User Identification within a Shared Account: Improving IP-TV Recommender Performance

Zhijin Wang, Yan Yang, Liang He, and Junzhong Gu

Department of Computer Science and Technology,
East China Normal University,
Shanghai, China
zhijin@ecnu.cn, {yangyan,lhe,jzgu}@cs.ecnu.edu.cn

Abstract. Multiple users share a common account in Internet Protocol Television (IP-TV) services. Can such shared accounts be identified solely on the basis of logs recorded by set top boxes (STBs)? Once a shared account is identified, can the different users sharing it be identified as well? We suppose different users within a shared account not only have different preferences for TV programs, but also get used to consuming services in different periods (e.g., after dinner or at weekend). We propose an algorithm to decompose users in composite accounts based on mining different preferences over different periods from consumption logs. In our experiments, the proposed algorithm outperforms traditional user-based collaborative filtering method 3-8 times when leveraging the decomposed users for personalized recommendation.

Keywords: User identification, Shared account, IP-TV recommendation, Experimentation.

1 Introduction

Internet Protocol Television (IP-TV) [10,16] services have been widely consumed in our daily life. It's a common phenomenon that families or roommates share television programs after they get back home or dormitory from work. With the development of IP-TV technologies, more and more multimedia resources (e.g., channels, programs and videos) are integrated into the services. In order to retrieve the preferred programs efficiently, recommender systems [1] are introduced into the services for IP-TV recommendations.

User experience can be improved when recommender systems are introduced into the IP-TV services. However, user identification is one of the most challengeable problems which degrades recommender performance [4]. The use of a single account by multiple users poses a challenge in providing accurate personalized recommendations. Informally, the recommendations provided to a shared account, comprising the ratings of two dissimilar users, may not match the interests of either of these users [21].

We aim to improve recommender performance by addressing the challenge of user identification in IP-TV services. According to a log recorded by a STB, a log contains: account id, program id, start time, end time, and genre(s).

Y. Manolopoulos et al. (Eds.): ADBIS 2014, LNCS 8716, pp. 219–233, 2014.

The user information is unavailable since the services are indistinctly shared by the users in a shared account. What's more, the interaction between user and set top box (STB) is very weak. Therefore, none of individual user informationcan be directly used for recommendation.

We suppose users within a shared account not only have different preferences for programs, but also get used to consuming services in different periods (e.g., after dinner or at weekend). There are two questions: (1) How to capture user preference over period accurately? (2) How to identify users based on the captured user preference over period?

To address these questions, we define: (1) a period which consists of non-overlapping sub-period(s); (2) a user who consumes services during these sub-periods. Based on these definitions, we propose an algorithm to carry out identification task.

To summarize, our contributions are as follows:

1. To the best of our knowledge, we are the first to study user identification as a problem of identifying preference and consumption time within a shared account.
2. We consider that user consuming behavior is periodic in IP-TV services, and a continuous period consists of non-overlapping sub-periods. Hence, the account preference over period can be captured by leveraging the designed implicit rating technique in the $\{account \times item \times time\}$ 3-dimensional space [2].
3. Based on the captured preferences, we design a user identification algorithm to identify users which mainly includes virtual user split stage and virtual user merge stage.
4. Finally, we demonstrate how the methods above can be applied to improve recommendation. We also study the effects of user splitting and user merging on recommendation.

The rest of the paper is organized as follows. Section 2 provides a brief review of related work on user identification and IP-TV recommendation. Section 3 describes our proposed approach to carry out identification task. Section 4 shows the settings in our experiments. Section 5 presents the experimental results. Section 6 analyses the users identification and recommender performance. Finally, we conclude in Section 7.

Table 1. Symbols

Symbol	Description
A, I, U	account set, item set, identified user set
P	the continuous period
p_k	the sub-period within P
v_{ak}	the virtual user within account a in sub-period k
s_h	several sub-periods within P
u_{ah}	the h-th identified user within account a
$u_{a\cdot}$	the identified users within account a
$\|A\|$	number of all accounts
$\|I\|$	number of all items
$\|U\|$	number of all identified users
$\|P\|$	number of all sub-periods
G	similarity graph of virtual users
d_{ait}	the duration of item i consumed by account a in sub-period t
r_{ait}	the implicit rating of account a to item i in sub-period t
S_{aij}	the preference similarity between virtual user i and j within account a
ρ	the threshold of preference similarity between two virtual users

Table 1 gives the main symbols used in this paper.

