Connecting Social Media To E Commerce: Cold-Start Product Recommendation Using Microblogging Information

Harshini S Babu[1] ,Manasa K[2] , Pooja H N[3], Rakshitha B T[4]
[1, 2, 3] Dept of Computer Science and Engineering
[4] Professor, Dept of Computer Science and Engineering
[1, 2, 3, 4] Atria Institute of Technology, Bangalore Karnataka, India

Abstract- *Collaborative filtering (CF) algorithms have been widely used to build recommender systems since they have distinguishing capability of sharing collective wisdoms and experiences. However, they may easily fall into the trap of the Matthew effect, which tends to recommend popular items and hence less popular items become increasingly less popular. Under this circumstance, most of the items in the recommendation list are already familiar to users and therefore the performance would seriously degenerate in finding cold items, i.e., new items and niche items. To address this issue, in this paper, a user survey is first conducted on the online shopping habits in China, based on which a novel recommendation algorithm termed innovator based CF is proposed that can recommend cold items to users by introducing the concept of innovators. Specifically, innovators are a special subset of users who can discover cold items without the help of recommender system. Therefore, cold items can be captured in the recommendation list via innovators, achieving the balance between serendipity and accuracy. To confirm the effectiveness of our algorithm, extensive experiments are conducted on the dataset provided by Alibaba Group in Ali Mobile Recommendation Algorithm Competition, which is collected from the real ecommerce environment and covers massive user behavior log data.*

Keywords- Cold items, collaborative filtering (CF), innovators, recommender system, serendipity.

I. INTRODUCTION

In the era of information overload, recommender system is developed to help users discover interested items in ecommerce [1]–[4]. Collaborative filtering (CF) algorithms are Digital Object Identifier 10.1109/TCYB.2018.2841924 widely used to build recommender systems since they have distinguishing capability of sharing collective wisdoms and experiences [1], [5]–[8]. However, many CF algorithms easily fall into the trap of the Matthew Effect [9], which makes them severely tend to recommend popular items. In this case, new items would be seldom discovered and niche items in the long tail will become increasingly less popular [10]. To be specific, in this paper, new items refer to items that are released less than one day and niche items refer to items that are released more than a week but have low item popularity. Moreover, according to our user survey reported in Section III, normal users can hardly discover these items by themselves due to the limited time spent on online shopping. As a result, it is necessary to develop a recommender system that can discover new items and niche items. Since new items may have extremely short time-tolive, e.g., some garments appeared in newly released movies, it is necessary for the recommender system to be real-time, i.e., react rapidly. Besides, many of the niche items in the long tail are extremely special which means they may only serve the interests of a small group of users. To help such cold items attract attention from users and also help users better discover their personalized needs, it is necessary to introduce serendipity into recommender systems.

To address the aforementioned issues, a user survey on the online shopping habits in China is conducted, based on which a novel CF algorithm, termed innovator-based CF (INVBCF) is proposed. In particular, we introduce the concept of innovators who are capable of discovering cold items into CF. A basic assumption is that users may feel surprised if the recommender system recommends what innovators bought recently. However, when making recommendations, unlike the existing methods, we do not force users to accept cold items because users have different receptivity to product's maturity. Accordingly, the proposed algorithm first calculates user activeness, conformity and personal innovator index (PII). The PII is used to classify active users into innovators and normal users. For each normal user, the items that its nearest innovators have interacted with are used to construct the candidate recommendation list. Next, the neighbors' PII and user's conformity are both integrated into the ranking function to rank the candidate recommendation list. As a result, items recommended by innovators with high PII can get high score for users with low conformity. Therefore the proposed algorithm successfully improves serendipity for recommender system while strikes the balance between serendipity and accuracy, i.e., surprising users without forcing them to accept cold items. difference between our As will be described in the related work section, the main proposed algorithm and the existing algorithms in addressing the cold items (i.e., new items and niche items) is that the existing algorithms mainly utilize side information such as item attributes, which causes extra computational cost [11], [12] or treat all users as innovators which is not true in real-world applications [13].

To evaluate the effectiveness of the proposed algorithm, extensive experiments have been conducted on the realworld e-commerce dataset provided by Alibaba Group in Ali Mobile Recommendation Algorithm Competition. Experimental results show that the proposed INVBCF algorithm outperforms the existing algorithms on serendipity while maintains high accuracy. Additionally, parameter analysis is conducted to analyze the influence of different values of parameters.

The contributions of this paper are summarized as follows.

1. A user survey is conducted which shows the online shopping habits in China and forms the foundation for constructing the proposed recommendation method.
2. A new recommendation algorithm termed INVBCF is proposed, which can recommend new items and niche items to users by introducing the concept of innovators, achieving the balance between serendipity and accuracy.
3. An offline component and an online component are designed for implementing the proposed recommendation algorithm. The online component can be executed on users' mobile devices which makes it possible to adjust the recommendation list in real time and significantly save communication cost and computing resources of servers.

The remainder of this paper is organized as follows. Section II introduces the related work in serendipitous recommendation. In Section III, we present a user survey that reflects the online shopping habits of users in Taobao[1] which is the largest online shopping mall in China. We then describe our algorithm in detail in Section IV. Experimental results are reported in Section V. Finally, we conclude this paper and present the future work in Section VI.

II. RELATED WORK

Traditional recommender systems usually use accuracy as the main measure to evaluate performance. However, improving accuracy does not mean improving user satisfaction [14]–[16]. For example, recommending items that the user already puts into cart can reach very high accuracy but does not make any sense. A good recommender system need not only accurately predict user shopping behaviors but also broaden user horizon and discover their potential interests [17]. Hence, serendipity is introduced into recommender system and has become a very hot research topic in recent years [17]–[28]. According to [29] and [30], it can be briefly described as follows.

Definition 1 (Serendipitous Recommendation): A recommendation result is said serendipitous if it is dissimilar to user historical interests while suffices user needs.

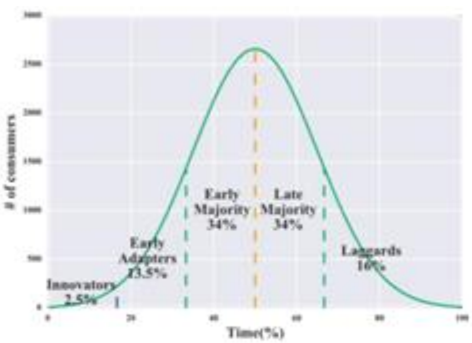

Fig. 1. Rogers' innovation adoption curve. The *x*-axis is the time-to-live of a product and the *y*-axis denotes the number of consumers purchasing this product.

To evaluate the recommendation result, we should measure the difference between the recommendation result and user historical interests. Also note that the earlier the recommender system recommends serendipitous items, the more surprised the users would feel.

A natural perspective for improving serendipity is to utilize side information like user profiles, content data, location information, etc. Murakami *et al.* [31] compared the performance of improving serendipity between the Bayesian model and the keyword filtering method. The results reveal that keyword filtering can better balance accuracy and serendipity. Zhang *et al.* [11] utilized music artist information by employing *Latent Dirichlet Allocation* technique and proposed two variants of item-based recommendation termed *Community-Aware Auralist*and *Bubble-Aware Auralist*, which can inject serendipity into music recommendation. Schedl and Hauger [12] proposed an algorithm which takes age, nation, style, and other factors into consideration while recommending music. The experimental results show that the additional information does improve user experience.

