Convenience-Based Periodic Composition of IoT Services

Bing Huang(✉), Athman Bouguettaya, and Azadeh Ghari Neiat

School of Information Technologies, The University of Sydney, Sydney, Australia
{bing.huang,athman.bouguettaya,azadeh.gharineiat}@sydney.edu.au

Abstract. We propose a novel service mining framework to personalize services in an IoT based smart home. We describe a new technique based on the concept of *convenience* to discover *periodic* composite IoT services to suit the smart home occupant's convenience needs. The key features of convenience is the ability to model the *spatio-temporal* aspects as occupants move in time and space within the smart home. We propose a novel framework for the *transient* composition of *spatio-temporal* IoT service. We design two strategies to prune non-promising compositions. Specifically, a *significance* model is proposed to prune insignificant composite IoT services. We describe a spatio-temporal *proximity* technique to prune loosely correlated composite IoT services. A periodic composite IoT service model is proposed to model the *regularity* of composite IoT services occurring at a certain location in a given time interval. The experimental results on real datasets show the efficiency and effectiveness of our proposed approach.

Keywords: IoT service · Periodic composite IoT services
Convenience

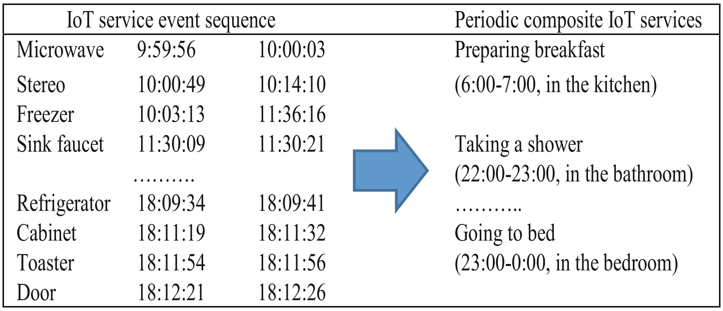
1 Introduction

The Internet is evolving from interconnecting computers to interconnecting things [1]. The Internet of Things (IoT) paradigm enables physical devices to connect and exchange information. IoT devices allow objects to be sensed or controlled remotely through the Internet [1]. The key challenge is that IoT devices are highly heterogeneous in terms of supporting infrastructure ranging from networking to programming abstraction [7]. Service-oriented Computing (SOC) is a promising solution for abstracting things on the Internet as services by hiding the complex and diverse supporting infrastructure [8]. This abstraction can shift the focus from dealing with technical details to how services are to be used [6]. We refer services for Internet of Things as IoT services. Daily life things such as a light is connected to the Internet and is represented as a light service.

An application domain for IoT is the smart home. A smart home can be considered as any regular home which has been augmented with various types of IoT services [15]. The purpose of a smart home is to make residents' life more

C. Pahl et al. (Eds.): ICSOC 2018, LNCS 11236, pp. 660–678, 2018.
https://doi.org/10.1007/978-3-030-03596-9_48

convenient and efficient [17]. Current research mainly focus on basic capabilities such as communication, computing, sensing and so on, which are indeed fundamental research topics [3]. However, these basic capabilities are not enough for IoT services. They should have more advanced intelligence i.e., the capability of understanding the physical world. To empower such high-level intelligence in smart homes, a key task is to discover *periodic composition of IoT services* which can represent periodic human activities. Periodic composite IoT services can be loosely defined as the composite IoT services' repeating occurrence at certain locations with regular time intervals. For example, a resident may have the habit of taking shower around 10 pm. It is of paramount importance to discover periodic composite IoT services. Periodic composition of IoT services can provide an insightful and concise explanation of IoT service usage patterns. These patterns can be used to design intelligent control of IoT services in smart homes to reduce residents' interactions with IoT services. Reducing those interactions provides more *convenience* for residents. Such periodic composite IoT services are also useful for human activity prediction. If an IoT service usage fails to follow its regular periodic composition, it could be a signal of abnormality.



IoT service event sequence			Periodic composite IoT services
Microwave	9:59:56	10:00:03	Preparing breakfast
Stereo	10:00:49	10:14:10	(6:00-7:00, in the kitchen)
Freezer	10:03:13	11:36:16	
Sink faucet	11:30:09	11:30:21	Taking a shower
	……….		(22:00-23:00, in the bathroom)
Refrigerator	18:09:34	18:09:41	………..
Cabinet	18:11:19	18:11:32	Going to bed
Toaster	18:11:54	18:11:56	(23:00-0:00, in the bedroom)
Door	18:12:21	18:12:26	

Fig. 1. Examples of periodic composite IoT services

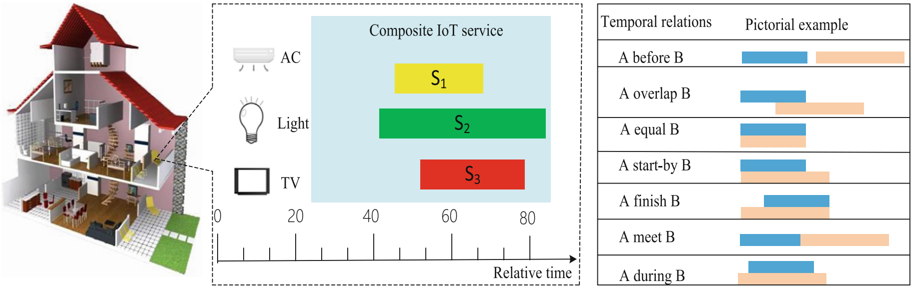
It is challenging to provide convenience by discovering periodic composite IoT services from IoT service usage history (i.e., IoT service event sequence). For example, Fig. 1 shows the IoT service usage history (on the left). We can see that it is difficult to extract periodic composite IoT services (on the right). We identify three key challenges.

- The set of IoT services are not known, which may be used collectively to fulfill a daily task. These IoT services are spatio-temporally correlated. We refer such set of IoT services as *composite IoT services.*
- There are many opportunities of establishing spatio-temporal relationships among IoT services, leading to an explosive number of possible composite IoT services. Many of these composite IoT services may be insignificant and loosely correlated. As a result, there is a need to prune insignificant and loosely correlated composite IoT services.

- The associated time interval and location for the periodic composite IoT services are not known. The composite IoT service may not occur exactly at the same time in a particular location. Therefore, there is a need to estimate the associated time interval and location.

In this paper, we focus on providing convenience by discovering periodic composite IoT services. At the first stage, we focus on discovering composite IoT services. Then, we employ *significance* and *proximity* strategies to prune insignificant and loosely correlated composite IoT services. At the third stage, we estimate associated time interval and location for the candidates generated in the second stage. Lastly, we measure how much convenience can be obtained by applying discovered periodic composite IoT services. The key contributions are as follows: (1) A new *IoT service model* and a *composite IoT service* model are proposed based on spatio-temporal features. (2) A *significance* model is proposed to prune insignificant composite IoT services. We also propose a *proximity* model in terms of *spatial-proximity* and *temporal-proximity* to filter out loosely correlated composite IoT services, (3) A *periodic composite IoT service* model is proposed to represent the regularity of composite IoT services occurring at a certain location in a time interval, (4) A *convenience* model is proposed to measure the benefits of applying periodic composite IoT services, and (5) A novel algorithm PCMiner (i.e., **P**eriodic **C**omposite IoT service **Miner**) is designed to discover periodic composite IoT services from event sequences.