2 Related Work

In this section, we introduce related work on user behavior in IP-TV services at first, and then describe the current research on user identification for recommendation which related to our proposed algorithms.

2.1 User Behavior in IP-TV Services

User behavior data can be categorized into implicit feedback [14], explicit feedback [5], and their combination [15]. Unlike searching webs with strong intentions [3,7] (e.g., click links, type text or speech input in search bars), user behavior in IP-TV services is often implicit or unconscious.

One of the most common used implicit rating technique in IP-TV services is binary rating $r_{ui} \in \{0, 1\}$(e.g., [8,20]). But it can't capture the difference of duration within a shared account, hence, the percentage of play time the account has watched is exploited to capture account preference which we will further discuss in 3.2.

In the aspect of temporal features, user consuming behavior can be divided into long-term & short-term behavior [6] and periodic [12] behavior. We suppose that user consuming behavior (temporal feature) in IP-TV services is periodic, and the preferences for programs may change as time goes by.

2.2 User Identification for Recommendation

It's still a challenge to identify users of a STB since the STB is typically used by multiple users, e.g. family members or roommates. The issue of user identification within a shared account has received attention only recently.

Zhang et al. studied the user identification as a subspace clustering problem at [21], a composite account was regarded as a union of linear subspaces and used subspace clustering for carrying out identification task. They applied EM, GPCA and other clustering algorithms to identify users who belong to the same account on CAMRa2011 [18] dataset and Netflix [5] dataset. Their target was to label accounts to users. In contrast, we are trying to decompose users within a shared account.

Said et al. at [19] regarded time as a type of contextual information, and used it to split account into sub-profiles which is a method of recommending movies to specified users. A single user profile was split into several, possibly overlapping, contextual sub-profile (home and cinema sub-profiles as presented) in contextual pre-filtering stage. The split sub-profiles were integrated into recommenders in contextual post-filtering stage. However, they didn't consider incorporating overlapping contextual sub-profile into one that often decreases recommendation performance somehow which we will discuss in section 5.

3 Our Proposed Approach

In this section, we illustrate our proposed approach. The approach contains problem definition, notations and user identification algorithm named VUI.

3.1 Problem Definition and Notations

To start with, let us consider a common scene that multiple users share a common account in IP-TV services. As figure 1 shows, an account corresponding to a STB shared by 3 kinds of family members: senior, younger and kids. The senior get used to demanding history series in the morning or afternoon, kids would like to play the sort of cartoon programs after school or dinner, and younger might prefer films after kids go to the bed.

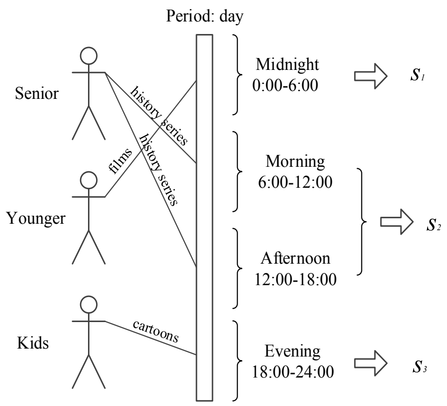


Fig. 1. An example of users sharing an account

From the common scene above we can find: (1) the consuming behavior is periodic; (2) different users get used to consuming the services in different part(s) of a period; (3) different users often have different preferences for programs (genres) provided by the services. Based on this common phenomenon, the problem of identifying users sharing a common STB can be regarded as distinguishing preference over period.

We introduce period P to describe periodic behavior. The period is defined as:

$$P := \bigcup_{k=1}^{|P|} p_k, \emptyset := \bigcap_{k=1}^{|P|} p_k \tag{1}$$

this definition means that a continuous period P (e.g., week) consists of several, non-overlapping sub-periods p_k (e.g., day of week). A user consumes services in more than one sub-periods, therefore the user is defined as:

$$u_{ah} := \{a, s_h | a \in A \cap s_h \subseteq P \cap s_h \neq \emptyset\} \tag{2}$$

where u_{ah} is the h-th user within account a who consumes services at s_h (e.g., Saturday and Sunday) in period P, A denotes all accounts in the system. As figure 1 provides, the senior consume services at s_1, the younger consume services at s_2, and kids s_3. Hence, all users U in system can be defined as:

$$U := \bigcup_{a \in A} \bigcup_{s_h \subseteq P} u_{ah} \tag{3}$$

Our goal is to identify U, and providing accurate recommendations for A by means of U. We are trying to reach the goal by addressing the two questions: (1) How to capture the preferences of user u_{ah} (h-th user within account a)? (2) How to determine the consuming time s_h of h-th user within each account?