Apart from side information, some efforts have been made for addressing the cold start problem from other perspectives [32], [33]. Wang *et al.* [32] for the first time developed an active learning-based framework for broadcast email prioritization, which exploits the CF features, handles implicit feedback, and considers users' time-sensitive responsiveness. The basic idea is to send the broadcast email to a small portion of users from the mailing list and then collect the time-sensitive feedbacks for predicting the priority of the email for the remaining users. algorithm is based on the assumption that users may feel surprised if the recommender system recommends what innovators buy at the current moment [13]. Then a variant of the algorithm is developed which takes community into consideration while building a real-time recommender system [35]. Connoisseurs, who well represent opinions in their communities and lead the trends, can make impact to their fans in communities soon. Both of these two algorithms show the capability of improving serendipity. However, these methods assume that all users can be innovators and use an ergodic Markov chain to model how innovators are followed through multiple steps, which is inconsistent with the fact that only 2.5% of the users can be regarded as innovators. What is more, Hu *et al.* [36] discovered that users may have their own tipping points when choosing items and proposed a framework which recommends items based on user tipping points so as to match the maturity stage with user tipping points.

It should be noticed that producing serendipitous recommendation has been extensively studied in other domains, e.g., collaborative tagging platforms [37], [38]. Zanardi and Capra [37] developed an efficient content search method termed *Social Ranking* using tag-based recommender system, where clustering of users is utilized for improving accuracy while clustering of tags is used for improving coverage. In [38], a query expansion and user

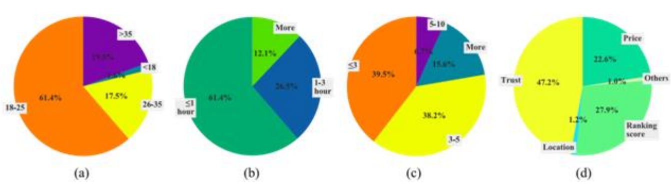

Fig. 2. Statistical graph of the user survey. (a) Age structure of respondents. (b) Distribution of time cost in a single interaction with website. (c) Distribution of the number of pages browsed in a single interaction with website. (d) Key factors that influence user decision

This active learning framework is quite effective in addressing the completely cold start problem of broadcast email prioritization. Furthermore, a novel cross-domain recommendation framework was proposed for handling large numbers of mailing lists [33]. Despite success in broadcast email prioritization, these methods are not directly applicable in e-commerce due to the reason that it is usually unsuitable to make a trial in a small portion of consumers for getting feedback.

Another perspective for improving serendipity is based on the Rogers' Innovation Theory. For illustration purpose, Fig. 1 shows the Rogers' Innovation Adoption Curve model. It is a sociological concept proposed by Rogers and was first introduced in the business model by Krueger in 2006 [34].

In e-commerce, the phenomenon can be interpreted as follows. While a new item is released, different users take different time periods to discover it. Innovators are those who can discover the item at the very beginning. Kawamae [13], [35] first introduced the Rogers' Innovation Theory into recommender system for improving serendipity. The profile enrichment approach was developed by means of deriving the most "authoritative" tags, so as to address the issues suffered by the traditional content-based (CB) and CF methods in folksonomies. However, due to the essential difference between e-commerce and folksonomic tagging system, the above approaches are not directly applicable in e-commerce for producing serendipitous recommendation results.

III. USER SURVEY

To better understand the online shopping habits in China, a user survey is conducted on two different periods: one starts from April 4th, 2016 and ends up on April 15th, 2016, the other starts from April 20th, 2017 and ends up on April 27th, 2017.

During the two survey periods, 570 answer sheets are collected, of which 59.1% are answered by students and the remaining 40.9% are by the workforce. Most of them (84.7%) have over one year online shopping experience.

According to the age structure shown in Fig. 2(a), 80.5% of them are under 35 years old, which means young people form the majority of online buyers. While 61.9% of the respondents prefer to browse online shopping website only when they have clear demands, 38.1% of the respondents say they would love to browse the website in their spare time.

It is found that most of the respondents spend less than 1 h on the website each time (61.4%) as shown in Fig. 2(b) and only browse the first 5 pages returned by the embedded search engine (77.7%) as shown in Fig. 2(c). Furthermore, they are also asked what factors are considered when sorting results returned by the search engine. This is a multiple-choice question with 63.7% of the respondents adopting sorting by sales volume, 43.9% of the respondents adopting sorting by price, 38.9% of the respondents adopting sorting by ranking scores, and 32.6% of the respondents adopting sorting by popularity. Interestingly, the order of factors that users choose to sort results is quite different from what they think is the most important factors that determine the purchase. From Fig. 2(d) we can see that nearly half (47.2%) of the respondents regard trust as the most important factor that influences their decision, then ranking score or comments (27.9%), and still, quite a lot (22.6%) of the respondents give priority to price.

The discovery of survey could be summarized as follows.

1. Most users using online shopping service are very inactive since they only interact with the website when they need to buy something and therefore they provide relatively little information.
2. Users are not likely to spend too much time on the website each time and they mostly rely on the toplisted results presented by the embedded search engine.
3. Sales volume, i.e., item popularity, is the first factor for most people when making purchasing decisions.
4. Trust, i.e., the quality, is of significance that users concern about.

Therefore, safe conclusion can be drawn that nearly all TABLE I users want to purchase high-quality items but they are not willing to spend too much time on finding such items. Moreover, the

STATISTICS OF ITEM POPULARITY AND USER ACTIVENESS OF THE USED DATASET

	min	25%	50%	75%	mean	max
Item popularity (Global)	1	1	1	1	1.86	956
Item popularity (Released more than one week)	1	1	1	2	2.04	956
User activeness (Global)	1	111	262	557	442.94	10984

above analysis shows that most of them can only discover popular items presented in the top list and the remaining massive new items and niche items cannot even get a chance for being discovered. In this case, a serendipitous recommender system is a necessity to help users discover new items and niche items. However, we should also note that users have different receptivity for new items and niche items, so we should strike the balance between serendipity and accuracy.

IV. PROPOSED FRAMEWORK

In this section, based on the experimental dataset provided by Alibaba Group in Ali Mobile Recommendation Algorithm Competition, we first define and analyze some key concepts used in our algorithm. Then the proposed algorithm is described in detail with the two components for implementation purpose.

A. Key Concepts

1) Item Popularity and User Activeness: To quantify the popularity of item *i*and the activeness of user *u*, two evaluation metrics P_i and A_u are introduced, respectively, as follows:

$$P_i = \sum_{(u,i) \in U} \max(I_{browse}(u,i), I_{collect}(u,i), I_{cart}(u,i), I_{purchase}(u,i)) \quad (1)$$

$$A_u = \sum_{(u,i) \in I} \max(I_{browse}(u,i), I_{collect}(u,i), I_{cart}(u,i), I_{purchase}(u,i)) \quad (2)$$

whereI and U denote the item set and the user set, respectively, the $I_{behavior}(u,i)$ function is an indicator function

IEEE Transactions on Cybernetics (Vol: PP, June 2018)

which is equal to 1 if user *u* has interacted with item *i*in the specific behavior type (namely browse, collect, cart or purchase), and 0 otherwise. In this way, the popularity of an item is measured by the number of users who have interacted with it, and the activeness of a user is measured by how many items that he or she has interacted with.