The rest of the paper is organized as follows. Section 2 formally defines key concepts. Section 3 details the proposed algorithm PCMiner. Section 4 shows the experimental results. Section 5 surveys the related work. Section 6 conclude the paper and highlights some future work.



(a)

(b)

Fig. 2. (a) An example of a composite IoT service; (b) Time interval relations

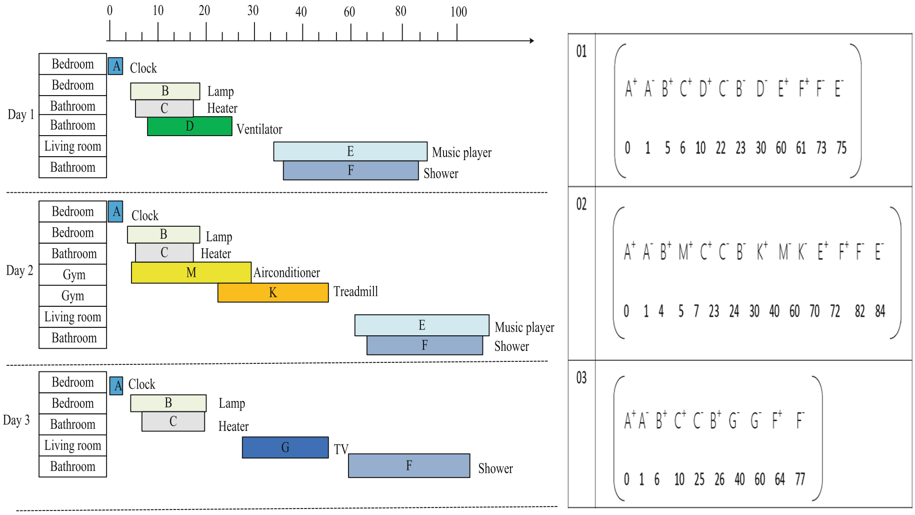
Motivating Scenario

We use the smart home as our motivating scenario. Sarah lives alone in a smart home. We assume everything such as lights, TV, oven, window, and floors are connected to the Internet and represented as IoT services. This smart home aims to improve Sarah's life convenience. Intuitively, *convenience* can be interpreted as a smart home system which is aware of a resident's potential needs by understanding physical environment or situations and respond proactively at the right time and in the right place. This reduces a resident's interactions with IoT services. Let us imagine an interesting convenient life scenario in Sarah's home. In a weekday morning, the clock wakes Sarah up at 8 am. Then the lamp is turned on automatically. Sarah gets up and prepares stuff for taking a shower. Meanwhile, the heater in the bathroom starts to heat. When Sarah steps into the bathroom, it is already warm. While Sarah is taking shower, the music player is playing her favorite music. At the same time, the kettle in the kitchen starts to work and the coffee maker starts to make a cup of Mocha coffee. After Sarah finishes showering and grooming, she goes to the kitchen to prepare for breakfast. When she is enjoying the breakfast and coffee, the TV is turned on automatically and displays her favorite sport news. After finishing the breakfast, Sarah goes to work. The TV and all the lights are turned off automatically.

A fundamental task to provide convenience is to augment IoT services with capabilities of understanding the periodic usage of composite IoT services. Let us consider Sarah performs daily activities by interacting with IoT services. These interactions are recorded as IoT service event sequences shown in Fig. 3. For example, $<E^{+}E^{-}, (60, 75)>$ denotes the music player is playing music from time 60 to time 75 and E^{+} (resp.E^{-}) denotes a turn on (resp. turn off) the music player event. There exist spatio-temporal relationships among IoT services. For example, in Day 1, the relationships between the music player service (i.e., E) and the shower service (i.e., F) shows that Sarah listens to music while taking a shower. In this regard, a collection of spatio-temporally correlated IoT services may represent an activity. We refer to such set of IoT services as *composite IoT services*. An example of a composite IoT service is shown in Fig. 2(a).

There are many ways of establishing spatio-temporal relationships among IoT services. According to Allen's temporal logic in Fig. 2(b), there are 57 ways of generating composite IoT services in Day 1 by a brute force approach (i.e., $C_6^2 + C_6^3 + C_6^4 + C_6^5 + C_6^6 = 57$). Many of them may be insignificant and loosely correlated. Therefore, we design the *significance* and *proximity* strategies to filter out these insignificant and loosely correlated composite IoT services.

The proximate and significant composite IoT services can represent the resident's daily activities. The residents usually performs his/her daily activities periodically in terms of time and location. For example, the resident usually goes to bed during 11pm to 12pm and wakes up during 8am to 9am. We refer such repeating composite IoT services at certain location with regular time intervals as *periodic composite IoT services*. The periodic composite IoT services can be serve as a knowledge basis for providing convenience. A *convenience* model is introduced to quantify how much benefits of applying periodic composite IoT services. In this paper, we focus on providing convenience by discovering periodic composite IoT services.



(a) (b)

Fig. 3. An example of service event sequences

2 System Model

We first introduce the notion of IoT services and composite IoT services based on spatio-temporal features [25]. Then, a significance model and a proximity model are proposed to prune non-promising composite IoT services. We introduce the notion of periodic composite IoT service to model the occurrence regularity of composite IoT services. Lastly, we provide a convenience model to quantify how much benefits of applying periodic composite IoT services.

2.1 IoT Service Model

Definition 1: IoT Service. An IoT service S_i is a tuple $S_i = <s_i, F_i, IS, FS>$, where:

- s_i is a unique service identifier.
- F_i is a set of functions that are offered by S_i.
- IS (Initial State) is a tuple $<s_i^+, st_i, sl_i>$, where
  - s_i^+ is a symbol of IS.
  - st_i is a start-time of S_i.
  - $sl_i = <x_s, y_s>$ is a start location of S_i, where $<x_s, y_s>$ is a GPS point.
- FS (Final State) is a tuple $<s_i^-, et_i, el_i>$, where
  - s_i^- is a symbol of FS.
  - et_i is an end-time of S_i.
  - $el_i = <x_e, y_e>$ is an end location of S_i, where $<x_e, y_e>$ is a GPS point.

We focus on the spatio-temporal features in the remainder of this paper. Thus, the representation of an IoT service S_i is simplified as $<(s_i^+, st_i, sl_i), (s_i^-, et_i, el_i)>$. For example, a light service is represented as $<(light^+, 7pm, (1,2)), (light^-, 9pm, (1,2))>$ which is described as lighting from 7pm to 9pm in the bedroom where $(1,2)$ is the GPS point in the bedroom and 7pm (resp. 9pm) is the start time (resp. end time).

2.2 Composite IoT Service Model

One IoT service may not accomplish a daily activity. Multiple IoT services may be composed to fulfill an activity [13]. These IoT services may be used collectively based on time and location correlations to accomplish a certain daily activity. We refer such spatio-temporally correlated IoT services as composite IoT services.

