A Framework for Personal Mobile Commerce Pattern Mining and Prediction

Eric Hsueh-Chan Lu, Wang-Chien Lee, and Vincent S. Tseng

Abstract—Due to a wide range of potential applications, research on mobile commerce has received a lot of interests from both of the industry and academia. Among them, one of the active topic areas is the mining and prediction of users' mobile commerce behaviors such as their movements and purchase transactions. In this paper, we propose a novel framework, called *Mobile Commerce Explorer* (*MCE*), for mining and prediction of mobile users' movements and purchase transactions under the context of mobile commerce. The *MCE* framework consists of three major components: 1) *Similarity Inference Model* (*SIM*) for measuring the similarities among stores and items, which are two basic mobile commerce entities considered in this paper; 2) *Personal Mobile Commerce Pattern Mine* (*PMCP-Mine*) algorithm for efficient discovery of mobile users' *Personal Mobile Commerce Patterns* (*PMCPs*); and 3) *Mobile Commerce Behavior Predictor* (*MCBP*) for prediction of possible mobile user behaviors. To our best knowledge, this is the first work that facilitates mining and prediction of mobile users' commerce behaviors in order to recommend stores and items previously unknown to a user. We perform an extensive experimental evaluation by simulation and show that our proposals produce excellent results.

Index Terms—Data mining, Mobile commerce.

---------- ◆ ----------

1 INTRODUCTION

WITH the rapid advance of wireless communication technology and the increasing popularity of powerful portable devices, mobile users not only can access worldwide information from anywhere at any time [28] but also use their mobile devices to make business transactions easily, e.g., via digital wallet. Meanwhile, the availability of location-acquisition technology, e.g., Global Positioning System (GPS), facilitates easy acquisition of a moving trajectory, which records a user movement history. Thus, we envisage that, in the coming future of Mobile Commerce (M-Commerce) age [27], some m-commerce services will be able to capture the moving trajectories and purchase transactions of users. Take the recent announced Shopkick [20] as an example, it gives mobile users rewards and offers when users checkin in stores and on items. Anticipating that some users may be willing to exchange their locations and transactions for good rewards and discounts, we expect more mobile commerce applications, whether they will bear a business model similar with Shopkick or not, will appear in the future. In this paper, we aim at developing pattern mining and prediction techniques that explore the correlation between the moving behavior and purchasing transactions of mobile users to explore potential M-Commerce features.

Owing to the rapid development of the Web 2.0 technology, many stores have made their store information, e.g., business hours, location, and features available online, e.g., via mapping services such as Google Map. Additionally, user trajectories can be detected by GPS-enabled devices, when users move around. For example, in [31], [33], the authors discuss how to collect and analyze user trajectories from GPS-enabled devices. When a user enters a building, the user may lose the satellite signal until returning to the outdoors. By matching user trajectories with store location information, a user's moving sequence among stores in some shop areas can be extracted. Fig. 1 shows a scenario, where a user moves among stores while making some purchase transactions (or transactions in short). Fig. 1(a) shows a moving sequence, where underlined store labels indicate some transactions being made there. Fig. 1(b) shows the transaction records of a user, where item i_1 was purchased when this user is in store A. The mobile transaction sequence generated by this user is {(A, {i_1}), (B, $\varnothing$), (C, {i_3}), (D, {i_2}), (E, $\varnothing$), (F, {i_3, i_4}), (I, $\varnothing$), (K, {i_5})}.