In order to capture the preferences of user u_{ah}, we introduce virtual user v_{ak} to present activities of an account in a sub-period. The virtual user is defined as:

$$v_{ak} := \{a, p_k | a \in A \cap p_k \in P \cap p_k \neq \emptyset\} \tag{4}$$

where v_{ak} means the activities of account a in sub-period k, and an identified user is a composite of virtual user(s). Therefore, the user preference can be composed by the preferences of corresponding virtual users. The problem of determining the consuming time s_h of h-th user within account a is equivalent to assigning virtual users to identified users within an account.

We suppose that users in reality have different preference for both programs (or genres) and periods. Hence, the users in an account can be identified by combinations of virtual users. In order to study how the combinations affect identified users and recommendation performance, we introduce the similarity graph G, which uses vertexes to denote virtual users, and uses edges to denote the similarity between virtual users. The similarity graph G is defined as:

$$G := G(v_{a\cdot}, s_{a\cdot\cdot}) \tag{5}$$

where $v_{a\cdot}$ presents all virtual users within account a, $s_{a\cdot\cdot}$ presents similarities among all virtual users.

3.2 Algorithm for User Identification

In this section, we illustrate the algorithm for identifying users within a shared account. Detail steps of Virtual user based User Identification algorithm (VUI) are given in Algorithm 1.

Implicit Rating Technique in 3d Space. We adopt time concerned 3-dimensional space $\{account \times item \times time\}$ to present account preference over period (or sub-periods). As figure 2 shows, the coordinates denote accounts A, items I and period P, the symbols (e.g., triangle, square, and star) in it means the corresponding programs consumed over period. And items present programs or genres. We formulate the implicit rating as follows:

$$r_{ait} = \frac{\exp(d_{ait})}{\sum_{t \in P} \exp(d_{ait})} \tag{6}$$

Algorithm 1. Pseudo code of VUI to identify users within shared accounts.

Input:
A, All accounts; I, All items; P, All sub-periods;
D, Duration of accounts to items.
Output:
U, All identified users.

// Implicit ratings R
for each account a, item i, sub-period t in A, I, P **do**
  $r_{ait} \leftarrow$ calculate implicit rating by means of d_{ait}.
end for
for each account a in A **do**
  // Split
  VirtualUsers V;
  for each sub-period p_k in P **do**
    $V.append(v_{ak})$;
  end for
  SimilarityGraph G;
  for each virtual user i in V **do**
    for each virtual user j in V **do**
      $S_{aij} \leftarrow$ calculate preference similarity between i and j by means of implicit rating R;
      $s_{aij} \leftarrow S_{aij}, \rho$ // Threshold ρ
      if $s_{aij} == 1$ **then**
        $G.insert(vertex(i), vertex(j), s_{aij})$;
      end if
    end for
  end for
  // Merge
  for each vertex v in $G.vertexes$ **do**
    if $v.visited == false$ **then**
      $v.visited = true$;
      $u_{ah}.add(v)$; // Add v to h-th user;
      Get list of vertexes that connect to v as L;
      Visit each vertex in L and add to u_{ah} recursively;
      $h = h + 1$;
    end if
  end for
  $U.add(u_{a\cdot})$; // Add identified users within account a
end for
return U;

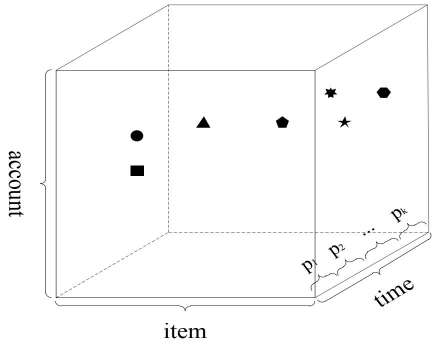


Fig. 2. $\{A \times I \times P\}$ 3d space

where d_{ait} is the duration of item i consumed by account a in sub-period t. Note that an account may demand an item more than one time, therefore, we do not choose the binary implicit rating techniques [8,20] or percentage of a program watched to the length of it [17]. The reasons are: (1) Accuracy. The binary implicit rating technique can describe a user has scanned a video, but it can't describe the degree of how the program is preferred. (2) Without length of programs. The provided dataset does not have the length of a file in terms of showing time of programs.

