FP-Growth-Based Temporal Association Mining Method for Smart Home User Control Habits Postprint

Authors: Liang Tiankai, Zeng Bi, Jianqi Liu

Date: 2018-12-13T00:00:00+00:00

Abstract

To address the inability of traditional association rule mining algorithms to efficiently and accurately mine sequential association control habits implicit in user operation records, this paper proposes a smart home user sequential association control habit mining algorithm based on FP-Growth. The algorithm comprises three stages that respectively generate transaction sets, sequential frequent itemsets, and final sequential association control habits based on a user control action forest, an improved FP-Growth algorithm, and a time constraint rule. Finally, comparative experiments conducted with real user operation records demonstrate that the proposed algorithm can enhance the efficiency of transaction set generation and more accurately discover users' sequential association habits in controlling smart home devices.

Full Text

Preamble

Vol. 37 No. 2
Application Research of Computers (ChinaXiv Partner Journal)
Accepted Paper

FP-Growth-Based User Temporal Association Control Habits Mining Method for Smart Home

Liang Tiankaia, **Zeng Bi**a, **Liu Jianqi**b
(a. School of Computer Science; b. School of Automation, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Traditional association rule mining algorithms cannot efficiently and accurately mine the temporal association control habits implicit in user operation records. To address this problem, this paper proposes a temporal association control habits mining algorithm for smart home users based on FP-Growth. The algorithm consists of three stages: transaction set generation, temporal frequent itemset generation, and final temporal association control habits generation, based on a user operation action forest, an improved FP-Growth algorithm, and a time constraint rule, respectively. Finally, comparative experiments using real user control records demonstrate that the proposed algorithm improves the efficiency of transaction set generation and can more accurately discover users' temporal association habits in controlling smart home devices.

Keywords: smart home; behavior prediction; data mining; association analysis; personalized recommendation

Chinese Library Classification: TP301.6
doi: 10.19734/j.issn.1001-3695.2018.07.0527

0 Introduction

With the leapfrog development of Internet of Things technology, computer network technology, and data mining technology, intelligence has become the new synonym for development trends in the new century. Against this backdrop, the development of smart homes has also shown explosive growth. A smart home system takes the individual residential space as its primary platform, connecting home devices to the network through IoT technology to enable remote control of home devices and build an efficient management system for residential facilities and family daily affairs. The intelligence level of smart home systems can be divided into three layers. The low-level intelligence only enables simple remote operation of smart home devices, where users send control commands via mobile terminals through wireless/wired networks to the smart home system' s control center, which then forwards the commands to the corresponding home devices via the home network. The medium-level intelligence enables trigger-based automated control of home devices based on environmental perception. For example, users can define a rule: "When the indoor temperature exceeds 30 degrees Celsius, turn on the air conditioner for me." Thereafter, the smart home system monitors the indoor environment through temperature sensors connected to the home network. When environmental changes reach the threshold that triggers the predefined rule (i.e., indoor temperature exceeds 30 degrees Celsius), the system automatically performs the corresponding control action (turning on the air conditioner). The highest level of intelligence requires the smart home system to have learning capabilities, enabling it to learn users' control habits for home devices from large volumes of user control records and autonomously control home devices under appropriate conditions on behalf of users, thereby achieving truly intelligent operation of home devices.

Various control behaviors of users toward smart home devices have certain connections and patterns. If such potential associative control habits can be identified from users' historical control records and used to develop smart home systems that better understand users, it will help the smart home industry evolve toward the highest level of intelligent systems. Currently, domestic and foreign scholars have achieved certain results in mining users' associative control habits. For example, reference [9] proposed using the Apriori association analysis algorithm to mine associative control habits implicit in users' historical interaction records. However, the associative control habit rules obtained by this method only simply express the relationships between users' various control actions and fail to fully express the temporal characteristics between user control actions, resulting in relatively low learning capability. Similarly, reference [10] proposed an association analysis algorithm based on hypothesis testing, which further proved that strong dependencies exist among home devices and can be effectively mined. To improve the noise resistance of association analysis algorithms for mixed and missing data, Cook et al. [11] utilized Episode Discovery (ED) algorithms to identify associative relationships implicit in users' control behaviors, which improved the algorithm' s ability to handle large-scale mixed data to some extent. To further enhance the processing capability for large-scale data, reference [12] proposed a double-layer ART1 pattern classification method based on ART (adaptive resonance theory) networks, establishing a mathematical model for the relationships between users' control actions while using cloud warehouses to store user control record data, effectively solving the problem of insufficient local storage resources for large-scale user control data. Although the above methods are significant and inspiring for the promotion of smart home applications and the improvement of smart home system intelligence levels, these algorithms still cannot guarantee that the rules within the mined user associative control habits have temporal ordering and strong temporal correlation, and thus cannot accurately mine users' potential temporal association control habits.