Definition 2: Composite IoT Service. A composite IoT service CS is a collection of IoT services that occur frequently in a particular spatio-temporal relationships. A composite IoT service is denoted by a tuple $CS = <S, sup(S)>$ where

- $S = \{<(s_1^+, st_1, sl_1), (s_1^-, et_1, el_1)>, ..., <(s_n^+, st_n, sl_n), (s_n^-, et_n, el_n)>\}$ represents n component IoT services where $<(s_i^+, st_i, sl_i), (s_i^-, et_i, el_i)>$ is a component IoT service S_i as defined in Definition 1 and $st_i \leq st_{i+1}$ and $st_i \leq et_i$. An example of a composite IoT service is shown in Fig. 2(a). By ordering all elements s_i^* ($*$ can be $+$ or $-$) in S in a non-decreasing order based on its associated time information st_i(or et_i), we can transform S into the following representation $S = <Seq, T, L> = \left\{\begin{matrix} \alpha_1 & ... & \alpha_i & ... & \alpha_{2n} \\ t_1 & ... & t_i & ... & t_{2n} \\ l_1 & ... & l_i & ... & l_{2n} \end{matrix}\right\}$, where $Seq = \{\alpha_1 ... \alpha_i ... \alpha_{2n}\}$ is a symbol sequence and $\alpha_i = s_j^*$ ($*$ can be $+$ or $-$), $T = \{t_1 ... t_i ... t_{2n}\}$ is the time information and $t_i \leq t_{i+1}$, and $L = \{l_1 ... l_i ... l_{2n}\}$ is the location information. For example, the composite IoT service in Fig. 2(a) can be represented as $\left\{\begin{matrix} s_2^+ & s_1^+ & s_3^+ & s_1^- & s_3^- & s_2^- \\ 48 & 50 & 58 & 65 & 70 & 75 \\ l_2 & l_1 & l_3 & l_1 & l_3 & l_2 \end{matrix}\right\}$ (i.e., $l_1 = (1,2)$, $l_2 = (2,4)$, $l_3 = (3,5)$)
- $sup(S)$ is the support for S. The support sup is the total number of occurrence in a database $S = <Seq, T, L>$ and $S' = <Seq', T', L'>$ be two composite IoT services. S is referred to as a sub-composite IoT service of S' , denoted as $S \sqsubseteq S'$, if $Seq = \{\alpha_1 ... \alpha_i ... \alpha_{2n}\}$ is a subsequence of $Seq' = \{\alpha'_1 ... \alpha'_i ... \alpha'_{2m}\}$, denoted as $Seq \sqsubseteq Seq'$ with $n \leq m$. $Seq \sqsubseteq Seq'$ is satisfied if there exist integers $1 \leq k_1 \leq k_2 ... k_n \leq k_{2m}$ such that $\alpha_1 \subseteq \alpha'_{k_1}, \alpha_2 \subseteq \alpha'_{k_2}, \ldots, \alpha_n \subseteq \alpha'_{k_n}$. Given an IoT service event sequence DB, the tuple (sid, S') (i.e., sid is a sequence ID and S' is the composite IoT service) is said to contain a sub-composite IoT service S if $S \sqsubseteq S'$. The support sup of S in DB, denoted

as $sup(S)$ is the number of tuples containing S. $sup(S)$ can be formalized as follows.

$$sup(S) = |\{(sid, S) \in DB | S \sqsubseteq S'\}| \tag{1}$$

2.3 Significance and Proximity Model for Composite IoT Services

There are many possibilities of establishing spatio-temporal relationships among IoT services, leading to an explosive generation of composite IoT services. Many of the composite IoT services are insignificant and loosely correlated. Thus, there is a need to filter out these non-promising composite IoT services. We explore this problem from two aspects. On the one hand, those composite IoT services that occur frequently are more likely to be significant that those occur less frequently. Thus, we propose a $significance$ model to quantify how much significance these composite IoT services are from the statistic aspect. On the other hand, the IoT services that occur proximately in terms of time and location are more likely to be correlated. For example, from the spatial perspective, the relationship between the TV and the light in the same dining room may reveal a high correlation between these two IoT services. However, the co-occurrence of using TV in the dining room and using the light in the bedroom may merely be a coincidence. From the temporal perspective, using the TV and the light in the evening reveal that there may exist a high correlation between the two IoT services during that time. However, using the TV in the evening and using the light in the morning may not have any correlation. In this regard, we use *proximity* to characterizes correlation strength among component IoT services in terms of *spatial-proximity* and *temporal-proximity*. By *spatial-proximity*, it characterizes the location correlation strength among component IoT services. By *temporal-proximity*, it characterizes the temporal correlation strength among component IoT services. The *proximity* model is adapted from the approach for measuring spatio-temporal interval data distance [11].

Definition 3: Significance. *Significance* is used for evaluating statistic importance of CS. Given a composite IoT service $CS = <S, sup>$, its *significance* is formalized as:

$$significance(S) = \frac{\sqrt{expect(S)}}{sup(S) - expect(S)} \tag{2}$$

where $expect(S)$ is the expected number of occurrence in a DB. To estimate $expect(S)$, we adapt the statics model proposed in [14] by considering IoT services' various usage frequency across different regions in smart homes [16]. In practice, IoT services' usage frequency may vary across different regions in smart homes. For example, if a resident spends most of his/her time in their living room during the day and only goes to their bedroom for sleeping, then IoT services in the living room will be used more frequently than those in the bedroom. Thus, composite IoT services for sleeping may be ignored when searching for frequent composite IoT services.

Given a composite IoT service CS whose symbol sequence is $Seq = \{s_1^*...s_i^*...s_m^*\}$, suppose that Seq is a possible outcome drawn from the symbol set $A = \{a_1...a_i...a_n\}$ with $P(a_i)$ following Bernoulli distribution such that $\sum_{i=1}^{n} P(a_i) = 1$. Given a DB and a region set $R = \{r_1, ..., r_k\}$, DB_{r_i} records IoT service event sequences occurring at region r_i. $Num(a_i)_{DB_{r_i}}$ is the number of the event a_i occurrence in database DB_{r_i}. $P(a_i)$ can be estimated by $\frac{Num(a_i)_{DB_{r_i}}}{\sum_{a_j \in A} Num(a_j)_{DB_{r_i}}}$. Therefore, the occurrence probability $P(S)$ and $expect(S)$ can be formalized as follows.

$$P(S) = \prod_{\forall s_i^+ \in Seq} P(s_i^*) \tag{3}$$

$$expect(S) = P(S) \cdot |DB_{r_i}| \tag{4}$$

where $|DB_{r_i}|$ is number of IoT service events in region r_i. Note that we only count $P(s_i^*)$ once because each s_i^* has two points (i.e., s_i^+ and s_i^-).

Definition 4: Proximity. Given a composite IoT service $CS = <S, sup>$ where S is in the form $S = <Seq, T, L> = \left\{ \begin{matrix} \alpha_1 & ... & \alpha_i & ... & \alpha_{2n} \\ t_1 & ... & t_i & ... & t_{2n} \\ l_1 & ... & l_i & ... & l_{2n} \end{matrix} \right\}$, its proximity function is defined as follows.

$$U = w_1 \cdot spatial\_proximity + w_2 \cdot temporal\_proximity \tag{5}$$

where $w_i (i = 1, 2)$ is a weight such that $w_i \in [0, 1]$ and $w_1 + w_2 = 1$. The *spatial_proximity* and *temporal_proximity* are formalized as follows.