There usually is an entangling relation between moving patterns and purchase patterns since mobile users are

---

- *Eric Hsueh-Chan Lu is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. E-mail: eric@idb.csie.ncku.edu.tw*
- *Wang-Chien Lee is with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802, USA. E-mail: wlee@cse.psu.edu*
- *Vincent S. Tseng is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C. E-mail: tsengsm@mail.ncku.edu.tw*
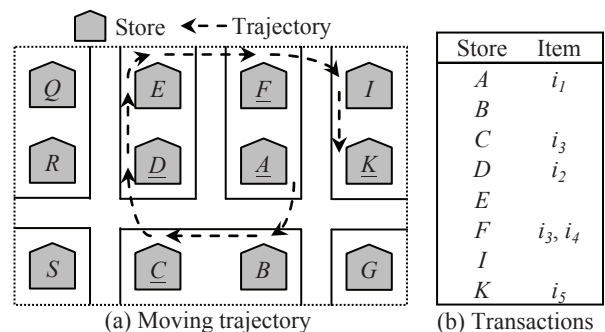
Fig. 1. An example for a mobile transaction sequence.

moving between stores to shop for desired items [30]. The moving and purchase patterns of a user can be captured together as *mobile commerce patterns* for mobile users. For example, the user taking the shopping trip shown in Fig. 1 may exhibit a moving pattern ABC and two purchase patterns $(A, \{i_1\})$ and $(C, \{i_3\})$. This pattern, which can be expressed as $\{(A, \{i_1\}) \xrightarrow{B} (C, \{i_3\})\}$, indicates that the user usually purchases item i_1 in store A and then purchases item i_3 in store C on the specific path ABC. Armed with knowledge of this pattern, an m-commerce service could push some discount coupons of item i_3 to the user to boost the sales of store C when the user purchases item i_1 in store A. To provide this mobile ad hoc advertisement, mining mobile commerce patterns of users and accurately predicts their potential mobile commerce behaviors obviously are essential operations that require more research. They are also presented in this paper.

To capture and obtain a better understanding of mobile users' mobile commerce behaviors, data mining [7] has been widely used for discovering valuable information from complex data sets. A number of studies have discussed the issue of mobile behavior mining analysis, even though the targeted patterns in these prior works are typically different. For example, Tseng *et al.* [26] studied the problem of mining associated service patterns in mobile web environments. They also proposed SMAP-Mine [23] for efficient mining of users' sequential mobile access patterns, based on the FP-Tree [8]. Chen *et al.* [5] proposed the path traversal patterns for mining mobile web user behaviors. Yun *et al.* [30] proposed a method for mining mobile sequential pattern (MSP) by taking moving paths of users into consideration. Jeung *et al.* proposed a prediction approach called Hybrid Prediction Model (HPM) [12] for mining the trajectory pattern of a moving object. While the aforementioned studies have been conducted for discovery of mobile patterns, few of them consider the personalization issue. Since patterns mined in these studies are typically from all users, they do not reflect the personal behaviors of individual users, especially when the mobile behaviors may vary a lot amongst different mobile users. In this paper, we aim at mining mobile commerce behavior of individual users to support m-commerce services at a personalized level.

As mentioned earlier, in addition to mining mobile patterns, predicting the next mobile behaviors of a user is a critical research issue. Existing work on mobile behavior prediction can be roughly divided into two categories. The first category is vector based prediction [18], [21], [22] and the second category is pattern based prediction [10], [12], [23], [30]. The idea of vector based prediction is to predict the next location of an object according to its moving direction and velocity. Vector based predictions assume that the predictive mobile behaviors of a user can be represented by mathematical models based on his recent movement in form of geographic information. Pattern based prediction models, on the other hand, capture semantic patterns that match the user's recent mobile behaviors well. Pattern based predictions are more precise than vector based predictions [12]. Hybrid Prediction Model (HPM) [12] represents the state of the art in the field of movement prediction for moving objects. HPM integrates both ideas of the pattern based prediction and vector based prediction. We argue that the vector based prediction models may not be appropriate for mobile user behavior prediction, since an object's movements are more complicated than what the mathematical formulas can represent [12]. Thus, our study follows the paradigm of pattern based prediction. Nevertheless, our work is uniquely different from the existing work because we aim at predicting the mobile commerce behavior in terms of both the movement and purchase transaction, while the existing work mostly focus on predict the movement only.