Split of Virtual Users. The split of virtual users is the process of determining sub-periods. We try two different methods to determine sub-periods. One is (1) empirical split method. In this method, the sub-periods are assigned by experience. The other is (2) average split method. The method split P in k equal length sub-periods. The split of virtual users is described at line 7-10, Algorithm 1.

Similarity Measurement between Virtual Users. Once virtual users in an account are obtained, the similarities among them can be measured, we adopt cosine method to measure the similarity between two virtual users since it's widely used. The similarity between virtual user v_{ai} and virtual user v_{aj} is defined as:

$$\begin{aligned} S_{aij} &= \cos(v_{ai}, v_{aj}) \\ &= \frac{\sum_{k \in I(v_{ai}) \cap I(v_{aj})} r_{aki} \cdot r_{akj}}{\sqrt{\sum_{k \in I(v_{ai})} r_{aki} \cdot \sum_{k \in I(v_{aj})} r_{akj}}} \end{aligned} \tag{7}$$

where r_{aki} denotes items k consumed by account a in sub-period i, and $I(v_{ai})$ denotes items set consumed by virtual user v_{ai}. The similarity measurement is used at line 14, Algorithm 1.

Similarity Threshold $\boldsymbol{\rho}$. In order to control the process of combinations of virtual users in the similarity graph, we use parameter $\rho \in [0, 1]$ to threshold the similarities of virtual users, and introduce $s_{aij} \in \{0, 1\}$ to present similarity. The binary similarity s_{aij} is defined as:

$$s_{aij} = \begin{cases} 1, S_{aij} \geqslant \rho \\ 0, otherwise \end{cases} \tag{8}$$

where S_{aij} means the similarity of virtual users measured by cosine method, and ρ denotes the similarity threshold. This means if the similarity of two virtual users is greater than ρ, the virtual users are regarded as similar, otherwise not similar.

We add similar virtual users to similarity graph if their binary similarity is equal to 1. The thresholding process is described at line 15-18, Algorithm 1.

Merging of Similar Virtual Users. According to steps above, the similarity graph can be obtained. Once a similarity graph is generated, the users can be identified by merging similar virtual user.

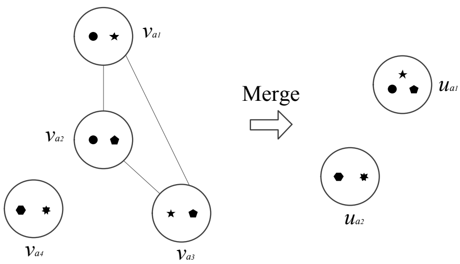


Fig. 3. An example of merging virtual users

As figure 3 shows, an example of merging similar users on similarity graph, the connected vertexes are merged to one as an identified user. In a word, we adopt deep-first-search (DFS) algorithm to carry out the merging task. An alternative way to carry out the task is bread-first-search (BFS) algorithm. The merge operation is described at line 22-31, Algorithm 1.

4 Experimental Setup

In this section, we illustrate dataset collection, evaluation metrics and algorithms for recommendation. We evaluate algorithm VUI on the dataset collected from the content provider SMG[1] in Shanghai, China. It should be noted that we focus on the evaluation of recommender performance by means of identified users, rather than the accuracy evaluation of algorithm VUI.

[1] `http://www.smg.cn/`

4.1 Dataset Collection

The logs in the services from SMG are during the period between March 1, 2011 till March 31, 2011. A log describes an account consumed a movie as well as genre, also the start time and end time of the services. We filter out logs of play time (calculated by start time and end time) less than 10 minutes. It contains 376,038 records, 5,933 videos categorized into 66 genres consumed by 14,856 accounts after being filtered. The records before March 25, 2011 are used for training, and the rest are as test set.