To solve the above problems, this paper proposes a Temporal Association Analysis Based on FP-Growth (TAABFPG) algorithm. The TAABFPG algorithm proposes using a forest structure to store user control records and generate transaction sets. Second, to accurately mine users' potential temporal association control habits with strong temporal correlation, a simple yet effective constraint rule is also proposed. Finally, this paper uses real user data provided by a smart home company to conduct comparative experiments with three classical and common association analysis algorithms, demonstrating the effectiveness and superior performance of the proposed algorithm.

---

1 Algorithm Framework

The TAABFPG algorithm proposed in this paper for mining smart home users' associative control habits based on FP-Growth can be divided into three stages: transaction set generation, temporal frequent itemset generation, and final user associative control habits generation. The algorithm flow is shown in Figure 1

[Figure 1: see original paper], with specific execution steps as follows:

a) **Transaction Set Generation:** Process user historical control data into action format, generate a user operation action forest, and finally obtain the transaction set by traversing the forest.

b) **Temporal Frequent Itemset Generation:** Based on the FP-Growth algorithm, this paper adds temporal processing improvements to the FP-tree generation process to enable it to produce temporal frequent itemsets that meet the minimum support threshold.

c) **User Associative Control Habits Generation:** This step includes temporal candidate association rule generation and final user control habit filtering. First, generate temporal candidate association rules that satisfy the minimum confidence threshold based on frequent itemsets, calculate the time constraint factor for candidate association rules, and then mine users' temporal association control habits with time constraints through the time constraint rule.

TAABFPG Algorithm Flow Description:

Input:
- `min(supp)`: Minimum support threshold for rules
- `min(conf)`: Minimum confidence threshold for rules
- θ: Time constraint coefficient between rule items (unit: minutes)
- `dataset`: User control records

Output: Temporal association control habits with time constraints

a) Build a user operation action forest containing several subtrees based on the generation time of user control records

b) Traverse the user operation action forest and generate the transaction set

c) Generate several temporal frequent itemsets that meet the minimum support requirement through the improved FP-Growth algorithm

d) Form temporal candidate association rules that meet the minimum confidence requirement based on temporal frequent itemsets

e) Add time constraint rules to filter temporal candidate association rules. The temporal candidate association rules that pass the filtering are the users' temporal association control habits that satisfy the time constraint of θ minutes

2 User Operation Action Forest and Transaction Set Generation

Users' historical control records are stored by day. If records from each day are simply composed into transactions, some associative relationships that exist when users control smart home devices will be lost. Therefore, how to convert user historical control data into a more reasonable transaction set is particularly important.

For user control records, this paper argues that user control actions of the same type with operation time intervals exceeding 30 minutes (ignoring operation dates) are often associated with different control actions. If not distinguished, incorrect associative control habits will be generated. For example: If a user likes to turn on the cooking machine at 8 AM to prepare breakfast and then turns on the radio to listen to the morning news; while at 6 PM, the user also habitually turns on the cooking machine to prepare dinner, followed by turning on the television to watch favorite shows while enjoying dinner. In this context, there should be two association rules for this user: {8 AM: turn on cooking machine → turn on radio} and {6 PM: turn on cooking machine → turn on television}. If the same control action at different times is not distinguished, associative control habit drift will occur, i.e., the algorithm calculates the average operation time for turning on the cooking machine as 13:00, which is far from the control actions of turning on the television around 8 AM and turning on the television around 6 PM in the time dimension, causing the loss of both associative control habits. Therefore, it is necessary to identify the control actions of turning on the cooking machine at 8 AM and 6 PM with operation time intervals exceeding 30 minutes as two different user control actions to avoid the above problem.