Table I shows some statistical data of *item popularity* and *user activeness* of the dataset used in experiments. We can see that, over 75% of the items have been interacted with users only once and the mean values of both *item popularity* and *user activeness* are far smaller than the max value. Besides, the popularity of items that have been released more than one week also severely tends to be low, which implies there are massive niche items lying in the long tail. Also, Fig. 3 shows the distributions of *item popularity* and *user activeness*. Both of them follow a linear trend with double logarithmic coordinates, which coincide with the long-tail distribution, especially *item popularity*. As we have discussed

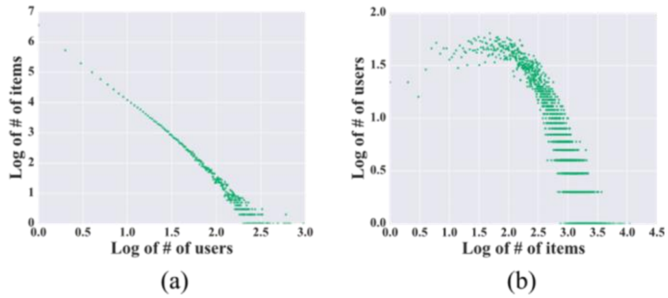

Fig. 3. Statistical graph of item popularity and user activeness of the used dataset. (a) Distribution of item popularity, with *x*-axis denoting the log of popularity and *y*-axis denoting the log of the number of items having such popularity. (b) Distribution of user activeness, with *x*-axis denoting the log of activeness and *y*-axis denoting the log of the number of users having such activeness.

in Sections I and III, cold items (including new items and niche items) and inactive users are the majority in real world.

2) Average Time Lag: First of all, to quantify the ability of users to discover new items, the average time lag (ATL) is defined as follows.

Definition 2 (ATL): The ATL of a user *u*, denoted as

$$\mathrm{ATL}_u \text{, is defined as} = \mathrm{ATL}_u = \frac{\sum_{i\in\mathcal{I}} (d_{u,i} - r_i)}{A_u} \quad (3)$$

where$I_u$denotes the set of items user *u* has interacted with, i.e., browse, collect, cart or purchase, $r_i$denotes the release date of item *i*, $d_{u,i}$denotes the date when user *u* discovers item *i*(first interacts with it), and A_u is the *user activeness* defined in (2).

The time lag between the release date $r_i$and the first interaction date $d_{u,i}$can reflect how long does it take for user *u* to discover item *i*. Short time lag implies the user is capable of discovering *new* items, which is a necessity of being innovator. However, it is worth noticing that large time lag may not always imply a user is incapable of discovering *new* items. As aforementioned in the user survey, most of the users purchase items only when they need them, meaning that they only surf online shopping websites if necessary, which can also cause long time lag to discover items. Based upon the above analysis, the time lag of discovering less popular items should have a larger impact on quantifying the ability of users to discover new items than the time lag of discovering popular items since users may already know popular items but do not need them. As a result, to better evaluate the ability of users to discover new items, inspired by the inverse document frequency, we define modified ATL (MATL_u) as follows.

Definition 3 (MATL): The MATL of a user *u*, MATL_u, is defined as

$$\mathrm{MATL}_u(4) = \frac{\sum_{i\in\mathcal{I}} (d_{u,i} - r_i) \cdot \mathrm{IIF}_i}{A_u}$$

where$\mathrm{IIF}_i = \log_2(|U|/P_i)$ is the inverse item frequency with |U| being the number of users.

Since the number of users is fixed, less popular items tend to have larger IIF than popular items. In the MATL, the time lag is weighted by IIF, which makes the modified time lag

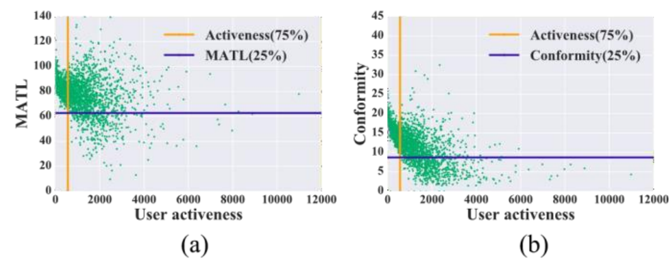

Fig. 4. Statistical graph of user behavior relations. (a) Relation between user activeness and MATL, with *x*-axis denoting user activeness and *y*-axis denoting MATL for users having such activeness. (b) Relation between user activeness and conformity, with *x*-axis denoting user activeness and *y*- axis denoting conformity for users having such activeness.

for a less popular item larger than the modified time lag for a popular item when they have the same time lag. In this way, the impact of less popular items is magnified while the impact of popular items is narrowed down. As a result, the MATL can better evaluate the ability of discovering new items and punish users who take a long time to discover less popular items.

Fig. 4(a) shows the relation between the *user activeness* and the MATL. The orange line shows the 75% percentile of the *user activeness* and the purple line shows the 25% percentile of the MATL. Together they divide the graph into four areas. The upper left contains all the inactive users and they have a high MATL_uwhich means they seldom discover new items. The upper right shows that there are still quite a lot of active users who do not have the ability to discover new items. But the lower part shows that all the users having low $\mathrm{MATL}_u$are active users, which verifies the proposition below.

Proposition 1: Users who have low MATL are also more likely to have more interactions with the website (i.e., active users).

3) Conformity: Conformity measures how likely a user would follow the mainstream. It is an important metric for identifying innovators who can discover niche items and preventing the recommender system from recommending improper items for users. The *conformity* of user *u*, which is measured by the average item popularity of items that user u has interacted with, is defined as follows.

Definition 4 (Conformity): The *conformity* of a user *u*, denoted as C_u, is defined as

$$C_u = \frac{\sum_{i \in I_u} P_i}{A_u} \quad (5)$$

Users with low conformity seldom follow the mainstream and hence they are more likely to discover niche items. Fig. 4(b) shows the relation between *user activeness* and *conformity*. The orange line shows the 75% percentile of *user activeness* and the purple line shows the 25% percentile of *conformity*. Together they divide the graph into four areas.

The upper left contains almost all the inactive users and they have a high *conformity* which means they seldom discover niche items. The upper right shows there are still some active users who do not have the ability to discover niche items. But the lower part shows that all the users having low conformity are active users, which verifies the proposition below.

Proposition 2: Users who have low conformity are also more likely to have more interactions with the website (i.e., active users).

4) Innovator: From Definition 1, we can see that there are two basic requirements for serendipitous recommendation, i.e., being dissimilar to user historical interests and sufficing user needs. To meet the former requirement, we can introduce some niche items which may be quite different to user historical interests but actually conform to user potential interest. To meet the latter requirement, we need to find a proper way to model user interests and recommend proper items, as will be discussed in the next section. What is more, to maximize serendipity, we should also shorten the time that users spend in discovering new items, as shown in the Rogers' Innovation Adoption Curve in Fig. 1. As a result, innovators are required to have low *conformity* for discovering niche items and low MATL for introducing new items at the very beginning. What is more, since the proposed recommendation method only considers items that have been interacted with innovators, the innovators with higher user activeness can ensure that more items are introduced into the candidate recommendation list.

According to the above analysis, innovators should have high user activeness A_u, low MATL $MATL_u$ and low conformity C_u. Combining these three features together, the PII is defined as follows to measure how likely a user is innovator.

Definition 5 (PII): The PII of a user *u*, denoted as PII_u, is defined as

$$PII_u = A_u^* - MATL_u^* - C_u^* \quad (6)$$

Note that in this formula, we rescale [2] the range of A_u, $MATL_u$ and C_u to the range of [0,1], aiming to avoid being influenced by the scale of a dimension.