In order to avoid problems related to cold start, for both accounts and items, we decide that accounts in the evaluation sets have to consume at least 100 programs. We evaluate our results on a subset of 100 randomly selected accounts due to the long running time of the experiments when the full dataset is used.

4.2 Evaluation Metrics

We use Precision and Recall metrics to measure the performance of all the mentioned algorithm, since they often attract lots of attention in a running system and are well known. The Precision metric is defined as:

$$Precision@N = \frac{\sum_{u \in U} |R(u,N) \cap T(u)|}{\sum_{u \in U} |T(u)|} \tag{9}$$

where N denotes the length of a recommendation list, $R(u,N)$ denotes the recommendation list to user u with length N, $T(u)$ means items has been consumed by identified user u in test set. The Recall metric is defined as:

$$Recall@N = \frac{\sum_{u \in U} |R(u,N) \cap T(u)|}{\sum_{u \in U} |R(u,N)|} \tag{10}$$

From these definitions, we can see that a larger $Precision@N$ or $Recall@N$ indicates a better performance.

4.3 Recommendation Algorithms

We adopt, one of the most famous collaborative filtering methods, K-Nearest Neighbor (KNN) method to provide recommendations, since it performs very well in practice (e.g., [9,11,13]), and we can also learn the benefit from identified users by comparing with recommendations without identification.

The Cosine method is used to measure the similarity among accounts in algorithm KNN. For convenience, we name recommendations for accounts as AccountKNN, and recommendations for identified users as VUI-KNN, respectively.

The Contextual User Profile (CUP) method [19] is implemented and used to compare with VUI-KNN, since (a) the method regards an account consists of two contextual user profiles (home and cinema) by consuming time, which is similar to identify users but not, and (b) the authors also use KNN method to provide recommendations. We name recommendations according to contextual user profiles as CUPs.

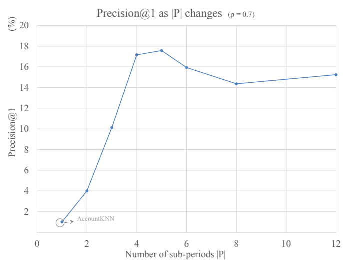
5 Experimental Results

In this section, we conduct several experiments to compare different parameters of VUI and different methods. Our experiments are intended to address the following questions:

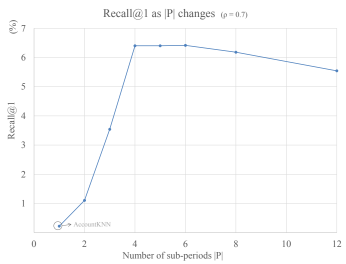
- How the parameters ($|P|$ and ρ) affect recommendations? In other words, how the assignment of sub-periods and combinations of virtual users affect recommendations?
- How the split methods affect recommendations?
- Can the KNN method take the advantage of VUI? Can the performance of VUI outperform that of CUPs?

5.1 Effects on Parameters $|P|$ and ρ

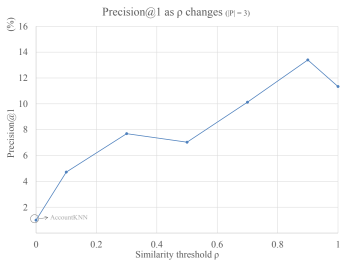
To study how the assignment of sub-periods and the process of combinations in similarity graph affect user identification and recommendations, we measure the performance in terms of precision and recall as $|P|$ or ρ change while holding other parameter. Here, we use average split method to assign equal length sub-periods. Note that, when $|P| = 1$ or $\rho = 0$, the VUI regards an account as a user, the AccountKNN algorithm is obtained.



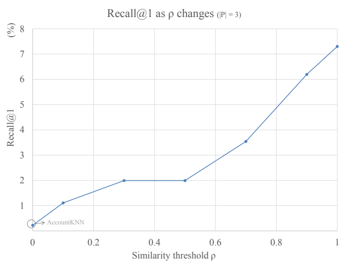
(a) Precision@1, change $|P|$



(b) Recall@1, change $|P|$



(c) Precision@1, change ρ



(d) Recall@1, change ρ

Fig. 4. The effect of $|P|$ and ρ on results

We fix ρ at 0.7 and change $|P|$ to measure precision and recall when making only one genre recommendation. As shown in figure 4(a) and 4(b), (1) the precision and recall value is significantly improved, when comparing with AccountKNN ($|P| = 1$, Precision: **1%**, Recall: **¡1%**); (2) the two optimal values are obtained at $|P| = 4$ and $|P| = 5$, and the precision value is slight over **17%**; (3) the performance starts degrading when $|P| > 5$.