Second, to efficiently generate transaction sets based on user control records, this paper proposes using a forest structure to store user control records, called the user operation action forest, and provides a corresponding forest traversal method to efficiently generate transaction sets. The specific process is as follows:

a) Build an empty forest and traverse the user' s historical control records for n days by day, generating one subtree per day, ultimately forming a forest containing n trees.

b) Generation of daily subtrees: Build a subtree with an empty node, then traverse the control actions of the user control records in chronological order of action generation time, and insert each action after the rightmost leaf node of a branch under the root node that does not already contain the same node. If all branches under the root node contain this action, store the control action as the right child node of the root node.

c) Traverse the action subtrees in the forest separately to generate the transaction set: Each branch under the root node of each subtree in the forest corresponds to one transaction. The multiple transactions generated from multiple branches of multiple trees ultimately form the transaction set for the user control records.

From a theoretical perspective, the time complexity of storing and generating transaction sets through the proposed method of building a user operation action forest is superior to the traditional sequential list storage structure, as proven below:

Assume there are n user records generated by a user over m days. If using the traditional sequential list storage method, all elements of the sequential list need to be traversed. If an element does not exist in the constructed transaction list set, it is appended to the transaction list where the most recent different action resides; otherwise, a new transaction list is created with it as the header. Its time complexity is at the $O(n^2)$ level [13].

If using the forest method to generate the transaction set, the time complexity of building the user operation action forest containing m subtrees with a total of n nodes is at the $O(n)$ level [14], while the time complexity of traversing this forest is at the $O(n \log n)$ level. Therefore, from a theoretical perspective, it can be proven that the forest processing method is more efficient than the traditional sequential list method. Second, the subsequent experimental section will also set up corresponding verification experiments to prove this.

---

3 Temporal Frequent Itemset Generation

In association analysis, frequent itemsets represent rules that occur with high frequency. The support of a rule reflects its proportion in the transaction set. The support of rule $\{X \to Y\}$ is defined as the percentage of transactions in transaction set D that contain both user control actions X and Y (denoted as action XY), i.e., the probability [15]. Where $|x|$ represents the number of transactions containing user control action x, and $len(D)$ represents the total number of transactions in the transaction set [15].

This algorithm proposes generating frequent itemsets based on the FP-Growth algorithm. The FP-Growth algorithm mainly includes two parts: FP-tree construction and FP-tree mining [16]. However, the traditional FP-Growth algorithm only uses support as the basis for building the header table during FP-tree mining, so the resulting frequent items lack temporal ordering and cannot meet the requirement of smart home recommendation systems to mine users' temporal association control rules with time constraints. To solve this problem, this paper improves the FP-tree mining process by adding temporal operations to ensure that the generated frequent itemsets meet temporal requirements.

To more intuitively illustrate the frequent itemset generation process, assume there exists a transaction set $T = \{\{A, E, C, B\}, \{A, C, D, E\}, \{A, C, G\}, \{E, F, H\}, \{A, C, D, G\}, \{A, C, E, G\},$

3.1 FP-Tree Construction

Building an FP-tree mainly includes three steps: header table construction, transaction set reconstruction, and FP-tree generation [16]. The specific process is as follows:

a) **Header Table Construction:** Based on the item header set $I = \{A, B, ...\}$ of the transaction set, calculate the corresponding support of each item header, then filter out items that meet the minimum support threshold (min(supp)), sort them, and obtain the header table of the transaction set. The header table format is: $\{(D, sup(D)), (A, sup(A)), ...\}$. Using the assumed transaction set T as an example, the header table meeting the minimum support of 10% generated for transaction T is shown in Table 1 .

b) **Transaction Set Reconstruction:** Based on the processed header table, delete elements in each transaction that do not exist in the header table, and sort them according to the header table order. Using the assumed transaction T as an example, the reconstructed transaction set is shown in Table 2 .

c) **FP-Tree Generation:** Perform hierarchical insertion of the tree according to the order of the processed transaction set. The FP-tree node is defined as (x, n), where x is an element from the header table and n is the count of how many times this item header has appeared on this path so far, i.e., the count field.