A crucial issue for pattern based prediction is that the predictions fail if there is no existing pattern to match. In the previous pattern based prediction models, pattern selection is typically based on exact matching, e.g., the similarity between different stores is 0. Take Fig. 1 as an example, the user has never been to store Q, store R, and store S. Since there is no pattern involving these stores, pattern based predictions do not work when a user first moves to these stores. To overcome this problem, our idea is to incorporate the similarities of stores and items into the mobile commerce behavior prediction. Consider the example in Fig. 1 again. Since the user has never been to store Q, the mobile patterns mined by this user do not contain any information about store Q. However, if we know that store A (where the user had visited before) is similar with store Q, we can make recommendation to the user based on the patterns exhibited in store A. In other words, we consider that the mobile behaviors of the user in store A may be similar with those in store Q. Thus, we can employ the inferred behaviors in store A to predict next mobile behaviors in store Q even though the user has never been to store Q. Hence, a fundamental issue is to derive the similarities of stores in this paper.

Multiple-level hierarchical structures can be defined to measure which stores are similar [6], [15], [26]. However, the method requires the users to set up hierarchical structures. It is difficult to determine suitable structures in a mobile commerce environment. In this paper, we develop a similarity inference model to automatically measure the similarities between stores and between items. Based on our observations, we identify two basic heuristics as the bases of our inference model: (1) two stores are similar if the items they sell are similar; (2) two items are dissimilar if the stores which sell them are dissimilar. Accordingly, we infer the store similarity and item similarity from each other. Although a number of similarity measures have been studied to measure the similarity of two vectors in the literature, they are not applicable in this work due to the following factors: (1) most of similarity measures can only process numerical data [13] but not the categorical data considered in this paper; (2) most of similarity measures consider the similarity between two vectors as 0, when their elements are different [13], [29]. It is not true in this work. For example, there are two stores A and B which only sell milk and coffee, respectively. The similarity between store A and store B should not be 0 since milk and coffee are both drinks; and (3) most of the similarity measures can only handle one entity type, while we

ERIC HSUEH-CHAN LU, WANG-CHIEN LEE, AND VINCENT S. TSENG: A FRAMEWORK FOR PERSONAL MOBILE COMMERCE PATTERN MINING AND PREDICTION

3

consider both store similarity and item similarity at the same time. In [11], Jeh *et al.* propose an iterative similarity computation method named SimRank. Although SimRank bears with similar ideas as SIM, SimRank is not applicable to our problem. Particularly, SimRank needs to set a decay factor C and a fixed number of iterations K to perform. In mobile commerce environments, it is difficult to determine which parameters are suitable.

To provide a high-precision mobile commerce behavior predictor, we focus on personal mobile pattern mining. Besides, to overcome the predictions failure problem, we incorporate the similarities of stores and items into the mobile commerce behavior prediction. Hence, in this paper, we propose a novel framework, namely *Mobile Commerce Explorer* (*MCE*), to mine and predict mobile users' movements and transactions under the context of mobile commerce. The *MCE* framework consists of three major components: 1) *Similarity Inference Model* (*SIM*) for measuring the similarities among stores and items, which are two basic mobile commerce entities considered in this paper; 2) *Personal Mobile Commerce Pattern Mine* (*PMCP-Mine*) algorithm for efficient discovery of mobile users' *Personal Mobile Commerce Patterns* (*PMCPs*); and 3) *Mobile Commerce Behavior Predictor* (*MCBP*) for prediction of possible mobile user behaviors. To our best knowledge, this is the first work that facilitates mining and prediction of mobile users' commerce behaviors that may recommend stores and items previously unknown to a user. Finally, through an extensively experimental evaluation, we show that our proposals deliver an excellent performance in terms of precision, recall and F-measure.

The advantages and contributions of this paper are five-fold.