To study effects on similarity threshold ρ, we hold $|P|$ at 3 (according to experience), and measure the performance of making one genre recommendations. Figure 4(a) and 4(b) reveals, (1) the recall value is still increasing as ρ increases; (2) the optimal precision is **13%**, ρ corresponding to 0.9;

5.2 Empirical Split versus Average Split

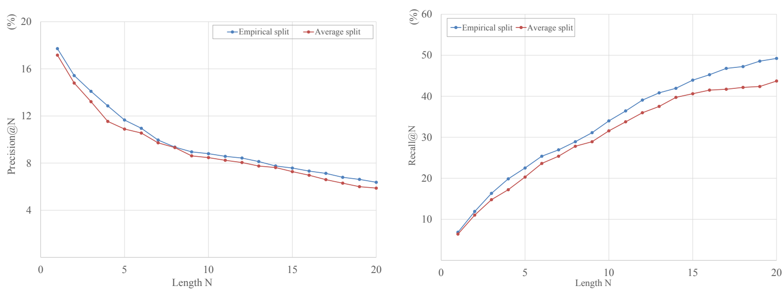
To study how split methods affect recommendations, we compare the designed sub-periods with the equal length sub-periods in terms of recommender performance.

According to the conducted experiments above, the optimal values are obtained at $|P| = 4$ and $|P| = 5$ (a slight better). We set $|P| = 4$ to compare the split methods, since it's more easier to empirically split up sub-periods than $|P| = 5$. The split sub-periods are shown in table 2, the differences are the end edge of afternoon and evening.

Table 2. The split up sub-periods ($|P| = 4$)

	Midnight	Morning	Afternoon	Evening
Average	0:00-6:00	6:00-12:00	12:00-18:00	18:00-24:00
Empirical	**23:50**-6:00	6:00-12:00	12:00-**19:00**	**19:00-23:50**

Recommendations are provided according to VUI-KNN, we compare the results on top-N recommendation as the length of recommendation list N changes.



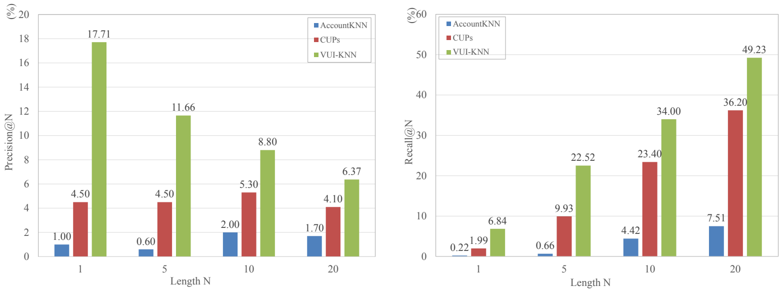
(a) Comparison of Precision@N (b) Comparison of Recall@N

Fig. 5. Comparison of empirical split method and average split method

As figure 5(a) and 5(b) states, the empirical split method gains a slight improvement when comparing with the average split method, but the improvement is not stable. A possible reason for the improvement is that users are off work after 18:00, they need to spend time on the way and can't receive programs immediately. The benefit of average split method is its simpleness and can be applied automatically. Actually, we used the average split method to carry out the split mission when $|P|$ is greater than 4.

5.3 Comparing with CUPs

The CUPs is configured as: (1) Each account is regarded as two context user profiles (Morning and Afternoon) by means of start time, context users are recognized as 'Morning' user if they consume items before 12:00, and 'Afternoon' user if after 12:00; (2) Recommendations are obtained by KNN method as well as our proposed VUI-KNN. Note that, we implement CUPs on SMG, instead of Moviepilot[2] dataset.