- *Spatial_proximity*: The *spatial_proximity* measures the average location proximity of all composite IoT service instances. The *spatial_proximity* for a composite IoT service instance is first formalized in Eq. (6). Then the average *spatial_proximity* for the composite IoT service is formalized in Eq. (7).

$$Spa = \sum_{i=1}^{n} \frac{1}{|x_i - x_{i+1}| + |y_i - y_{i+1}|} \tag{6}$$

$$spatial\_proximity = \frac{\sum_{j=1}^{sup} Spa_j}{sup} \tag{7}$$

where n is the total number of component IoT services, $l_i = <x_i, y_i>$ and $l_{i+1} = <x_{i+1}, y_{i+1}>$ are two locations for two consecutive component IoT services, and sup is the support for the composite IoT service. For example, for the composite IoT service in Fig. 2(a), its *spatial_proximity* score is $Spa = \frac{1}{|l_2 - l_1|} + \frac{1}{|l_1 - l_3|} = \frac{1}{|2-1|+|4-2|} + \frac{1}{|1-3|+|2-5|} = 0.53$. In this paper, we use Manhattan distance proposed in [12] to measure the proximity because it is computing efficient.

- *Temporal_proximity*: The *temporal_proximity* measures the average temporal proximity of all composite IoT service instances. We adapt the technique of evaluating the distance between time-interval based data in [11]. For each component IoT service instance $S_i = \{(s_i^+, st_i, sl_i), (s_i^-, et_i, el_i)\}$, we utilize a function f_i with respect to t to map the temporal aspect of S_i. f_i is formalized as follows.

$$f_i(t) = \begin{cases} 1, & t \in [st_i, et_i] \\ 0, & otherwise \end{cases} \tag{8}$$

Then we have a set of functions $\{f_1, f_2, ...f_n\}$ corresponding to the composite IoT service instance. The *temporal_proximity* for the composite IoT service instance is calculated by Eq. (9). The average *temporal_proximity* is calculated by Eq. (10).

$$Temp = \frac{\int_{t_1}^{t_{2n}} \sum_{i=1}^{n} f_i(t)\, dt}{(t_{2n} - t_1) \cdot n} \tag{9}$$

$$temporal\_proximity = \frac{\sum_{j=1}^{sup} Temp_j}{sup} \tag{10}$$

where t_1 and t_{2n} are the first and the last time information of CS, respectively, and n is the number of component IoT services. For example, the *temporal_proximity* score of the composite IoT service {<stove, [18:00, 19:00]>, <washing machine, [18:40, 19:20]>} can be calculated as $\frac{(18{:}40-18{:}00)+(19{:}00-18{:}40)\cdot 2+(19{:}20-19{:}00)}{(19{:}20-18{:}00)\cdot 2} = 0.625$. This composite IoT service can be interpreted as *when the resident is cooking, he/she is also doing laundry.* Another composite IoT service is {<stove, [18:00, 19:00]>, <fan, [18:00, 19:00]>} and its *temporal_proximity* score is 1. Thus the latter composite IoT service is considered to be more temporally proximate than the former.

2.4 Periodic Composite IoT Service Model

In this section, we introduce the novel notion of periodic composite IoT service to model the regularity of repeating composite IoT services.

Definition 5: Periodic composite IoT service. A periodic composite IoT service PC is defined as the repeating composite IoT services at certain locations with regular time intervals. It is denoted by a tuple $PC = <CS, T, L, P>$ where

- CS is a composite IoT service.
- $T = <T_s, T_e>$ is a representative time interval associated with CS, where T_s and T_e are the start time and end time of CS, respectively. Suppose all start time and end time of CS in DB constitutes the set $\tau = \{<st_1, et_1>, <st_2, et_2>...<st_m, et_m>\}$. We need to find the representative time interval $<T_s, T_e>$ which minimizes the dissimilarity between the instance $<st_i, et_i>$. We define the dissimilarity dis between two time intervals.

$$dis = |T_s - st_i| + |T_e - et_i| \tag{11}$$

Thus, the total dissimilarity between $<T_s, T_e>$ and τ can be defined by Eq. (12).

$$Dis(T, \tau) = \sum_{i=1}^{m} |T_s - st_i| + |T_e - et_i| \tag{12}$$

To minimize $Dis(T, \tau)$, Eq. (12) can be transformed into two known minimization problems, that is, find T_s and T_e to minimize $\sum_{i=1}^{m} |T_s - st_i|$ and $\sum_{i=1}^{m} |T_e - et_i|$, respectively. T_s is the median of the start time set $\{st_1, st_2 ... st_m\}$ and T_e is the median of the end time set $\{et_1, et_2 ... et_m\}$. The proof can be found in [19].

- L is the region location of CS such as the bedroom and the bathroom.
- P is the probability of CS occurring around time interval T at location L. Suppose the time information of a CS instance is $<st_j, et_j>$. The CS is said to occur around time interval T if their dissimilarity dis is no more than a tolerance threshold ζ, that is, $|T_s - st_j| + |T_e - et_j| \le \zeta$. P can be formalized as follows.

$$P = \frac{Num}{TNum} \tag{13}$$

where Num is the number of CS occurrence around time interval T at location L. $TNum$ is the total number of CS occurrence in the database.

2.5 Convenience Model

The discovered periodic composite IoT services can be served as knowledge basis for building an intelligent system to provide *convenience* for the residents. By convenience, it is interpreted as the benefits of applying periodic composite IoT services via reducing residents' interactions with IoT services. The convenience can be quantified as follows.

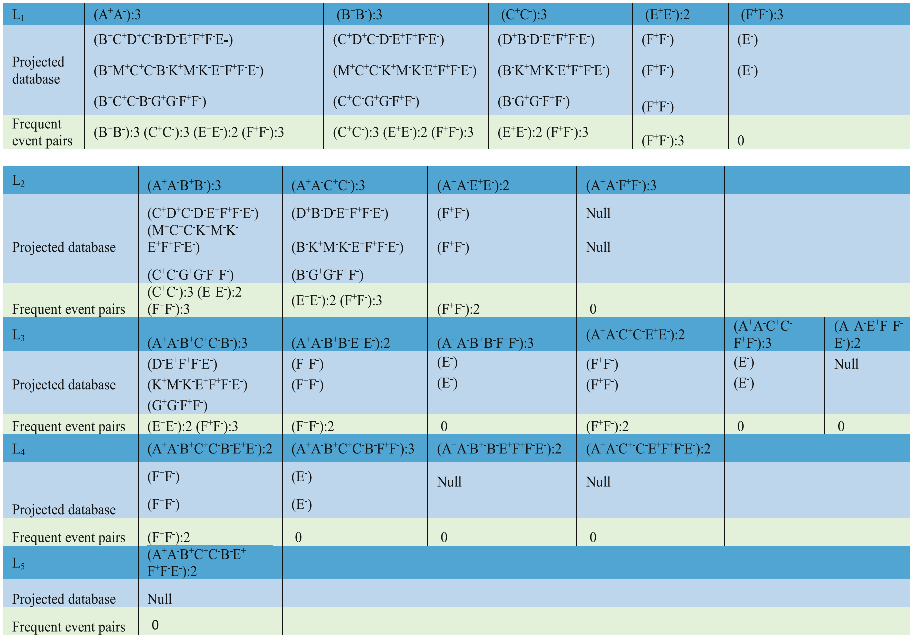
Definition 6: Convenience. Given an IoT service event sequence $\{<(a_1^+, st_1, sl_1), (a_1^-, et_1, el_1)> \ldots <(a_n^+, st_n, sl_n), (a_n^-, st_n, sl_n)>\}$ during the time period $[st_1, et_n]$, suppose this sequence is initialized from a set of periodic composite services $\{\ PC_1, PC_2 \ldots PC_m\}$. According to the representative time information of the PC_m, we can roughly estimate the next PC_{m+1} occurrence. The IoT service events involved in PC_{m+1} is $\{\ b_1,\ b_2 ... b_m\ \}$. Suppose the actual event set occurs next is $\{\ c_1, c_2 \ldots c_k\ \}$. Therefore, the amount of convenience can be quantified by Eq. (14).