To demonstrate the rationale of the definition of PII, assume that there are two users *a* and *b* having the same user activeness (the numbers of items they have interacted with are the same), i.e., $A_{ua} = A_{ub}$, and they have exactly the same time lag of discovering items, i.e., the same $d_{u,i} - r_i$ for all items they have interacted with. However, suppose that user *a* interacts with items that are all popular and user *b* interacts with items that are all unpopular, i.e., the item popularity P_i of each item user *a* has interacted with is larger than P_i of each item user *b* has interacted with. In this case, according to (4), $MATL_{ua}$ is smaller than $MATL_{ub}$. However, by (5), C_{ua} is larger than C_{ub}. Therefore, according to (6), it is still possible that the personal innovator index PII_{ub} is larger than PII_{ua}, i.e., user *b* is more likely to discover unpopular items.

B. Proposed Recommender System

For implementation purpose, the proposed recommender system is divided into an offline component and an online component. The offline component is used to train parameters used in the algorithm with the latest data and the online component is used to present recommendation result directly to users. Since massive data are generated on the website and mobile devices every second, the processing speed is of significance. The offline component is not usually executed

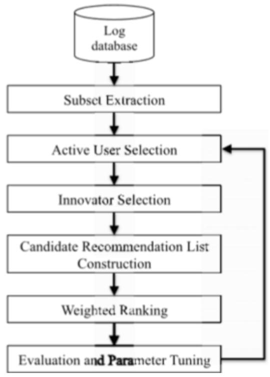

Fig. 5. Flowchart of the offline component of the proposed recommender system.

in real time because processing data collected from massive users and items can be time consuming. However, the needs of user are changing all the time. Hence, catching the needs of users accurately in real time via the online component is also very important for a successful recommender system. To this end, both of the offline component and the online component of the proposed recommender system are designed in this paper.

1) Offline Component: The offline component is designed to train parameters that we use in the proposed method. Also note that, a normal user may become senior and gradually learns how to discover cold items. Meanwhile an innovator may degenerate to a normal user. As a result, the offline component needs to keep track on user behaviors so as to provide up-to-date information for the online component. The flowchart of the offline component of the proposed recommender system is shown in Fig. 5. In what follows, we will introduce the offline component in detail.

Subset Extraction: The first step is to extract subset from the entire log database according to the need of the particular tasks, i.e., recommend a specific genre, category, or make recommendation based on short-term interests. Next, the subset of the log database is further divided into three parts, namely, feature train log D*F*, neighbor train log D*N*, and test log D*T*. For a user *u*, D*F* is used to calculate *u*'s features and D*N* is used to find *u*'s neighbors who are used to construct candidate recommendation list for *u* in the subsequent steps. Besides, D*T* is used to evaluate the performance so that we can adjust parameters to achieve the best performance. D*N* is the latter part of D*F* and $\mathcal{D}_F \bigcap \mathcal{D}_T = \emptyset$. Let t_F, $t_N$and $t_T$denote the time span of D*F*, D*N* and D*T* respectively. A longer $t_F$can ensure high confidence and stability of user features. However, the recommender system may fail to catch user recent performance if $t_F$is setting too long. In addition, in serendipitous recommendation, there is a time span between the date the recommender system recommends items and the date users actually discover the recommended items by themselves. To measure the length of such time span, D*T* should be across a time span, i.e., $t_T$is larger than 1, which is different from the traditional predicting task that predicts for one specific day. IEEE Transactions on Cybernetics (Vol: PP, June 2018)

Active User Selection: In this step, the full user set U is divided into cold user set U*c*and active user set U*a*according to *user activeness* A_u calculated from the feature train log D*F*. In particular, U*a*= {*u* ∈U|A_u≥ threshold of user activeness}.

In the real e-commerce environment, cold users seriously influence the performance of recommender system since they do not provide enough information. Although the implicit feedback is much abundant compared to the explicit feedback, as aforementioned, cold users still occupy the vast majority. For U*c*, serendipitous recommendation does not work well since they highly rely on the popularity metric. As a result, we need to offer different recommendation schemes for users in U*c*, i.e., CB methods [39]–[42], demographic-based methods [43]–[45] or we can even utilize their social information [41], [46], [47]. Once a user *u* in U*c*provides enough feedback, he or she can be joined into U*a*and enjoy serendipitous recommendation services. In general, users in U*c*are filtered out in this phrase and only the users in U*a*are retained to further participate in the proposed recommender system.

Innovator Selection: In this step, the innovators Uinnovatorare selected from the active users U*a*according to personal innovator index PII_ucalculated from the feature train log D*F*. In particular, after sorting the active users in the descending order of PII_u, the most active users are selected as Uinnovator, such that (|Uinnovator|/|U|) × 100% is equal to the innovator percentage (%). Additionally, the normal users are defined as Unormal= U*a*− Uinnovator, i.e., the remaining active users.

Candidate Recommendation List Construction: In this step, from the neighbor train log D*N*, the candidate recommendation list is constructed for each normal user based on the items the nearest innovators have interacted with. In particular, for each normal user *u* ∈Uinnovator, the *K* nearest innovators Ninnovator(*u*) are first selected with the largest cosine similarities of the user-item interaction vectors obtained from the neighbor train log D*N*. And then the item sets of the *K* nearest innovators within the neighbor train log D*N* that this normal user *u* has not interacted with are used to construct the candidate recommendation list Icandidate(*u*) of the normal user *u*. That is, Icandidate(*u*) = {*i*∈I*N*|∃*v* ∈Ninnovator(*u*) s.t.*I*behavior(*v*,*i*) = 1, and *I*behavior(*u*,*i*) = 0}, where I*N* denotes the item set in the neighbor train log D*N*. Notice that, unlike the existing CF, the nearest neighbors of one user are not selected from the users having the most similar item-interaction behaviors. Instead, only innovators having the most similar item-interaction behaviors will be selected as the nearest neighbors. As a comparison, the latter may have much weaker ties than the former. This can be taken as the usage of weak ties in making recommendation based on innovators [48]. Moreover, this can also significantly reduce the time complexity of computing similarity from |U| ∗ |U| as in user-based CF (UBCF) to |U| ∗ |Uinnovator|, where |Uinnovator| is far smaller than |U|.

Weighted Ranking: After obtaining the candidate recommendation list Icandidate(u) for each normal user u, in this step, the weighted ranking is applied for all the items in Icandidate(u) such that the top k items with the largest ranking values are used to form the final recommendation list Irecommend(u) and recommended to the normal user u.

Typically, a basic ranking

Fig. 6. Flowchart of the online component. function of the item $i$to use u is defined as follows:

$$\mathrm{Rank}(u, i) = \sum_{v \in \mathrm{Ninnovator}(u)} w_{u,v} \times r_{v,i} \qquad (7)$$

whereNinnovator(u) denotes the nearest innovator set of user u, $w_{u,v}$denotes the similarity between user u and neighbor innovator v, $r_{v,i}$denotes user v's interest in item i. With implicit feedback data, $r_{v,i}$can be regarded as the behavior score, i.e., v's interest in $i$is measured according to the way how v has interacted with i. For example, browsing gets 1 point and purchasing gets 4 point. Otherwise, we can just simply set $r_{v,i}$as 1 if user v has interacted with item i. As analyzed in [49]–[52], it is likely that slightly different results will be generated by the two different scoring methods. Since the main focus of this paper is to investigate how the innovators can be used to improve serendipitous recommendation, $r_{v,i}$is set by the latter approach for simplicity.