- We propose the MCE framework, a new approach for mobile commerce behavior mining and prediction. The problems and ideas in MCE have not been well explored in the research community.
- We propose a novel model SIM for automatically measuring the similarities among stores and items from a mobile transaction database.
- To understand the personal mobile behaviors, we propose a novel algorithm PMCP-Mine for mining PMCPs from a mobile transaction database.
- Based on the SIM and PMCP-Mine, we propose a novel prediction technique MCBP for precisely prediction of the mobile behaviors which include movements and transactions of a user.
- We design a simulation model and conduct a series of experiments to evaluate the performance of our proposal. The results show superior performance over other mining techniques in terms of predictive precision and recall.

The remainder of this paper is organized as follows. We briefly review the related work in Section 2. In Section 3, we formulate the problem. In Section 4, we first introduce the proposed framework MCE. Then, we describe the proposed approaches SIM, PMCP-Mine, and MCBP. In Section 5, we perform an empirical performance evaluation. Finally, in Section 6, we summarize our conclusions and future work.

2 RELATED WORK

In this section, we review and classify relevant previous studies into three categories: 1) similarity measures, 2) mobile pattern mining techniques, and 3) mobile behavior predictions.

Similarity Measure. There have been many studies on measuring the similarity between two objects. The first one is based on multiple-level hierarchical structures. In [15], Lu first proposes the concept of multiple-level hierarchical structure in data mining. In [6], Han *et al.* propose the multiple-level association rules mining. In this study, taxonomy is incorporated for representing the hierarchical relations of items. In [26], Tseng *et al.* first applies the multiple-level hierarchical concept to mine associated service patterns in mobile web environments. Based on the structure, the items in the same level are regarded as similar items. However, we do not know the relations between the items in the different levels. The second one is sequence alignments. In [11], Jeh *et al.* propose the SimRank to iteratively compute the similarities between objects. The idea is that two objects are similar if they are related to similar objects. To improve the efficiency of SimRank, in [32], Yin *et al.* develop the hierarchical structure named SimTree to reduce the computation cost and the storage of object similarities but still discover the relationships between objects. In [29], Xin *et al.* propose a pattern distance measure based on set similarity between two association patterns. The concept of set similarity is to apply Jaccard Measure to calculate the similarity of two sets. Let S_1 and S_2 be two sets, the set similarity $set\_similarity(S_1, S_2)$ is defined as (1). However, set similarity is not applicable to store similarity in mobile commerce. For example, there are two stores A and B which only provides milk and coffee, respectively. The similarity of store A and store B should not be 0, since milk and coffee belong to the same drink category.

$$set\_similarity(S_1, S_2) = \frac{S_1 \cap S_2}{S_1 \cup S_2} \tag{1}$$

Mobile Pattern Mining. In recent years, a number of studies have discussed the usage of data mining techniques to discover useful rules/patterns from WWW [19], transaction databases [1], [2], [3], [8], [17] and mobility data [14], [23], [26], [30]. Mining association rules [1] are proposed to find important items in a transaction database. In [2], Agrawal *et al.* propose the Apriori algorithm to mine the association rules. In [17], Park *et al.* propose the DHP algorithm to improve the performance of association rule mining. In [19], Pei *et al.* propose an algorithm named WAP-Mine to efficiently discover web access patterns in web logs, using a tree-based data structure without candidate generation. Sequential pattern mining has been first introduced in [3] to search for time-ordered patterns, known as sequential patterns within transaction databases. For the studies considering the relation between location and service, in [5], Chen *et al.* propose the path traversal patterns for mining web user behaviors. Tseng *et al.* [26] first study the problem of mining associated service patterns in mobile web environments. SMAP-Mine [23] has been proposed by Tseng *et al.* for

efficiently mining users' sequential mobile access patterns, based on the FP-Tree [8]. Lee *et al.* propose T-MAP [14] to efficiently find the mobile users' mobile access patterns in distinct time intervals. Yun *et al.* propose the Mobile Sequential Pattern (MSP) [30] to take moving paths into consideration and add the moving path between the left hand and the right hand in the content of rules. In [25], Tseng *et al.* propose the TMSP-Mine for discovering the temporal mobile sequence patterns in a location-based service environment. Jeung *et al.* propose a prediction approach called Hybrid Prediction Model (HPM) [12] for estimating an object's future locations based on its pattern information. This paper considers that an object's movements are more complicated than what the mathematical formulas can represent. However, there is no work consider user relations in the mobile pattern mining.