$$convenience = \frac{|\{b_1, b_2 ... b_m\} \cap \{c_1, c_2 ... c_k\}|}{|\{c_1, c_2 ... c_k\}|} \tag{14}$$

where $|\{c_1, c_2 ... c_k\}|$ is the number of events and $|\{b_1, b_2 ... b_m\} \cap \{c_1, c_2 ... c_k\}|$ is the number of correctly estimated events.

3 Discovering Periodic Composite IoT Service Approach

We develop the algorithm PCMiner to efficiently discover periodic composite IoT services from IoT service event sequences. Algorithm 1 shows the details of PCMiner. The algorithm consists of four phases. The mining process starts with *dividing the search space*. Then PCMiner *searches all composite IoT services* in a determined space. Third, PCMiner applies the *significance* and *proximity* strategies to remove non-promising composite IoT services. Finally, PCMiner collects time information and location information for candidates generated in the third phase. Based on these information, time period and location corresponding to the candidates are estimated which leads to generating a set of *periodic composite IoT services*. For the sake of consistency with the terms from data analysis techniques, we use *event patterns* to refer to composite IoT services occurring in service event sequences. We use the running example in Fig. 3 to illustrate the process of PCMiner shown in Fig. 4.



L_1	$(A^{+}A^{-})$:3	$(B^{+}B^{-})$:3	$(C^{+}C^{-})$:3	$(E^{+}E^{-})$:2	$(F^{+}F^{-})$:3
Projected database	$(B^{+}C^{+}D^{+}C^{-}B^{-}D^{-}E^{+}F^{+}F^{-}E-)$ $(B^{+}M^{+}C^{+}C^{-}B^{-}K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(B^{+}C^{+}C^{-}B^{-}G^{+}G^{-}F^{+}F^{-})$	$(C^{+}D^{+}C^{-}D^{-}E^{+}F^{+}F^{-}E^{-})$ $(M^{+}C^{+}C^{-}K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(C^{+}C^{-}G^{+}G^{-}F^{+}F^{-})$	$(D^{+}B^{-}D^{-}E^{+}F^{+}F^{-}E^{-})$ $(B^{-}K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(B^{-}G^{+}G^{-}F^{+}F^{-})$	$(F^{+}F^{-})$ $(F^{+}F^{-})$ $(F^{+}F^{-})$	(E^{-}) (E^{-})
Frequent event pairs	$(B^{+}B^{-})$:3 $(C^{+}C^{-})$:3 $(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(C^{+}C^{-})$:3 $(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(F^{+}F^{-})$:3	0

L_2	$(A^{+}A^{-}B^{+}B^{-})$:3	$(A^{+}A^{-}C^{+}C^{-})$:3	$(A^{+}A^{-}E^{+}E^{-})$:2	$(A^{+}A^{-}F^{+}F^{-})$:3		
Projected database	$(C^{+}D^{+}C^{-}D^{-}E^{+}F^{+}F^{-}E^{-})$ $(M^{+}C^{+}C^{-}K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(C^{+}C^{-}G^{+}G^{-}F^{+}F^{-})$	$(D^{+}B^{-}D^{-}E^{+}F^{+}F^{-}E^{-})$ $(B^{-}K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(B^{-}G^{+}G^{-}F^{+}F^{-})$	$(F^{+}F^{-})$ $(F^{+}F^{-})$	Null Null		
Frequent event pairs	$(C^{+}C^{-})$:3 $(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(F^{+}F^{-})$:2	0		
L_3	$(A^{+}A^{-}B^{+}C^{+}C^{-}B^{-})$:3	$(A^{+}A^{-}B^{+}B^{-}E^{+}E^{-})$:2	$(A^{+}A^{-}B^{+}B^{-}F^{+}F^{-})$:3	$(A^{+}A^{-}C^{+}C^{-}E^{+}E^{-})$:2	$(A^{+}A^{-}C^{+}C^{-}F^{+}F^{-})$:3	$(A^{+}A^{-}E^{+}F^{+}F^{-}E^{-})$:2
Projected database	$(D^{-}E^{+}F^{+}F^{-}E^{-})$ $(K^{+}M^{-}K^{-}E^{+}F^{+}F^{-}E^{-})$ $(G^{+}G^{-}F^{+}F^{-})$	$(F^{+}F^{-})$ $(F^{+}F^{-})$	(E^{-}) (E^{-})	$(F^{+}F^{-})$ $(F^{+}F^{-})$	(E^{-}) (E^{-})	Null
Frequent event pairs	$(E^{+}E^{-})$:2 $(F^{+}F^{-})$:3	$(F^{+}F^{-})$:2	0	$(F^{+}F^{-})$:2	0	0
L_4	$(A^{+}A^{-}B^{+}C^{+}C^{-}B^{-}E^{+}E^{-})$:2	$(A^{+}A^{-}B^{+}C^{+}C^{-}B^{-}F^{+}F^{-})$:3	$(A^{+}A^{-}B^{+-}B^{-}E^{+}F^{+}F^{-}E^{-})$:2	$(A^{+}A^{-}C^{+-}C^{-}E^{+}F^{+}F^{-}E^{-})$:2		
Projected database	$(F^{+}F^{-})$ $(F^{+}F^{-})$	(E^{-}) (E^{-})	Null	Null		
Frequent event pairs	$(F^{+}F^{-})$:2	0	0	0		
L_5	$(A^{+}A^{-}B^{+}C^{+}C^{-}B^{-}E^{+}F^{+}F^{-}E^{-})$:2					
Projected database	Null					
Frequent event pairs	0					

Fig. 4. The Process of PCMiner

Phase I: Dividing Search Space. The layout of a smart home consists of multiple regions such as a bedroom and a kitchen. Each IoT service event is associated with a region. For example, turn on the lamp event occurs in the bedroom. Given a region set $r = \{r_1, r_2 ... r_n\}$, we divide the database DB into multiple smaller databases DB_{r_i}. Each DB_{r_i} records IoT service event sequences occurring in the region r_i. In later phases, the discovering process performs on each sub-databases. For the purpose of illustrating PCMiner, we assume all IoT service event sequences in the running example shown in Fig. 3 are from the same region and constitute a sub-database.

Phase II: Searching Event Patterns. PCMiner employs a divide-and-conquer, pattern-growth principle from Prefixspan [23] as follows: event sequence databases are recursively projected into a set of smaller *projected databases* based on the current event patterns. Event patterns are then grown by searching the smaller projected databases.

Definition 7: Projected database. Let p be an event pattern in a database DB. The p-projected database, denoted as $DB|_p$, is the collection of suffixes of event sequences in DB with regard to the prefix p.