To better provide serendipitous recommendation, the ranking function for different users should be tailored to their respective receptivity to product's maturity. Also, an innovator with high PII_uhas more influence than an innovator with low PII_u. As a result, the similarity is redefined by considering the PII and *conformity* as follows:

$$\hat{w}_{u,v} = w_{u,v} + \alpha \times \left(\mathrm{PII}_v^* - C_u^*\right). \qquad (8)$$

Note that in this formula, we rescale the range of $\mathrm{PII}_v$and C_u to the range of [0,1], aiming to avoid being influenced by the scale of a dimension. The revised similarity $\hat{w}_{u,v}$is then used in (7) to rank the candidate recommendation list Icandidate(u) for each user. If the recommendation comes from an innovator v with high PII_vvalue, the similarity $\hat{w}_{u,v}$will be enhanced. Besides, to avoid recommending cold items to an inappropriate user u who tends to prefer popular items, the *conformity* C_u is taken into consideration to reduce similarity $\hat{w}_{u,v}$. Hence, cold items discovered by innovators can only get relatively high ranking score Rank(u,i) if the user has low *conformity* which implies he or she is willing to accept cold items. The scaling factor α is used to adjust the contributions of $\mathrm{PII}_v$and C_u.

Evaluation and Parameter Tuning: After generating the final recommendations for all normal users, some evaluation measures are calculated by comparing the recommendation lists and the ground-truth item lists in the test log DT. Therefore, the parameters like the threshold of A_u, innovator percentage (%), the number of neighbors$K$ and the scaling factor α can be adjusted to generate better results.

2) Online Component: The online component is designed to offer serendipitous recommendation for users in real time, the flowchart of which is shown in Fig. 6. For a user u, the user database Du is updated whenever u interacts with the e-commerce mobile application. Besides, the innovator database Dinnovator(i.e., the log database associated with the innovator set Uinnovator) is updated daily according to the offline component. Top-K nearest innovators are found using KNN once u produces new feedbacks. Then the subsequent procedure is similar to what we have discussed in the offline component. Note that, since innovators are minority users, the scale of innovator database is small enough so that it can be transported to users' mobile devices like cellphones with acceptable communication cost. For example, an innovator database collected from 142 innovators in seven days, containing more than 10000 records, is just around 400 KB. As a result, we do not need to compute and update users' recommendation list on the server. Instead, the online component can be directly executed on users' cellphones. In general, the largest advantage of the proposed online component is that the recommendation list can be updated in real time according to user's newest feedback while servers do not need to compute recommendation result and transport it to user's mobile device whenever a user interacts with the mobile application. Therefore, the online component can significantly reduce communication cost and computing resources.

V. EXPERIMENTS

In this section, we first introduce the dataset and its usage. Then we introduce the evaluation metrics. Specifically, we propose a new metric called *the average distance*, to evaluate the difference between the recommendation result and user historical interests. After that, parameter analysis is conducted to analyze the effect of the five parameters on our method. An interesting fact is that the best performance on serendipity is achieved when 3% of the users are considered as innovators, which is very close to the Rogers' Innovation Theory, i.e., about 2.5% of the users are innovators. In addition, feature analysis is conducted to show the impact of different features. Finally, comparison experiments and case studies are also conducted to evaluate the effectiveness of the proposed method. In general, the proposed algorithm outperforms the existing algorithms on serendipity while maintains high accuracy. All the experiments are implemented in Anaconda 1.4.0 edition on a workstation (Windows10 64bit, 12 Intel 2.40GHz processors, 128GB of RAM).

A. Dataset

The log dataset we use in this paper is provided by Alibaba Group in Ali Mobile Recommendation Algorithm Competition, [3] which comes from real e-commerce and covers massive user behavior log data. In addition, the dataset has not been preprocessed except for some necessary encryption for protecting user personal information. The log Dataset contains totally 23291027 records collected from November 18, 2014 to December 18, 2014, including all behavior log of 20000 randomly selected users. It covers 4758484 different items in 9557 categories that have been interacted with users at least once.

Table II shows a sketch of the log dataset. Each log record contains six fields. Field user_idis the unique identifier for

TABLE II
SOME SAMPLES OF LOG DATA

user_id	item_id	behavior	user_geo	item_cat	time
10001082	53616768	1	None	9762	2014-12-02 15
10001082	285259775	1	97lk14c	4076	2014-12-08 18
10001082	4368907	1	None	5503	2014-12-12 12
…	…	…	…	…	…

each user and field item_idis for items. Field behavioris used to mark how users interact with items and is given in digital form, namely, number 1 denotes browsing, number 2 denotes collecting, number 3 denotes adding to cart, and number 4 denotes purchasing. Field item_catis also given in digital form and marks the unique category for each item. Encrypted field user_geois given to mark where the action is taken. However, due to the fact that users tend to close the location service on their cellphone, most of the log data are missing geohash information. Finally, the generating time for log is also provided which is accurate to one hour. In this paper, we abandon field user_geoand extract the date information from field time. Moreover, dates are then encoded into 0–30 to be more convenient for calculations.

In this paper, we aim to accurately locate user short-term interests. To this end, 21 consecutive days (three consecutive weeks) data are extracted and tested in each experiment. The first two weeks of the three consecutive weeks are used to generate features (i.e., D*F*), the middle week of the three consecutive weeks is used to implement the recommendation task (i.e., D*N*), and the last week of the three consecutive weeks is used to build testing data (i.e., D*T*). The new dataset used in the next experiment can be obtained by moving forward the start and the end date of the previous experiment. In this way, the log dataset is divided into eleven datasets in total. The first one is used to determine the optimal parameters of the proposed algorithm and the remaining ten datasets are used to compare the final performance of the proposed algorithm with other algorithms under unbiased estimation.

B. Evaluation Measures

Six evaluation measures are used for evaluating the recommendation performance from the following three perspectives.

Accuracy: Following the common strategy [39], [53], [54], precision and recall are utilized to evaluate the accuracy of the recommendation list, i.e., how well does the proposed recommender system hit user interests, which are defined as follows:

$$\text{Precision} = \frac{|\mathcal{R}\cap\mathcal{T}|}{|\mathcal{R}|}, \qquad \text{Recall} = \frac{|\mathcal{R}\cap\mathcal{T}|}{|\mathcal{T}|} \quad (9)$$

whereR denotes the user-item pairs in recommendation list and T denotes the user-item pairs in test log.

Serendipity: To evaluate the serendipity of recommendation, we use the average difference (AD) time [13] and the average distance (AvgDistance) as evaluation metrics.

The AD time describes the average temporal interval between the date that the recommender system recommends items and the earliest date that users discover items by themselves. Since the testing set covers seven days, we can observe

IEEE Transactions on Cybernetics (Vol: PP, June 2018)

how serendipitous recommendation works on shifting the date that users discover items to an earlier time. The AD time metric is defined as follows:

$$AD = \frac{\sum_{(u,i)\in\mathcal{R}_{hit}} |d_{u,i} - \hat{d}_{u,i}|}{|\mathcal{R}_{hit}|} \quad (10)$$

whereR_{hit}denotes the successfully predicted user-item pairs, $d_{u,i}$denotes the date when user u discovers the recommended item i, $\hat{d}_{u,i}$denotes the date when the recommender system recommends item $i$to user u, namely, the first day in the latter week. AD demonstrates how the recommendation result surprises users from the time perspective. The higher AD value indicates the more surprised the users would feel.