Mobile Behavior Prediction. The studies on mobile behavior predictions can be roughly divided into two categories. The first category is vector based prediction that can be further divided into two types: (1) linear models [18], [22] and (2) non-linear models [21]. The non-linear models capture objects' movements with sophisticate regression functions. Thus, their prediction accuracies are higher than those of the linear models. Recursive Motion Function (RMF) [21] is the most accurate prediction method in the literature based on regression functions. The second category is pattern based prediction. In [10], Ishikawa *et al.* derive a Markov Model (MM) that generates Markov transition probabilities from one cell to another for predicting the next cell of the object. In HPM [12], the form of a trajectory pattern is $R_{t1}^{j1} \wedge R_{t2}^{j2} \wedge ... \wedge R_{tm}^{jm} \rightarrow R_{tn}^{jn}$ with a confidence c, where R_{ti}^{ji} indicates a user in location R_{ti}^{ji} at time t_i, i.e., when the premise of the pattern occurs, the consequence will also occur with probability c. However, these methods can only predict the next spatial locations of objects. SMAP-Mine [23] has been proposed to discover sequential mobile access rules and predict the user's next locations and services. The form of the rule is $\{(r_i, s_i)\} \rightarrow \{(r_j, s_j)\}$ with a confidence c, where r_i and r_j are locations, and s_i and s_j are services. It implies that a user requesting s_i in r_i will have next location and service as r_j and s_j with c probability. In [30], Yun *et al.* propose the Mobile Sequential Pattern (MSP) to predict the next mobile behaviors. The form of the pattern is $\{(r_i, s_i), (r_1), (r_2), (r_3), (r_j, s_j)\}$, where item (r_i, s_i) indicates a user request service s_i at location r_i. The pattern above means that a user requests service s_i in location r_i and then requests service s_j in location r_j via a specific path $r_1 r_2 r_3$.

The idea of Collaborative Filtering (CF) [9] may be applied to the prediction of user's behavior. Collaborative filtering can be divided into two types: 1) user-based collaborative filtering and 2) item-based collaborative filtering. The user-based collaborative filtering is based on the behaviors of other similar users. For example, suppose that John and Bob are similar based on their profiles or preferences. We may refer to the behaviors of Bob to predict the next behavior of John. However, the behaviors of two users are not always similar even if the two users are very similar. For the item-based collaborative filtering, the prediction concept is based on user behavior asso-

ciated with similar items. For example, suppose that Coffee and Milk are similar based on their item categories or properties. When we know that a user has purchased Coffee and we try to predict the next behavior of this user, we may refer to the next behavior after this user purchases Milk. However, it is difficult to define the similarity between items. Generally speaking, collaborative filtering techniques rely on user ratings on items to predict user purchase behavior and are not applicable to our study.

3 PROBLEM FORMULATION

In this section, we first define some terms used in discussion of our research work and then specify our research goal. Table 1 summarizes the notations used in the paper. Let $MTS = <(s_1, I_1), (s_2, I_2), ..., (s_n, I_n)>$ be a mobile transaction sequence of length n for a user, where (s_t, I_t) denotes that the user purchases the item set I_t in the store s_t, $\forall 1 \le t \le n$. The elements of the sequence are in ascending order of time.