The searching process consists of three sub-phases. PCMiner first finds the set of 1-length event patterns. Then, PCMiner constructs projected databases for each 1-length event pattern generated in the phase one. Third, the event patterns are grown by searching their corresponding projected databases. Each of these sub-phases is detailed as follows.

1. Find the set of 1-length event patterns L_1. Given a database as shown in Fig. 3, PCMiner first scans the database to count the number of each event pairs and discards those events whose support is less than the minimum support threshold. If the *minsup* threshold is 2, all discovered 1-length event patterns whose support is not less than 2 constitute the 1-length event pattern set L_1. For example, in Fig. 4, (A^+A^-): 3 denotes the event pattern and its associated support count.

2. Construct projected databases for each 1-length event pattern. Let $L_1 = \{\alpha_1^1, \alpha_2^1 ... \alpha_n^1\}$ be the set of 1-length event patterns. For each α_i^1, a corresponding projected database $DB|_{\alpha_i^1}$ is created. $DB|_{\alpha_i^1}$ is a collection of suffix event sequences with regard to the prefix α_i^1.

3. k-length event pattern α *is grown to the* $(k+1)$*-length event pattern* α' *through searching the projected database* $DB|_\alpha$ *corresponding to* α $(k \geq 1)$. For a prefix α, PCMiner scans its projected database $DB|_\alpha$ once to find the set of local frequent event pairs $\{e_1, e_2 \dots e_n\}$ and discards infrequent ones. Note that since event pairs are counted, these single events in the projected database will not be counted again. Frequent event pairs e_i are appended to the prefix α, generating the new frequent event pattern α' with the length increased by 1. Therefore, the set of $(\mathrm{k}+1)$-length event patterns prefixed with α are generated.

We illustrate the process of finding event patterns prefixed with (A^+A^-). By scanning the (A^+A^-) projected database $DB|_{(A^+A^-)}$, its local frequent event pairs are $(B^+B^-$: 3), $(C^+C^-$: 3), $(E^+E^-$: 2), and $(F^+F^-$: 3). Thus, the set of all 2-length event patterns L_2 prefixed with (A^+A^-) are found, and they are: $(A^+A^-B^+B^-$: 3), $(A^+A^-C^+C^-$: 3), $(A^+A^-E^+E^-$: 2), and $(A^+A^-F^+F^-$: 3). Recursively, all 2-length event patterns are used to find 3-length event patterns by constructing and searching their projected databases. By projecting $(A^+A^-B^+B^-)$, we find frequent event pairs from its projected database which are (C^+C^-: 3), (E^+E^-: 2), and (F^+F^-: 3). By appending these frequent event pairs to the prefix ($A^+A^-B^+B^-$), we have 3-length event patterns ($A^+A^-B^+C^+C^-B^-$: 3), ($A^+A^-B^+B^-E^+E^-$: 2), and ($A^+A^-B^+B^-F^+F^-$: 3). Similarly, we find ($A^+A^-C^+C^-E^+E^-$: 2) and ($A^+A^-C^+C^-F^+F^-$: 3) prefixed with ($A^+A^-C^+C^-$), and ($A^+A^-E^+F^+F^-E^-$: 2) prefixed with ($A^+A^-E^+E^-$). We find L_4 and L_5 in the same approach.

Phase III: Calculate Significance and Proximity for Event Patterns. For each event pattern generated in phase II, we collect the time information and location information from the event sequences. Based on these information, we calculate the statistic significance for each event pattern by Definition 3. We also discard insignificant ones if its significance is less than the significance threshold *minsig*. Given proximity threshold *minpro*, we calculate average proximity for each event pattern by Definition 4 and filter out those patterns whose proximity are less than *minpro*. For the running example, if the weight for *spatial_proximity* and *temporal_proximity* is set to be 0 and 1, respectively, the proximity of all 2-length event patterns are $Prox(A^+A^-B^+B^-) = 0.418$, $Prox(A^+A^-C^+C^-) = 0.358$, $Prox(A^+A^-E^+E^-) = 0.098$, $Prox(A^+A^-F^+F^-) = 0.082$, $Prox(B^+C^+C^-B^-) = 0.906$, $Prox(B^+B^-E^+E^-) = 0.224$, $Prox(B^+B^-F^+F^-) = 0.215$, $Prox(C^+C^-E^+E^-) = 0.209$, $Prox(C^+C^-F^+F^-) = 0.199$, $Prox(E^+F^+F^-E^-) = 0.879$. If the *minpro* is set to be 0.3, event patterns whose proximity is less than 0.3 are filtered out. The ultimate outcomes of our example are $<E^+F^+F^-E^->$ and $<A^+A^-B^+C^+C^-B^->$ and their respective proximity are 0.879 and 0.494.

Phase IV: Generating Periodic Event Patterns. After performing phase III, we obtain significant and proximate event patterns. Based on the time and location information collected in phase III, the algorithm estimates the time period, location, and probability for each event pattern by Definition 5. The outcomes of this phase is a set of event patterns associated with time intervals, location, and a probability (i.e., periodic composite IoT services).

4 Experimental Results

We systematically evaluate the approach proposed in this paper. The language used is Java and the experiments are performed on a 1.6 GHz AMD processor and 2 GB RAM under Windows 7. We evaluate the proposed approach using three real datasets, namely, Data1, Data2, Data3. Specifically, Data1 and Data2 are from CASAS datasets, which are collected in smart home environment [18]. For location information, we refer to the layout of sensors attached on objects for grouping objects into corresponding locations. Data1 and Data2 are in the format of <date, time stamp, sensor ID, on/off> (e.g., <2008-02-27, 12:46:37, M13, OFF >). Data3 is collected from a single apartment for two weeks [10]. The Data3 are in the format of <id, start time, end time, location> (e.g., <light,

Algorithm 1. PCMiner algorithm

```
Input: A spatio-temporal database ($DB$), region set $r = \{r_1, r_2 ... r_n\}$, support threshold ($minsup$), significance threshold ($minsig$), proximity threshold ($minpro$), weight of the $spatial\_proximity$ ($w_1$), and weight of the $temporal\_proximity$ ($w_2$).
Output: A set of event patterns, a set of periodic event patterns.

Procedure Divide_Search_Space ($DB$, $r$)
    transform $DB$ into $|r|$ numbers of sub-databases based on $r$;
    return $DB_{r_i}$;

Procedure Search_Event_Pattern ($DB_{r_i}$, $minsup$)
    $L_1$ = find_1-length_pattern($DB_{r_i}$, $minsup$ );
    Prefixspan($\alpha$, $l$, $DB_{r_i}|_\alpha$)//$\alpha$ is an event pattern, $l$ is the length of $\alpha$ ($l \geq 1$), $DB_{r_i}|_\alpha$ is the $\alpha$-projected database.
    Scan $DB_{r_i}|_\alpha$ once, find the set of local frequent events $\{e_1, e_2 ... e_n\}$;
    for each frequent event $e_i$ do
        $\alpha' = \alpha + e_i$; //append $e_i$ to $\alpha$ to form the new event pattern $\alpha'$ .
    for each $\alpha'$ do
        add $\alpha'$ to the event pattern set $patternset$;
        construct $\alpha'$-projected database $DB_{r_i}|_{\alpha'}$, and call Prefixspan ($\alpha'$, $l+1$, $DB_{r_i}|_{\alpha'}$);
    return $patternset$;

Procedure Compute_Significance_Proximity ($patternset$, $minsig$, $minpro$, $w_1$, $w_2$);
    for each pattern $\alpha$ in $patternset$ do
        computeSignificance($\alpha$, $minsig$); // Calculate significance for each pattern by Definition 3 and discard those patterns whose significance are less than $minsig$.
        computeProximity ( $\alpha$, $w_1$, $w_2$, $minpro$);// Calculate proximity for each pattern by Definition 4 and discard those patterns whose proximity are less than $minpro$.
    return $patternset'$;

Procedure Generate_Periodic_Event_Patterns ($patternset'$, $DB_{r_i}$);
    for each pattern $\alpha$ in $patternset'$ do
        find associated time and location information from $DB_{r_i}$;
        estimate the time interval $T$, location $L$, and probability $P$ for $\alpha$ by Definition 5;
    return $<\alpha, T, L, P>$;
```