Using the assumed transaction set T as an example, first build an empty root node and insert transaction 1 into the FP-tree, generating the FP-tree shown in Figure 2(a). Then insert transaction 2 into the FP-tree, generating the FP-tree shown in Figure 2(b). By analogy, the final FP-tree is generated as shown in Figure 3 [Figure 3: see original paper].

3.2 FP-Tree Mining

Mining the generated FP-tree can yield frequent itemsets that meet the minimum support requirement. However, the traditional FP-Growth algorithm only uses support as the basis for processing during FP-tree mining and cannot generate temporal frequent itemsets, failing to meet the needs of mining temporal association control for smart home users. To solve this problem, this algorithm proposes an improved FP-tree mining algorithm with temporal processing to mine temporal frequent itemsets. The specific steps are as follows:

a) According to the order of the header table, traverse the item header x in reverse order from back to front to obtain its corresponding conditional pattern base: traverse from the root node to the corresponding item header node according to the hierarchical structure of the FP-tree, and change the count field of the nodes passed. Finally, obtain its corresponding conditional pattern base in the form

$\{[(A, t_A), n_A], [(B, t_B), n_B], ..., [(E, t_E), n_E]\}_x$, where t_x is the average operation time of element x (i.e., user operation action) (ignoring operation date).

b) Calculate the support of elements in the conditional pattern base other than x, and delete elements that do not meet the minimum support requirement.

c) Perform temporal operations on the elements of the processed conditional pattern base to finally form the temporal conditional pattern base in the format: $\{[(A, t_A), n_A], [(B, t_B), n_B], ...\}_x$, where $t_A \leq t_B \leq ....$

d) Combine elements of the temporal conditional pattern base according to their original order to form several frequent items of length greater than 1 that contain x.

e) Iteratively repeat steps a) and b) until all item headers have been traversed, obtaining the temporal frequent itemset.

To better illustrate the frequent itemset generation process, take the FP-tree generated from transaction set T (shown in Figure 3) as an example:

1) For item header G, which is at the last position in the header table, generate its conditional pattern base.

2) Calculate the support of each element in the conditional pattern base except G: solving yields $\{(A : 43\%), (C : 43\%), (E : 14\%)\}$. All elements meet the minimum support requirement of 10% and are retained.

3) Perform temporal sorting on the filtered conditional pattern base: Assume $t_A < t_C < t_E < t_G$, then the temporally processed temporal conditional pattern base is: $\{[(A, 3), t_A], [(C, 3), t_C], [(E, 1), t_E], [(G, 3), t_G]\}$.

4) Combine elements of the temporal conditional pattern base in their original order to obtain the temporal frequent itemset P for this item header with length greater than 1 and containing G: $\{(A : 3, E : 1, G : 3), (A : 3, C : 3, G : 3), (E : 1, C : 3, G : 3), (A : 3, E : 1, C : 3, G : 3)\}$.

5) And so on, until all item headers have been traversed, generating all frequent items that meet the minimum support requirement.

---

4 Generation of User Temporal Association Control Habits

The generation of user associative control habits includes two main parts: temporal candidate association rule generation and obtaining the final user control habits through filtering of temporal candidate association rules. First, generate temporal candidate association rules that meet the minimum confidence threshold based on frequent itemsets, calculate the time constraint factor for candidate association rules, and then mine users' temporal association control habits with time constraints through the time constraint rule.