Definition 1. A mobile transaction sequence $MTS' = <(s_1', I_1'), (s_2', I_2'), ..., (s_m', I_m')>$ is a *sub-pattern* of another mobile transaction sequence $MTS = <(s_1, I_1), (s_2, I_2), ..., (s_n, I_n)>$, denoted as $MTS' \subset MTS$, if $m \le n$ and there exists a strictly increasing sequence $(k_1, k_2, ..., k_m)$ of indices, such that for all $j = 1, 2, ..., m$, $s_j' = s_{k_j}$ and $I_j' \subseteq I_{k_j}$. Here, MTS is called the *super-pattern* of MTS'. For example, let $MTS_1 = <(A, \{i_1, i_2\}), (B, \varnothing), (C, \varnothing), (D, \{i_3\})>$ and $MTS_2 = <(A, \{i_1\}), (B, \varnothing), (D, \{i_3\})>$, MTS_2 is a *sub-pattern* of MTS_1.

Definition 2. Given a personal mobile transaction database $D_u = \{MTS_1, MTS_2, ..., MTS_z\}$ for user u where D_u contains z MTSs, the support of a mobile transaction sequence MTS (denoted as $sup(MTS)$) is defined in (2) as shown below. A MTS is called a *frequent MTS*, if $sup(MTS)$ is higher than or equal to a specified support threshold δ.

$$sup(MTS) = \frac{\left| \{MTS_i \mid MTS \subset MTS_i, 1 \le i \le z\} \right|}{z} \qquad (2)$$

Definition 3. A frequent $MTS = <(s_1, I_1), (s_2, I_2), ..., (s_n, I_n)>$ is called a *moving path P*, if $I_t = \varnothing$, $\forall 1 \le t \le n$, and denoted as $P = <s_1, s_2, ..., s_n>$. For example, given a $MTS = <(A, \varnothing), (B, \varnothing), (C, \varnothing), (D, \varnothing)>$, the moving path P of MTS is $<A, B, C, D>$.

TABLE 1
NOTATIONS.

Notation	Description
MTD	Mobile transaction database.
D_u	Mobile transaction database for user u, $D_u \in MTD$.
MTS	Mobile transaction sequence, $MTS \in D_u$.
P	Moving path.
$PMCP$	Personal mobile commerce pattern.
s	A store.
S	A set of stores.
Ω_i	The set of stores where the item i is sold.
i	An item.
I	A set of items.
Γ_s	The set of items sold in the store s.
SID	Store-Item Database.
ISD	Item-Store Database.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

ERIC HSUEH-CHAN LU, WANG-CHIEN LEE, AND VINCENT S. TSENG: A FRAMEWORK FOR PERSONAL MOBILE COMMERCE PATTERN MINING AND PREDICTION

5

Definition 4. The *Personal Mobile Commerce Pattern* (*PMCP*) of a frequent *MTS* = <(s_1, I_1), (s_2, I_2), …, (s_n, I_n)> discovered from a given user u (denoted as $PMCP_u$) is defined in Equation (3).

$$PMCP_u = u :< (s_{k_1}, I_{k_1}) \xrightarrow{P_1} (s_{k_2}, I_{k_2}) \xrightarrow{P_2} ... \xrightarrow{P_{r-1}} (s_{k_r}, I_{k_r}) > \quad (3)$$

where $r \le n$, $k_j < k_{j+1}$, $I_{k_j} \neq \varnothing$, $\forall 1 \le j \le r$, and P_v is the *moving path* between s_{k_v} and $s_{k_{v+1}}$, $\forall 1 \le v \le r$ - 1. For example, let *MTS* = <(A, {i_1, i_2}), (B, $\varnothing$), (C, $\varnothing$), (D, {i_3}), (E, $\varnothing$), (F, {i_4})> which is mined from the user u. The $PMCP_u$ of *MTS* is (A, {i_1, i_2}) $\xrightarrow{BC}$ (D, {i_3}) $\xrightarrow{E}$ (F, {i_4}).

Definition 5. A *Store-Item Database* (*SID*), maintaining information about each store s and a set of items Γ_s sold in the store s, is denoted as follows.

$$SID = \{< s_1, \Gamma_{s_1} >, < s_2, \Gamma_{s_2} >, ..., < s_\alpha, \Gamma_{s_\alpha} >\}$$

where α indicates the total number of stores.