7:00, 8:00, bedroom>). In addition, all datasets are annotated with corresponding daily activities. There are 5 and 8 activities in Data1 and Data2, respectively. For Data3, 23 activities are recorded and annotated. We conduct four sets of experiments. The first set is to evaluate the performance and scalability of PCMiner. The second set is to evaluate the effectiveness of the pruning strategies (i.e., significance and proximity). The third set is to evaluate the applicability of the proposed approach by showing the discovered periodic composite IoT services from real datasets. The fourth set is to measure the convenience by applying the discovered periodic composite IoT services.

The first set of experiments is conducted on a dataset which is a combination of three datasets i.e., Data1, Data2, and Data3. We vary the support threshold *sup* from 4% to 10%. Figure 5(a) shows the execution time of PCMiner decreases by increasing the support threshold. Figure 5(b) illustrates that the number of discovered event patterns decreases by increasing the support.

In the second set of experiments, we assess the effectiveness of $significance$ and *proximity* in pruning non-promising event patterns. Similar to our previous experiment, we use the combined dataset. We set the $significance$ to be 0.01. We test the effectiveness of $significance$ in reducing insignificant IoT service event patterns while varying different support threshold. Figure 6(a) depicts

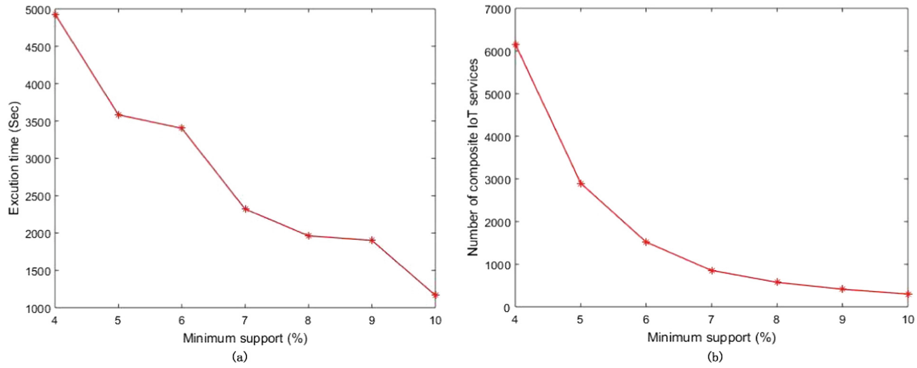


Fig. 5. Performance and scalability of PCMiner

the number of discovered patterns and significant patterns at different support threshold. The results show that the *significance* strategy performs effectively in pruning insignificant event patterns, which is an expected results. For example, the *significance* strategy can prune event patterns from 2954 to 1108 at the 5% support threshold. In addition, we test the effectiveness of *proximity* strategy in filtering out loosely correlated event patterns. Since the GPS point of each service in not available in the datasets, we set the weight of *spatial_proximity* to be 0 and the weight of *temporal_proximity* to be 1. We set the *proximity* to be 0.39. Figure 6(b) illustrates the number of discovered event patterns and proximate event patterns at different support threshold. The results show that the *proximity* strategy is effective. For example, the *proximity* strategy can prune loosely correlated event patterns from 2954 to 1705 at the 5% support threshold. These are expected results because the *significance* and *proximity* strategies enable PCMiner to filter out non-promising event patterns in each iteration and the search scope is shrunk for the next iteration.

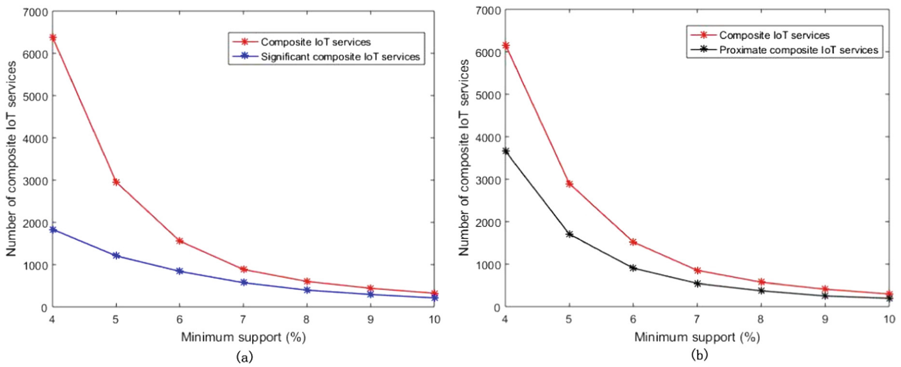


Fig. 6. Effectiveness of significance and proximity strategies

We perform the third set of experiments on Data3 to evaluate the applicability of our proposed approach. Table 1 shows the primary discovered composite IoT services. Some of the composite IoT services are indeed difficult to be discovered because they are less frequent. For example, the “lawn work” and “going out for entertainment” compositions occur only once during two weeks. Next, we check the discovered periodic compositions. Ideally, we want to associate one time interval with a composition. However, we discover that some IoT service compositions may associate multiple time intervals. For example, the “taking medication” service composition occurs in the morning and in the evening. This is a very practical issue. In this experiment, we group the discovered composition instances using a preliminary technique, that is, two time intervals are grouped together if they overlap. The tolerance threshold ζ is set to be 2 h. We can see from Table 1 that the resident performs some activities regularly. For example, one of the striking periodic activities is the “preparing breakfast”. There is 75% chance that he/she will “prepare breakfast” during 5:16 and 6:51.

We conduct the fourth set of experiments on Data3 to measure how much convenience can be obtained by applying the discovered results in Table 1. We showcase some preliminary results. For example, based on the representative time, the “preparing breakfast” is likely to be followed by “watching TV” and “watching TV” is likely to be followed by “going out for shopping”. Given the “preparing breakfast” activity on 4/22/2003, we can obtain 50% convenience.