4.1 Generation of Temporal Candidate Association Rules

Traditional association rule analysis algorithms only use confidence as the screening standard in the process of generating final association rules, so the generated association rules lack strong temporal correlation. To ensure that the rule items in the final generated association rules have strong temporal correlation, this paper proposes a method using a dynamic time constraint rule based on the dynamic time constraint factor Δt to filter temporal candidate association rules. The specific filtering process is as follows:

a) Each temporal candidate association rule has a dynamic factor. If the rule's Δt is greater than the given time constraint coefficient θ between rule items, the rule is considered invalid and abandoned. Otherwise, proceed to the next screening operation.

b) If the rule is not invalid, use formula (4) to calculate the average time difference $t_{i,j}$ between the i-th and j-th rule items in the rule. Assume the rule has n rule items: $t_{i,j} = |t_i - t_j|$, where $1 \leq i < j \leq n$.

c) If $t_{i,j} > \theta$, break the rule chain after the i-th item and retain the first i items of the temporal candidate rule as the final rule to ensure the rule has strong temporal correlation and timeliness characteristics. Otherwise, continue traversing the next pair of rule items.

d) Continuously repeat steps b) and c) until all entries of the temporal candidate rule have been traversed.

e) Repeat steps a) to d) for all temporal candidate association rules until all temporal candidate rules have been traversed.

Temporal Candidate Association Rule Filtering Algorithm Flow Description:

Input:
- r: Set of temporal candidate association rules
- θ: Time constraint coefficient between rule items (unit: minutes)

Output: Temporal association rules with time constraints

1. For all temporal candidate association rules that have been traversed once:
    2. For $(i = 1, j = 2; j \leq$ number of rule items; $i++, j++)$:
        3. If $t_{i,j} > \theta$:
            4. Break the rule chain after the i-th item and retain the first i items as the final rule

    3. Return all rules that finally pass the screening

Association analysis can find relationships among data in large datasets, existing in the forms of frequent itemsets or association rules. In this algorithm, to better mine associative control habits that match user interests and are understandable to users, the algorithm must also calculate the confidence of each frequent itemset to generate association rules that meet the minimum confidence requirement.

The confidence of rule $\{X \to Y\}$ is the percentage of transactions in frequent itemset T where users continue to execute action Y directly or indirectly after executing action X (denoted as action Y|X), i.e., the conditional probability [17]:

$$conf(X \to Y) = P(Y|X) = \frac{sup(XY)}{sup(X)}$$

Taking the frequent itemset $P_i = \{A : 3, E : 1, G : 3\}$ generated for item header G in Section 3.2 as an example: It can be seen that the number of events where "actions A, E, and G occur simultaneously" is 1 as influenced by action E. According to the frequent itemset, the number of events where "actions A and E occur simultaneously" is 4, so the confidence of frequent itemset P_i can be calculated as:

$$conf(P_i) = \frac{P_i(AEG)}{P_i(AE)} = \frac{1}{4} = 0.25$$

If the minimum confidence is set to 0.2, then the frequent itemset can be considered to represent a valid temporal association rule $\{A \to E \to G\}$, which is identified as a temporal candidate association rule. The format of temporal candidate association rules is: $\{(A \to E \to G), (t_A, t_E, t_G), \Delta t\}$, where Δt is the time constraint factor, representing the average time difference between the average control time of the rule' s second item and the first item, which can be solved by formula (3):

$$\Delta t = \frac{1}{2}(t_2 - t_1) = t_2 - t_1$$

4.2 Filtering of Temporal Candidate Association Rules

---

5 Experiments

5.1 Experimental Environment

The verification experiments in this paper were run on a personal computer with 8 GB memory and an Intel(R) CoreTM i7-6770 CPU with a clock speed of 3.40

GHz, running the Windows 7 Professional operating system. All algorithms in this paper were implemented in the Python programming language.

5.2 Dataset

This section uses real user control records provided by a smart home company for comparative experiments. After deleting records containing missing values, the experimental data contains a total of 557,372 control records for a user over nearly one year (from September 29, 2017 to August 3, 2018), involving 9 smart home devices. According to the company' s data specification, the experimental dataset used in this paper contains 11 temporal association control habits with strong correlation. The details of the experimental data are shown in Table 3 .