Definition 6. A *Item-Store Database* (*ISD*), maintaining information about each item i and a set of stores Ω_i where the item i is sold, is denoted as follows.

$$ISD = \{< i_1, \Omega_{i_1} >, < i_2, \Omega_{i_2} >, ..., < i_\beta, \Omega_{i_\beta} >\}$$

where β indicates the total number of items.

Definition 7. Let s_p and s_q be two stores. The similarity of s_p and s_q is denoted by $sim(s_p, s_q)$, and $0 \le sim(s_p, s_q) \le 1$.

Definition 8. Let i_x and i_y be two items. The similarity of i_x and i_y is denoted by $sim(i_x, i_y)$, and $0 \le sim(i_x, i_y) \le 1$.

Research Objective. The goal of this study is to develop a framework for mining and prediction of the mobile commerce behaviors, including the movements and transactions of a user, based on the user's current mobile transaction sequence.

4 PROPOSED METHOD

In this section, we describe our design of a personal mobile commerce mining and prediction framework, called *Mobile Commerce Explorer* (*MCE*), which incorporates three innovative techniques, including 1) *Similarity Inference*

Fig. 2. The mobile commerce explorer framework.

TABLE 2
AN EXAMPLE OF MOBILE TRANSACTION DATABASE.

T_{id}	U_{id}	Mobile Transaction Sequence
1	1	(A,{i_1}), (B,$\varnothing$), (C,{i_3}),(D,{i_2}), (E,$\varnothing$), (F,{i_3,i_4}), (I,$\varnothing$), (K,{i_5})
2	1	(A,{i_1}), (B,$\varnothing$), (C,$\varnothing$), (D,{i_2})
3	1	(A,{i_1}), (B,$\varnothing$), (C,$\varnothing$), (D,{i_2}), (E,$\varnothing$), (F,{i_3,i_4}), (I,$\varnothing$), (K,{i_3})
4	1	(A,{i_1}), (D,{i_6}), (C,{i_5})
5	2	(A,{i_1}), (E,$\varnothing$), (F,$\varnothing$), (K,{i_2}), (I,{i_2})
6	2	(B,{i_3}), (A,{i_1}), (E,$\varnothing$), (F,$\varnothing$), (K,{i_2})
7	2	(A,{i_1}), (E,$\varnothing$), (F,$\varnothing$), (K,{i_2}), (I,$\varnothing$)
8	2	(A,{i_1}), (E,$\varnothing$), (F,{i_3}), (K,{i_2}), (I,{i_8})
9	3	(B,{i_1}), (A,$\varnothing$), (E,{i_3}), (D,$\varnothing$), (E,$\varnothing$)
10	3	(B,$\varnothing$), (A,$\varnothing$), (E,$\varnothing$), (D,$\varnothing$), (B,{i_1}), (D,{i_7})
11	3	(B,{i_1}), (A,$\varnothing$), (E,{i_3}), (D,$\varnothing$)
12	4	(D,{i_4}), (B,$\varnothing$), (A,{i_3})
13	4	(I,{i_5}), (F,$\varnothing$), (E,$\varnothing$), (D,{i_4})
14	4	(I,{i_6}), (F,$\varnothing$), (E,{i_1}), (D,{i_4})

Model (*SIM*) for measuring the similarities among stores and items, which are two basic mobile commerce entities considered in this paper; 2) *Personal Mobile Commerce Pattern Mine* (*PMCP-Mine*) algorithm for efficient discovery of mobile users' *Personal Mobile Commerce Patterns* (*PMCPs*); and 3) *Mobile Commerce Behavior Predictor* (*MCBP*) for prediction of possible mobile user behaviors.