(a) Precision@N (b) Recall@N

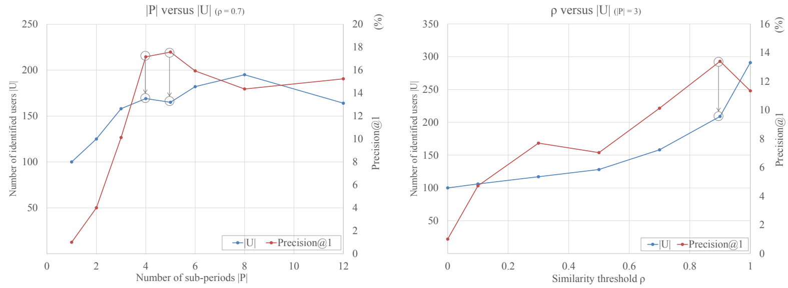
Fig. 6. Comparison of methods in terms of $Precison@N$ and $Recall@N$ with $N = \{1, 5, 10, 20\}$

As stated in figure 6(a) and 6(b), the effects of 3 comparable methods mentioned in section 4.3, (1) VUI-KNN outperform CUPs by about 1.5-3 times, the increase becomes slow when N grows; (2) CUPs runs better than AccountKNN by about 2-3 times, this improvement is closed to [19] as well; (3) for AccountKNN, the effects of precision with $N = \{1, 5\}$ are worse than that with $N = \{10, 20\}$, but the recall is still increasing, a possible reason is the recommendations provided to the accounts not match the interests of either of these users, the mismatch is decreasing when N increases to a proper value which is between 5 and 10.

6 Analysis and Discussion

In this section, we analyze how the parameters affect the number of identified users, and also how the identified users affects recommender performance. According to these analyses, we discuss about the limitations of the proposed algorithm VUI.

[2] `http://www.moviepilot.de`



(a) $|U|$ versus $|P|$ (b) $|U|$ versus ρ

Fig. 7. Number of identified users $|U|$ and Precision@1 as parameters change

6.1 Identification and Performance Analysis

We try to discover the relationship between the number of identified users and the recommender performance. The experimental results are plotted in figure 7, and the performance is in terms of precision.

As figure 7(a) reveals, (1) the precision value increases when users are identified; (2) the two optimal precision values are found at $|U| = \{165, 169\}$; (3) the precision starts decreasing when $|U| > 165$. The optimal precision value is found at $|U| = 294$ in figure 7(b). According to figure 7(a) and 7(b), the optimal precision value is between $|U| = \{165, 169\}$.

The following summarizes the key conclusions we observe from the results: (1) The recommender performance is improved when users within accounts are identified for personalized recommendations. A reason for the improvement is that recommendations based on identified users alleviate the problem of recommending given item to wrong users within a shared account. (2) There exists a pair of P and ρ leading to the best performance of recommendations. In other words, less or more identified users ($|U|$ is too small or too large) will degrade the performance.

1. Less users identified, when P is split into few sub-periods and ρ is set very close to 1, which may regard two (or more than two) real users as one. Hence, a possible reason for the performance degrading is recommending items to who unlike them.
2. More users identified, when P is split into too many sub-periods and ρ is set very close to 0, which may regard a real user as two (or more than two) identified users, and the preferences of the real user are divided into several parts by the identified users. Hence, the opportunity of recommending right items to the real user may decrease since the KNN recommends items other users also preferred.

The best pair of P and ρ found in the pervious section reflects user consuming behavior in IP-TV services. when $|P|$ is set to 4, P consists of the four sub-periods: Midnight (0:00-6:00), Morning (6:00-12:00), Afternoon (12:00-18:00) and Evening (18:00-24:00), and $\rho = 0.7$, we get the best performance.

It also means that, a user has his/her own preference in sub-periods when consuming the services.

6.2 Discussion

Multiple users share a common account in IP-TV services, in order to recommend right items to right users within these shared accounts, we try to identify users for personalized recommendations.

We suppose that users have different tastes, thus different recommendations are required for them. The algorithm VUI is proposed to distinguish users by time and preference patterns. The recommendation performance is significantly improved by the identified users.

The performance of VUI is affected or decided by the parameters. We learn the parameters by cross-validation method in our experiments. But the parameters (e.g., P and ρ) can't be obtained automatically. The potential work is to learn the parameters, which can be regarded as learning user consuming behavior in terms of time and preference.