However, according to Definition 1, time factor cannot fully reflect serendipity. The difference between the recommendation result and user historical interests is also of significance. Therefore, we propose the AvgDistance metric which is inspired by the distance measurement proposed in [29]. To fully make use of the category information, we redefined the distance measurement as follows:

$$d(i, \mathcal{I}_u) = \frac{1 + m_{\mathcal{I}_u} - m_{\mathcal{I}_{u,C(i)}}}{1 + m_{\mathcal{I}_u}} \quad (11)$$

where $m_{\mathcal{I}_{u,C}(i)}$ denotes the number of items belonging to the same category of item $i$in $\mathcal{I}_u$(item $i$belongs to category C), and $m_{\mathcal{I}_u}$denotes the maximum number of $m_{\mathcal{I}_{u,C}}(i)$. Since the distance measurement only measures the distance for a single item, we expand it to measure the distance for the whole recommendation set. The AvgDistance is defined as follows:

$$AvgDistance = \frac{\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{R}(u)} d(i, \mathcal{I}_u)}{|\mathcal{T}|} \quad (12)$$

whereR (u) denotes the items the recommender system recommends to user u. The higher AvgDistance indicates the more surprised the users would feel.

Novelty and Coverage: In order to comprehensively evaluate the effectiveness of the proposed method, we also use novelty and coverage as evaluation metrics. A recommended item is said to be novel if the target user never found it before. Since all kinds of interactive records are included in the dataset, we can identify and filter items that a user has found before. As a result, in this paper, we use the average popularity of recommended items to measure novelty, i.e., the lower average popularity implies higher novelty. In addition, the coverage metric is measured by the proportion of recommended items in the full item set, i.e., the higher coverage implies more items can be recommended to users. The average popularity

(AvgPopularity) metric and the coverage metric are defined as follows:

$$AvgPopularity = \frac{\sum_{i\in\mathcal{R}} P_i}{|R|}, \quad Coverage = \frac{|R|}{|I_{train}|} \quad (13)$$

whereI_{train}denotes the set of items used to train neighbors.

C. Parameter Analysis

In this section, we analyze the effect of the five parameters on our method, namely, the threshold of A_u, the innovator percentage (%), the number of neighbors (K), the scaling factor

TABLE IIIDEFAULT VALUES OF THE THRESHOLD OF *Au*, THE INNOVATOR

PERCENTAGE (%), THE NUMBER OF NEIGHBORS (K), THE SCALING FACTOR α, AND THE NUMBER OF RECOMMENDED ITEMS (k)

Parameter	Threshold of A_u	Innovator percentage (%)	K	α	k
Default value	10	3%	10	1.0	20

α, and the number of recommended items for each user (k). Following [55] and [56], when analyzing the effect of a single parameter, we fix the others. For clarity, the default values of the five parameters are listed in Table III.

1) *Threshold of A_u:* From Fig. 7, we can see that, the accuracy of INVBCF tends to increase when increasing the threshold of A_u. This is because, INVBCF brings cold items into recommendation list and users with higher A_u also have higher receptivity to cold items. Also, we can notice that, the serendipity of INVBCF first increases and then decreases. This is because, users with high A_u may feel less surprised to cold items since they can discover cold items on their own. The average popularity is low when we set high threshold of A_u. This is because, for users with high A_u, cold items can occupy the top list after ranking in INVBCF. However, the coverage of INVBCF declines when increasing the threshold of A_u since filtering too many users would result in the loss of candidate items. Overall, we suggest to set the threshold of A_u as 10 since it generates the best score on the serendipity measures and also performs well in terms of other measures.

2) *Innovator Percentage (%):* From Fig. 8, we can see that, the accuracy of INVBCF first increases and then decreases when we increase the innovator percentage (%). Marking too many users as innovators doesn't help improving accuracy since innovators are the minority in the real-world scenario. However, using too few innovators also causes negative effects since they can not fully represent other users. As for serendipity, we can obtain the best score when setting the innovator percentage (%) as 3%. We also notice that, the proportion of innovators is 2.5% in the Rogers Innovation Adoption Curve, which is very close to the proportion of innovators when we set the innovator percentage (%) as 3%. When we set the innovator percentage (%) as 2%, we obtain the best score on the average popularity and the other settings perform very similarly on this metric. As a drawback of INVBCF, reducing the number of innovators also reduces the items that can be recommended since the recommended items are generated only from items that have been interacted with innovators. Overall, we suggest to set the innovator percentage (%) as 3% since it generates the best score on the serendipity measures and also performs well in terms of other measures.

3) *Number of Neighbors (K):* From Fig. 9, we can see that, the accuracy of INVBCF does not show a stable trend when we increase the number of neighbors. It depends on the specific situation. From the perspective of serendipity, it first increases and then decreases when increasing the number of neighbors. This is because, using more neighbors can better fit user potential interest but using too many neighbors does introduce noises. Moreover, increasing the number of neighbors also causes high average popularity and low coverage.

This is because, the item vectors of different innovators are mostly orthogonal when we use a small number of neighbors, i.e., more items can be included in the recommendation list, especially for cold items. But when we increase the number of neighbors, popular items have higher probability of occupying the top list. Overall, we suggest to set the number of neighbors as 10 since it generates the best score on the serendipity measures and also performs well in terms of other measures.

4) *Scaling Factor (α):* From Fig. 10, we can see that, the accuracy of INVBCF first increases and then slightly decreases when we increase the value of the scaling factor. This is because, increasing the value of the scaling factor can improve the influence of user's conformity and neighbor's PII, which enables the recommender system to introduce cold items properly. However, introducing cold items does slightly decline accuracy since users significantly tend to prefer popular items. From the perspective of serendipity, we see a difference between using and not using the scaling factor. The result shows that using scaling factor does help improving serendipity. For average popularity and coverage, the difference between using and not using the scaling factor is obvious. In the meanwhile, the performance is quite close while using scaling factor. Overall, we suggest to set the value of the scaling factor as 1.0 since it generates the best score on the serendipity measures and also performs well in terms of other measures.

5) *Number of Recommended Items for Each User (k):* From Fig. 11, increasing the number of recommended items surely increases recall but decreases precision. We can also notice that, the serendipity first increases and then decreases when we increase the number of recommended items. This is because, although cold items can be included in the top list, but popular items are still the majority of the top list. Recommending the top 20 items performs the best on serendipity and then more and more popular items get involve when increasing the number of recommended items. It is more intuitive if we look at the performance on average popularity. Also, increasing the number of recommended items will improve coverage, but the speed of improvement will be slowed down finally. Overall, recommending around 20 items is the best since the recommender system can strike the balance between multiple performance measures in this setting.

D. Feature Analysis

Since the algorithm portrays innovators from three different perspectives, namely, user activeness, conformity, and MATL, in this section, we analyze these three features to get better understanding of how these features influence the recommendation result in INVBCF by removing each of them from (6). The impact of features is shown in Table IV.

User Activeness (Au): By comparing the third column with the second column, we can see that, the accuracy metrics decrease distinctly without the constraint of Au. In this case, innovators are not required to have high activeness and hence they may not have the actual abilities of innovators but coincidentally discover some cold items in the choosing with lots of popular items and hence decrease serendipity of the recommendation lists. Furthermore, novelty and coverage decrease slightly as well. Overall, the algorithm still works well on the accuracy metrics.