Table 3 Experimental Dataset

Record Start Time	Record End Time	Smart Home Devices	User Potential Temporal Association Habits
September 29, 2017 00:00	August 3, 2018 15:22	Living room curtain, smart socket, living room light, front door, back door, infrared transmitter, smart switch, fan, air conditioner	1. Open front door → Open living room light → Turn on air conditioner cooling2. Turn on smart switch → Turn on smart socket → Turn on infrared3. Turn off air conditioner → Turn off living room light → Close front door4. Close living room curtain → Turn on hall light5. Turn off hall light → Open living room curtain6. Turn off smart switch → Turn off smart socket → Turn off infrared7. Turn on smart socket → Turn on main light8. Turn on air conditioner cooling mode → Turn on air conditioner dehumidification mode9. Turn off air conditioner → Turn on fan10. Open back door → Close back door11. Turn off fan → Turn on air conditioner cooling mode

5.3 Experimental Results and Analysis

To verify that the transaction set generation method proposed in this paper is more efficient, 50 repeated experiments for generating transaction sets were conducted using both the user operation action forest method and the sequential list method. The runtime of each experiment for the two transaction set methods is shown in Figure 4 [Figure 4: see original paper].

Finally, by calculation, the average runtime for generating transaction sets using the two methods are: the user operation forest method averages 79.80998 seconds, while the traditional sequential list method requires an average of 120.41024 seconds to generate the same transaction set, representing an efficiency improvement of approximately 33.72%. The experimental results prove that the transaction set generation method based on the user operation action tree proposed in this paper can more efficiently generate transaction sets required for the association rule analysis stage.

Second, to verify that the algorithm proposed in this paper can more accurately mine users' potential temporal association control habits, repeated experiments were conducted using the traditional Apriori algorithm, traditional Eclat algorithm, traditional FP-Growth algorithm, and the TAABFPG algorithm proposed in this paper. The F1 value was used as the final evaluation standard for algorithm performance, which can be solved by formula (5) [18]:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Where $precision$ represents precision, which can be solved by formula (6); $recall$ represents recall, which can be solved by formula (7):

$$precision = \frac{|relevent \cap retrieved|}{|retrieved|}$$

$$recall = \frac{|relevent \cap retrieved|}{|relevent|}$$

Where $|\{relevent\}|$ represents the number of users' potential temporal association habits; $|\{retrieved\}|$ represents the number of rules mined by the algorithm (users' temporal association control habits); $|\{relevent\} \cap \{retrieved\}|$ represents the number of valid rules among the rules mined by the algorithm that belong to users' potential temporal association habits.

The repeated experiments were arranged as follows: Set the minimum support threshold range to $\{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9\}$, and the minimum confidence threshold range to $\{0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9\}$. Then randomly combine minimum support and minimum confidence without repetition in sequence, forming 49 pairs of $\{minimum\ support, minimum\ confidence\}$ combinations such as $\{0.6, 0.6\}$. Finally, use the four algorithms to conduct 49 experiments with different minimum support and minimum confidence thresholds.

Table 4 Algorithm Performance Indicators (Mean Values)

Algorithm	Rules Count	Valid Rules Count	Precision	Recall	F1 Value
Apriori	15.469	11.239	72.686%	93.69%	0.818
ECLAT	15.469	11.240	72.701%	93.68%	0.818
FP-Growth	15.469	11.240	72.694%	93.71%	0.818
TAABFPG	13.449	11.376	84.579%	94.79%	0.894

As shown in Table 4, the indicators of the Apriori algorithm, Eclat algorithm, and FP-Growth algorithm are almost identical, indicating that if mining efficiency is not considered, the mining effects of the three are basically the same. However, the algorithm proposed in this paper has the highest precision, indicating that it can accurately mine potential correct rules without generating too many useless rules. Second, the algorithm proposed in this paper has the highest average F1 value, indicating that its performance in mining users' temporal association control habits is superior to the classical and common Apriori algorithm, Eclat algorithm, and FP-Growth algorithm. Therefore, the experimental results can confirm that the proposed algorithm is effective and has better performance, and can more accurately mine users' potential temporal association control habits with time constraints.

---

6 Conclusion

To meet the needs of smart home systems to mine users' temporal association control habits with time constraints, this paper proposes an association analysis algorithm based on the FP-Growth algorithm and data structure knowledge that can effectively mine users' temporal association control habits from large volumes of user historical control data. To improve the efficiency of transaction set generation, a forest-based data structure is proposed to store user control records, and transaction sets are efficiently generated by traversing the forest. Second, to address the problem that traditional association analysis algorithms cannot guarantee temporal ordering and strong temporal correlation among rule items in the mined association rules, an effective time constraint rule is proposed, which improves the precision of user associative control analysis algorithms by calculating dynamic time constraint factors. Finally, three common association analysis algorithms were selected as baseline algorithms for multiple repeated experiments, and the experimental results verified that the proposed algorithm can greatly improve the efficiency of transaction set generation while demonstrating superior performance in mining users' temporal association control habits.

