol. 70, no. 3, pp. 37–44, May 2013.

[19] 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.

[20] P. G. Campos, F. Díez, and I. Cantador, "Time-aware recommender systems: A comprehensive survey and analysis of existing evaluation protocols," *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 67–119, Feb. 2014.

[21] Y. Ding and X. Li, "Time weight collaborative filtering," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, ser. CIKM '05. New York, NY, USA: ACM, 2005, pp. 485–492.

[22] Q.-M. Zhang, A. Zeng, and M.-S. Shang, "Extracting the information backbone in online system," *PLoS ONE*, vol. 8, no. 5, p. e62624, May 2013.

[23] Y. Koren, "Collaborative filtering with temporal dynamics," *Communications of the ACM*, vol. 53, no. 4, pp. 89–97, Apr. 2010.

[24] L. Xiong, X. Chen, T. K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with bayesian probabilistic tensor factorization," in *SIAM International Conference on Data Mining*, Sep. 2010, pp. 211–222.

[25] Q. Guo, W.-J. Song, L. Hou, Y.-L. Zhang, and J.-G. Liu, "Effect of the time window on the heat-conduction information filtering model," *Physica A: Statistical Mechanics and its Applications*, vol. 401, pp. 15–21, 2014.

[26] L. Baltrunas and X. Amatriain, "Towards time-dependant recommendation based on implicit feedback," in *RecSys 2009 Workshop on Context-Aware Recommender Systems*, New York, NY, 2009.

[27] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, "Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems," in *Proceedings of the Third ACM Conference on Recommender Systems*. New York, NY, USA: ACM, 2009, pp. 265–268.

[28] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[29] P. Jaccard, "Étude comparative de la distribution florale dans une portion des alpes et des jura," *Bulletin de la Société Vaudoise des Science Naturelles*, vol. 37, pp. 547–579, Dec. 1901.

[30] L. A. Adamic and E. Adar, "Friends and neighbors on the web," *Social Networks*, vol. 25, pp. 211–230, 2001.

[31] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *Europhysics Letters*, vol. 81, no. 5, p. 58004, 2008.

ISBN: 978-988-19253-8-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)