Modified ATL (MATLu): By comparing the fifth column with the second column, we can see that, the serendipity

IEEE Transactions on Cybernetics (Vol: PP, June 2018)

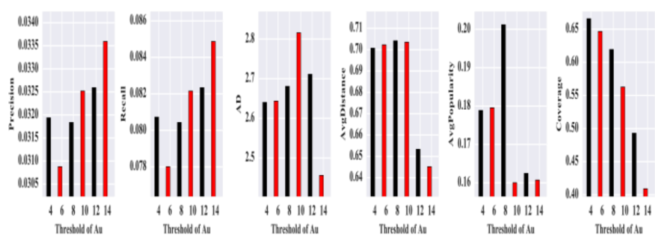

Fig. 7. Parameter analysis: the performance with different threshold of *Au* in INVBCF.

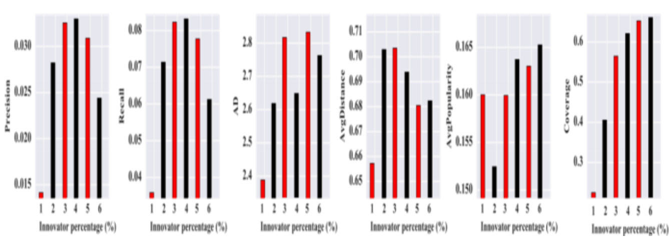

Fig. 8. Parameter analysis: the performance with different innovator percentage (%) in INVBCF.

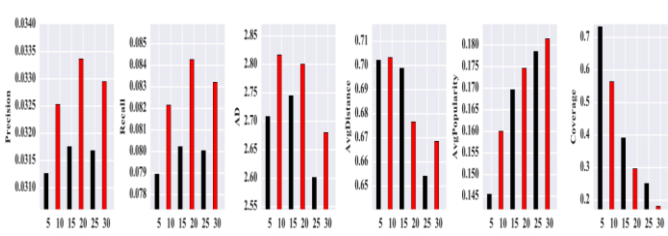

Fig. 9. Parameter analysis: the performance with different number of neighbors in INVBCF.

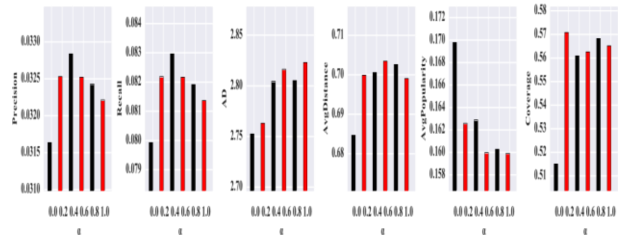

Fig. 10. Parameter analysis: the performance with different scaling factor α in INVBCF.

time span. In addition, also notice that the algorithm can still work well on other performance metrics, i.e., serendipity, novelty, and coverage, which shows the effectiveness of other two features. Conformity (Cu): By comparing the fourth column with the second column, we metrics decrease without the constraint of MATLu. In this case, the innovators may interact with lots of mature items and hence decrease serendipity of the recommendation lists. Coverage decreases as well but novelty is slightly better than the best performance of the INVBCF algorithm, i.e.,

can see that, the serendipity metrics decrease without the constraint of C_u. In this case, the innovators may interact 0.150909. Overall, the algorithm works well on accuracy representations for products, we use the user vector to metrics. represent items. Based on the vector representations, items are clustered by applying the K-means method.

E. Comparison Results

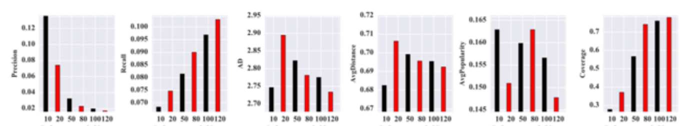

Fig. 11. Parameter analysis: the performance with different number of recommended items in INVBCF.

In this section, we report the comparison results between INVBCF and other six recommendation methods. The design of comparison experiments aims to answer the following questions: 1) Does introducing serendipity significantly influence the accuracy of recommender system? 2) Does the proposed recommendation method help users improve the probability of discovering potential interests earlier? and 3) How does the proposed recommendation method perform on other metrics like novelty and coverage?

TABLE IV
IMPACT OF DIFFERENT FEATURES

Metrics	INVBCF	without A_u	without C_u	without $MATL_u$
Precision	0.0738	0.0686	0.0727	0.0724
Recall	0.0746	0.0693	0.0735	0.0731
AD	2.8945	2.7405	2.4637	2.4497
AvgDistance	0.7060	0.7941	0.6801	0.7124
AvgPopularity	0.1509	0.1459	0.1555	0.1507
Coverage	0.3694	0.3811	0.3647	0.3674

1) Compared Methods: Six methods have been compared as follows.

1) *UBCF [57], [58]:* It recommends items bought by similar users.

2) *Item-Based CF (IBCF) [1], [59]:* It recommends items that are the most similar to the items bought by the user previously.

3) *Most Popular (MostPop):* This method recommends the top-*k* most popular items to users. Notice that in this case, all target users have the same recommendation list.

4) *Random Unpopular (RandUnpop):* This method recommends less popular items randomly. The threshold between popular and less popular is simply set to the median of *item popularity*, i.e., only the items with popularity lower than the median can be recommended. Also, to better evaluate the performance of the Random Unpopular method, we repeat ten times of tests with different random seeds and report the mean results.

5) *Cluster-Based CF (CBCF):* This method is a variant of the prod2vec-cluster method proposed in [60]. Since the data format of the testing data is not literal, instead of using the prod2vec model to find vector

6) *Personal Innovator Probability (PIP) [13]:* It is a typical serendipitous recommendation method that uses an ergodic Markov chain to model how innovators are followed through multiple steps and hence obtain the PIP for each user pair. Finally, the PIP is used to rank items and form the users' recommendation lists.

The parameters in the proposed INVBCF method are set as discussed before and the best parameters in the other methods are tuned as suggested by the authors. The comparison results are shown in Table V.

2) Accuracy: Overall, since the testing set covers seven days, the recommender system is unable to capture the latest happenings for users and hence the accuracy tends to be lower than predicting in the single day manner. On one hand, from the experimental results we can see that introducing serendipity does influence the accuracy. To help cold items get involved in recommendation list, there is a tradeoff between accuracy and serendipity. On the other hand, we can also see that the simplest most popular method actually works quite well on this dataset if we only consider accuracy. This fact again confirms that the majority of users can only discover top items by themselves and hence we should introduce serendipitous recommendation so that users can discover items that they cannot discover without the help of recommender system.