Second, users'control habits may be influenced by external environmental factors (such as weather), but the real user control records used in the verification experiments of this paper do not include features of external environmental factors, making it impossible to use external environmental factors to guide the mining of users' control habits. Therefore, how to collect more external environmental factors that may cause significant changes in users' control habits and integrate them into the features of user control records to make the mined user control habits more consistent with real user situations is the focus of future research.

References

[1] Tong Xiaoyu, Fang Bingyi, Zhang Yun. Analysis of the development of IoT smart home [J]. Mobile Communications, 2010, 34(9): 16-20.

[2] Wen Tianle. On the current situation and development tendency of smart home in China [C]// Proc of the 7th International Conference on Education, Sports, Arts and Management Engineering. Paris: Atlantis Press, 2017: 272.

[3] Zhu Minling, Li Ning. State of art and trend of smart home in China [J]. Video Engineering, 2015, 39(4): 82-85.

[4] Ji Enqing, Shi Haigang, Li Hongyi, et al. Research on new remote control platform for smart home system using mobile phones [C]// Proc of AMM. Switzerland: Trans Tech Publications, 2014: 267-274.

[5] Xiao Biyin. Design and implementation of rule based uncertainty reasoning for smart house [D]. Chengdu: University of Electronic and Technology of China, 2016.

[6] Lee H S, Jung H W, Jung J Y, et al. A case study on the elderly people' s behavior for developing smart home service-focus on analyzing behaviors filling up by oneself for 24hours- [J]. Journal of Materials Science Materials in Medicine, 2012, 23(2): 307-14.

[7] Noury N, Hervé T, Rialle V, et al. Monitoring behavior in home using a smart fall sensor and position sensors [C]// Proc of the 1st Annual International Conference on Microtechnologies in Medicine and Biology. Piscataway, NJ: IEEE Press, 2000: 607-610.

[8] Lyu Peizhuo, Dai Hongtai. Research on the method of predicting the behavior of smart home users [J]. China New Technology and New Products, 2016(3): 19-20.

[9] Kong Yinghui, Liu Jing. Research on data mining of user behavior and control strategy in smart home [J]. Video Engineering, 2013, 37(24): 39-42.

[10] Jin K K, Kim K B, Jo S. A service scenario generation scheme based on association rule mining for elderly surveillance system in a smart home environment [J]. Engineering Applications of Artificial Intelligence, 2012, 25(7): 1355-1364.

[11] Heierman E O, Cook D J. Improving home automation by discovering regularly occurring device usage patterns [C]// Proc of the 3rd IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2003: 537.

[12] Niu Shaofeng. Research on the method of smart home pattern recognition based on data warehouse [D]. Beijing: Beijing University of Posts and Telecommunications, 2014.

[13] Hu Shengrong. Data structure tutorial and solution [M]. Beijing: Tsinghua University Press, 2011: 16-30.

[14] Yan Weimin, Wu Weimin. Data structure [M]. Beijing: Tsinghua University Press, 2009: 135-152.

[15] Zhou Zhihua. Machine learning [M]. Beijing: Tsinghua University Press, 2016: 197-224.

[16] Li Haoyuan, Wang Yi, Zhang Dong, et al. Pfp: parallel FP-Growth for query recommendation [C]// Proc of ACM RECSYS. NewYork: ACM Press, 2008: 107-114.

[17] Harrington P. Machine Learning in Action [M]. Beijing: Post and Telecom Press, 2012: 200-243.

[18] Hu Wenjiang, Hu Dawei, Gao Yongbing, et al. Friend recommendation algorithm based on association rules and tags [J]. Computer Engineering and Science, 2013, 35(2): 109-113.

Note: Figure translations are in progress. See original paper for figures.