Table 1. Primary discovered periodic composite IoT services from Data3

Periodic composite IoT services	Representative time intervals, location, probability
Taking medication	(1:54–2:00, in the kitchen, 0.5), (19:23–19:38, in the kitchen, 0.29)
Preparing breakfast	(5:45–6:20, in the kitchen, 0.75)
Preparing lunch	(11:15–12:05, in the kitchen, 0.64)
Preparing dinner	(18:15–18:48, in the kitchen, 0.5)
Going out for shopping	(7:52–8:05, in the hallway, 0.67)
Watching TV	(7:00–7:23, in the living room, 0.27), (14:27–15:22, in the living room, 0.33)

5 Related Work

A Web service mining framework is proposed to discover interesting composite services from available services [4]. This framework models Web services using ontologies and an efficient algorithm is proposed to discover composite services. In [24], a graphic model is proposed to represent the dependency among services. A general service mining framework is proposed based on an ontology service model [5]. Service relationships are established via ontology attributes. A Correlation Degree method is presented to evaluate the correlation strength among services. Most of existing work consider service relationships based on their input/output correlations, pre/post condition correlations etc. However, in the context of IoT, spatio-temporal relationships among IoT services are implicit and subtle. In [12], an efficient algorithm $CoPMiner$ is developed to mine the temporal relationships among appliances in the smart home environment. The key idea of $CoPMiner$ is to transform interval-based event sequences into endpoint based sequences. It also reformulates the problem of discovering temporal patterns among appliances as discovering frequent patterns from endpoint sequences. Location information regarding appliances is utilized to filter out insignificant temporal patterns. However, temporal distance is not considered in [12], which may result in undesirable frequent temporal patterns. In [9], an efficient algorithm IEMiner is proposed to discover temporal patterns for classification. In [2], a novel graph-based approach is proposed to capture the subtle relationships among things based on things' usage time, location, and users' social network information. The Random Walk with Restart method is applied to discover things relationships.

There are many research on human activity discovery. In [16], an efficient algorithm COM is proposed to discover human activity pattens from sensor event data. These patterns are used to build a HMM model for recognizing human activities. In [20], a probabilistic and Markov chain approach is proposed to discover complex human activity patterns. These patterns associated with context information are used to recognize activities. A general framework is proposed to address the problem of complex activity prediction by mining temporal sequence patterns from video [21]. A probabilistic suffix tree model is introduced to model activities. There have been little research into the human activity recognition which considers the periodic feature. For example, [22] discovers periodic activities from trajectory data such as staying in the office during daytime and staying at home in the evening.

6 Conclusion and Future Work

We addressed the problem of discovering periodic composite IoT services to provide personalized convenience to residents. An IoT service model and a composite IoT service model are proposed in terms of spatio-temporal aspects. The experimental results show our proposed *significance* and *proximity* strategies are effective in pruning non-promising composite IoT services. The periodic composite IoT service model is introduced and is applied to provide convenience. We introduce a new algorithm PCMiner to discover periodic composite IoT services. Future work includes improving the performance of PCMiner. Furthermore, we will apply the discovered periodic composite IoT services to build an intelligent system for providing convenience.

Acknowledgment. This research was partly made possible by NPRP 9-224-1-049 grant from the Qatar National Research Fund (a member of The Qatar Foundation) and DP160100149 and LE180100158 grants from Australian Research Council. The statements made herein are solely the responsibility of the authors.

References

1. Atzori, L., Iera, A.: The Internet of Things: a survey. J. Comput. Netw. **54**(15), 2787–2805 (2010)
2. Yao, L.: Unveiling correlations via mining human-thing interactions in the web of things. ACM Trans. Intell. Syst. Technol. **8**(5), 62 (2017)
3. Miorandi, D.: Internet of Things: vision, applications and research challenges. Ad Hoc Netw. **10**(7), 1497–1516 (2012)
4. Zheng, G., Bouguettaya, A.: Service mining on the web. IEEE Trans. Serv. Comput. **2**(1), 65–78 (2009)
5. Huang, B., Bouguettaya, A., Dong, H., Chen, L.: Service mining for Internet of Things. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 566–574. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_36
6. Ghari Neiat, A., Bouguettaya, A.: Crowdsourcing of Sensor Cloud Services, pp. 1–116. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91536-4
7. Issarny, V., Bouloukakis, G., Georgantas, N., Billet, B.: Revisiting service-oriented architecture for the IoT: a middleware perspective. In: Sheng, Q.Z., Stroulia, E., Tata, S., Bhiri, S. (eds.) ICSOC 2016. LNCS, vol. 9936, pp. 3–17. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46295-0_1
8. Bouguettaya, A., et al.: A service computing manifesto: the next 10 years. Commun. ACM **60**, 64–72 (2017)
9. Patel, D., Hsu, W.: Mining relationships among interval-based events for classification. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 393–404. ACM (2008)
10. Tapia, E.M., Intille, S.S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. In: Ferscha, A., Mattern, F. (eds.) Pervasive 2004. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24646-6_10
11. Shao, W., Bouguettaya, A.: Clustering big spatiotemporal-interval data. IEEE Trans. Big Data **2**, 190–203 (2016)
12. Chen, Y.-C., Chen, C.-C., Peng, W.-C., Lee, W.-C.: Mining correlation patterns among appliances in smart home environment. In: Tseng, V.S., Ho, T.B., Zhou, Z.-H., Chen, A.L.P., Kao, H.-Y. (eds.) PAKDD 2014. LNCS (LNAI), vol. 8444, pp. 222–233. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06605-9_19
13. Bouguettaya, A., et al.: End-to-end service support for mashups. IEEE Trans. Serv. Comput. **3**, 250–263 (2010)
14. Chen, Y.-C.: Significant correlation pattern mining in smart homes. ACM Trans. Intell. Syst. Technol. (TIST) **6**(3), 35 (2015)
15. Wu, Q.: Cognitive internet of things: a new paradigm beyond connection. IEEE Internet of Things J. **1**, 129–143 (2014)
16. Rashidi, P., Cook, D.J.: COM: a method for mining and monitoring human activity patterns in home-based health monitoring systems. ACM Trans. Intell. Syst. Technol. (TIST) **4**(4), 64 (2013)
17. Meyer, S.: A survey of research on context-aware homes. In: Proceedings of the ACSW, pp. 159–168 (2003)
18. Singla, G., Cook, D.J.: Tracking activities in complex settings using smart environment technologies. Int. J. Biosci. Psychiatry Technol. (IJBSPT) **1**(1), 25 (2009)
19. De Souza, R.M.C.R., De Carvalho, F.A.T.: Clustering of interval data based on cityblock distances. Pattern Recognit. Lett. **25**(3), 353–365 (2004)

20. Saguna, S., Zaslavsky, A.: Complex activity recognition using context-driven activity theory and activity signatures. ACM Trans. Comput.-Hum. Interact. (TOCHI) **20**(6), 32 (2013)
21. Li, K., Fu, Y.: Prediction of human activity by discovering temporal sequence patterns. IEEE Trans. Pattern Anal. Mach. Intell. **36**, 1644–1657 (2014)
22. Li, Z.: Mining periodic behaviors for moving objects. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1099–1108(2010)
23. Pei, J.: Mining sequential patterns by pattern-growth: the prefixspan approach. IEEE Trans. Knowl. Data Eng. **16**(11), 1424–1440 (2004)
24. Huang, F., Smidts, C.: Causal mechanism graph – a new notation for capturing cause-effect knowledge in software dependability. Reliab. Eng. Syst. Saf. **158**, 196–212 (2017)
25. Neiat, A.G.: Crowdsourced coverage as a service: two-level composition of sensor cloud services. IEEE Trans. Knowl. Data Eng. **29**, 1384–1397 (2017)