Generally speaking, with the exception of MostPop and RandUnpop, all other methods perform reasonably close on accuracy. The CBCF method performs the best on accuracy since it well captures the significance of time correlation. It is slightly better than the proposed INVBCF method and the traditional UBCF method. The traditional

TABLE V
COMPARISON RESULTS IN TERMS OF DIFFERENT EVALUATION MEASURES

Metrics	INVBCF	UBCF	IBCF	MostPop	RandUnpop	CBCF	PIP
Precision	0.0738	0.0744	0.0705	0.0582	0.0110	**0.0744**	0.0688
Recall	0.0746	0.0752	0.0713	0.0589	0.0111	**0.0752**	0.0695
AD	**2.8945**	1.9058	1.5077	2.5574	2.6000	2.4649	2.7621
AvgDistance	**0.7060**	0.5541	0.5495	0.2463	**0.7202**	0.4972	0.5890
AvgPopularity	**0.1509**	0.2029	0.1891	0.6350	**0.1013**	0.2087	0.1616
Coverage	**0.3694**	0.2573	0.2239	0.0040	**0.4750**	0.1950	0.2838

TABLE VI
CASE STUDIES ON NEW ITEMS AND NICHE ITEMS

Types	Item IDs	INVBCF Rec	INVBCF Hit	UBCF Rec	UBCF Hit	IBCF Rec	IBCF Hit	MostPop Rec	MostPop Hit	RandUnpop Rec	RandUnpop Hit	CBCF Rec	CBCF Hit	PIP Rec	PIP Hit
New	2[illegible]3 23[illegible]7621 3[illegible]567[illegible]	√	√	√ ×	× × ×	× × ×	× × ×	× × ×	× × ×	√ ×	× × ×	× ×	× × ×	√	√
Niche	36217[illegible]33 3[illegible]1133[illegible] [illegible]7655171		√	× × ×	× × ×	√ × ×	× × ×	× × ×	× × ×	× ×	× × ×	× × ×	× × ×	× ×	× ×

√

also demonstrates the effectiveness of the proposed method. items into recommendation list.

IBCF method works slightly worse than the aforementioned methods since it is more personalized while online buyers are more socialized in China according to our user survey described in Section III. The PIP method works a bit less well on accuracy since it focuses much on improving serendipity. The MostPop method works poorly on accuracy and the RandUnpop method performs the worst on accuracy.

3) *Serendipity:* From the table we can see that both the proposed INVBCF method and the PIP method do help users improve the probability of discovering interesting item earlier. They can recommend proper items to users with the AD time about three days. We can also see that the RandUnpop method has the third highest AD but it is not reliable according to its poor accuracy. The MostPop method has the fourth highest AD and this is not because it provides good serendipity but because people actually interact with popular items every day. The CBCF method works slightly worse than the MostPop method. Finally, the UBCF method and the IBCF method generate the worst results in terms of AD.

Besides the AD time, to fully evaluate serendipity, we should also compare the difference between the recommendation result and user historical interests, i.e., AvgDistance. From the table we can see that the RandUnpop method wins on the AvgDistance metric since it only recommends less popular items. But we can also see that the proposed INVBCF method works very well on the AvgDistance metric compared to the other recommendation methods. The PIP method works worse on this metric which metrics too, with little effect on precision and recall. From the coverage prospective, it successfully introduces 10% more items than the UBCF and IBCF methods. As a result, it can better broaden user horizons. From the novelty perspective, the AvgPopularity of the recommendation list generated by the proposed method is much lower than the recommendation lists generated by the UBCF and IBCF methods. The PIP method works quite well on novelty and coverage too but is worse than the proposed method. The CBCF method works worse than the UBCF and IBCF methods since it suffers from a problem that popular items tend to be in the same cluster and items in this cluster have the largest probability of being purchased after purchasing items from other clusters. That is to say, the probability of recommending popular items is much larger than recommending less popular items. Overall, we can say that the proposed INVBCF method successfully introduces cold

IEEE Transactions on Cybernetics (Vol: PP, June 2018)

The UBCF, IBCF, and CBCF methods have similar performance on the AvgDistance. The MostPop method is the worst since most of the users are already familiar with popular items and popular categories.

According to the above analyze, a safe conclusion can be drawn that the proposed INVBCF method does help users improve the probability of discovering their potential interests earlier. Besides, it also helps users to have more access to unfamiliar items (both new items and niche items) and therefore broadens their horizons. Therefore, the above comparison results have confirmed the effectiveness of the proposed method in terms of serendipity.

4)Novelty and Coverage: From the table, there is no doubt that the RandUnpop method wins on both novelty and coverage at the expense of precision and recall, which are extremely low. However, we see the proposed INVBCF method has achieved good performance on these two

F. Case Studies

Although we have used AD and AvgDistance to verify the superiority of the proposed method on improving serendipity.

To better understand the capability of our method in discovering *new* items and *niche* items, we further conduct two case studies. The results are shown in Table VI. Note that in this paper, new items refer to items that are released less than one day and niche items refer to items that are released more than a week but have low item popularity, i.e., item popularity smaller than or equal to 2 according to Table I. If an item is included in recommendation lists, the entry below the corresponding Rec column is filled with √, and × otherwise.

Similarly, if an item is recommended accurately, the entry below the corresponding Hit column is filled with √, and × otherwise.

From Table VI we can see that, the proposed INVBCF method is the best, while the existing PIP method also works pretty well on discovering new items. Specifically, new items are not only included in recommendation lists but also recommended to the right person. Although the other methods, namely, the UBCF method, the RandUnpop method, and the CBCF method are also able to discover new items, they fail to recommend new items accurately. In addition, the IBCF and MostPop methods can hardly discover new items. We can also see that, the proposed INVBCF method outperforms other methods on discovering niche items and recommending them accurately. The PIP method recommends accurately but cannot discover as many niche items as INVBCF. Although the IBCF method and the RandUnpop method are also able to discover niche items, they fail to recommend niche items accurately. In addition, the UBCF, MostPop, and CBCF methods can hardly discover niche items.

Additionally, we also investigate the transformation between innovators and normal users in each timeframe (say every day from December 9, 2014 to December 18, 2014). As shown in Fig. 12, about 90 innovators degenerate into normal users and 100 normal users evolve into innovators in each timeframe. It is worth pointing out that there is no degenerated innovator reverting back to innovator again in the testing dataset. This is mainly because the transformation between innovators and normal users is a process from quantitative change

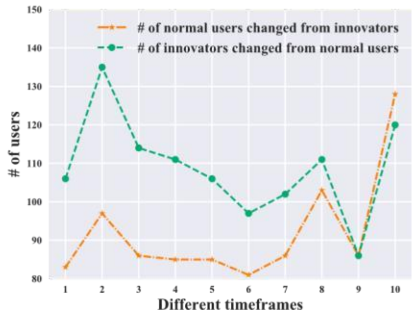

Fig. 12. Transformation between innovators and normal users.

to qualitative change. That is, with continuously interactions with the system, a normal user becomes senior and gradually learns how to discover new and niche items. Meanwhile an innovator degenerates to a normal user since he or she interacts with the system less frequently or is caught up by normal users.

VI. CONCLUSION

In this paper, we have addressed the serendipity issue of recommendation by developing a novel recommendation method termed INVBCF based on the user survey on the online shopping habits in China. In particular, by introducing the concept of innovators, new items and niche items can be introduced into the recommendation list and hence achieving the balance between serendipity and accuracy. For efficiently catching the needs of users accurately in real time, an offline component and an online component have been designed, which also saves communication cost and computing resources. The experimental results show that our proposed method beats other methods on serendipity while maintaining good performance on accuracy, novelty and coverage as well.
In our future work, we plan to investigate the social structure in discovering innovators provided that the social structure is given. Even in the case when there is no explicit social structure, it can be constructed where nodes identify users while edges specify that two users have interacted with the same items. In this way, if the PII of one user is below the threshold but the majority of its neighbors are innovators, this user is taken as an innovator according to the theory of the strength of weak ties [48]. Therefore, the innovators are not only determined by the PII but also affected by the neighbors with the weak ties.

REFERENCES

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

[2] S. Deng *et al.*, "A recommendation system to facilitate business process modeling," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1380–1394, Jun. 2017.

[3] S. Pyo, E. Kim, and M. Kim, "LDA-based unified topic modeling for similar TV user grouping and TV program recommendation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1476–1490, Aug. 2015.