7 Conclusion and Future Work

In this paper, we define the problem of user identification as mining different preferences over different periods from consumption logs. According to this definition, an algorithm for user identification is proposed to predict users within a shared account in IP-TV services. The process of user identification consists of two phases. The first is to partition a day and identify behavior specific to different periods. Secondly, periods for which discovered usage patterns are similar are regarded as associated with the same actual user. The association process is carried out by leveraging DFS algorithm in a similarity graph.

The predicted users are able to improve recommender performance in terms of precision and recall. The optimal precision value and recall value are obtained when $|P| = 4$ and $\rho = 0.7$, $|P| = 4$ also reflects the kinds of user consuming preferences in terms of periods, and the number of identified users corresponding to the optimal performance can also be found by the cross-validation method in the conducted experiments.

The evaluation of such methods is a potential future direction of this work.

Acknowledgments. We would like to thank SMG for sharing the IP-TV consumption logs, and the anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper. This work is supported by the Shanghai Science and Technology Commission Foundation (No.12dz1500205 and No.13430710100).

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17(6), 734–749 (2005)

2. Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Recommender Systems Handbook, pp. 217–253 (2011)
3. Ageev, M., Lagun, D., Agichtein, E.: Improving search result summaries by using searcher behavior data. In: SIGIR 2013, pp. 13–22 (2013)
4. Bambini, R., Cremonesi, P., Turrin, R.: A recommender system for an iptv service provider: a real large-scale production environment. In: Recommender Systems Handbook, pp. 299–331 (2011)
5. Bell, R.M., Koren, Y.: Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In: ICMD 2007, pp. 43–52 (2007)
6. Bennett, P.N., White, R.W., Chu, W., Dumais, S.T., Bailey, P., Borisyuk, F., Cui, X.: Modeling the impact of short- and long-term behavior on search personalization. In: SIGIR 2012, pp. 185–194 (2012)
7. Grasch, P., Felfernig, A., Reinfrank, F.: Recomment: towards critiquing-based recommendation with speech interaction. In: Recsys 2013, pp. 157–164 (2013)
8. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: ICDM 2008, pp. 263–272 (2008)
9. Katz, G., Ofek, N., Shapira, B., Rokach, L., Shani, G.: Using wikipedia to boost collaborative filtering techniques. In: Recsys 2011, pp. 285–288 (2011)
10. Kim, E., Pyo, S., Park, E., Kim, M.: An automatic recommendation scheme of tv program contents for (ip)tv personalization. TBC 57(3), 674–684 (2011)
11. Koren, Y.: Collaborative filtering with temporal dynamics. In: KDD 2009, pp. 447–456 (2009)
12. Li, Z., Wang, J., Han, J.: Mining event periodicity from incomplete observations. In: KDD 2012, pp. 444–452 (2012)
13. Liu, N.N., Zhao, M., Xiang, E.W., Yang, Q.: Online evolutionary collaborative filtering. In: Recsys 2010, pp. 95–102 (2010)
14. Ma, H.: An experimental study on implicit social recommendation. In: SIGIR 2013, pp. 73–82 (2013)
15. Pero, Š., Horváth, T.: Opinion-driven matrix factorization for rating prediction. In: Carberry, S., Weibelzahl, S., Micarelli, A., Semeraro, G. (eds.) UMAP 2013. LNCS, vol. 7899, pp. 1–13. Springer, Heidelberg (2013)
16. Pyo, S., Kim, E., Kim, M.: Automatic and personalized recommendation of tv program contents using sequential pattern mining for smart tv user interaction. Multimedia Syst. 19(6), 527–542 (2013)
17. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: Recommender Systems Handbook. Springer (2011)
18. Said, A., Berkovsky, S., Luca, E.W.D., Hermanns, J.: Challenge on context-aware movie recommendation: Camra2011. In: Recsys 2011, pp. 385–386 (2011)
19. Said, A., Luca, E.W.D., Albayrak, S.: Inferring contextual user profiles - improving recommender performance. In: Proceedings of the 3rd Workshop on Context-Aware Recommender Systems. IEEE (2011)
20. Xu, M., Berkovsky, S., Ardon, S., Triukose, S., Mahanti, A., Koprinska, I.: Catch-up tv recommendations: show old favourites and find new ones. In: Recsys 2013, pp. 285–294 (2013)
21. Zhang, A., Fawaz, N., Ioannidis, S., Montanari, A.: Guess who rated this movie: Identifying users through subspace clustering. In: UAI 2012, pp. 944–953 (2012)