4.1 System Framework

The proposed MCE framework consists of three modules, 1) a mobile network database, 2) a data mining mechanism, and 3) a behavior prediction engine (See Fig. 2). The mobile network database maintains detailed store information which includes locations. Our system has an "offline" mechanism for similarity inference and PMCPs mining, and an "online" engine for mobile commerce behavior prediction. When mobile users move between the stores, the mobile information which includes user identification, stores, and item purchased are stored in the mobile transaction database. Table 2 shows an example of mobile transaction database which contains 4 users and 14 mobile transaction sequences. In the offline data mining mechanism, we develop the SIM model and the PMCP-Mine algorithm to discover the store/item similarities and the PMCPs, respectively. In the online prediction engine, we propose a mobile commerce behavior predictor (MCBP) based on the store and item similarities as well as the mined PMCPs. When a mobile user moves and purchases items among the stores, the next steps will be predicted according to the mobile user's identification and recent mobile transactions. The framework is to support the prediction of next movement and transaction.

4.2 Similarity Inference Model (SIM)

An essential task in our framework is to determine the similarities of stores and items. The problem may be solved by using store and item category ontology. However, the store or item ontology may not match with the mobile transaction database. Our goal is to automatically compute the store and item similarities from the mobile transaction database, which captures mobile users' moving and transactional behaviors (in terms of movement among stores and purchased items). From the database,
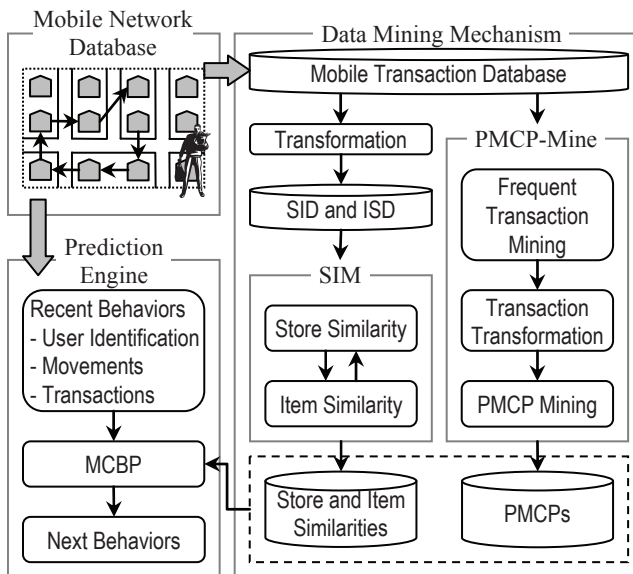
we have the following information available: 1) for a given store, we know which items are available for sale; 2) for a given item, we know which stores sell this item. The information can help us to infer which stores or items are similar. As we observe that people usually purchase similar items in certain stores, these stores may be considered as similar. For example, people may purchase hamburgers, French fries, or Cokes in McDonalds and Burger King, we consider them as similar stores. We propose a parameter-less data mining model, named *Similarity Inference Model* (*SIM*), to tackle this task of computing store and item similarities.

Before computing the SIM, we derive two databases, namely, *Store-Item Database* (*SID*) and *Item-Store Database* (*ISD*), from the mobile transaction database. An entry SID_{pq} in database *SID* represents that a user has purchased item q in store p, while an entry ISD_{xy} in database *ISD* represents that a user has purchased item x in store y. Table 3 shows the transformed *SID* and *ISD* from mobile transaction database in Table 2. There are 8 stores and 8 items in this database. After obtaining *SID* and *ISD*, the major challenge we have to tackle on is to automatically compute the similarities between stores and items.

We derive the Similarity Inference Model (SIM) to capture the similarity score between stores/items. For every pair of stores or items, SIM assigns them a similarity score. As shown in Fig. 3(a), s_1 and s_2 are two stores and Γ_{s_1} and Γ_{s_2} are two item sets which are sold in stores s_1 and s_2, respectively. In Fig. 3(b), i_1 and i_2 are two items and Ω_{i_1} and Ω_{i_2} are two store sets where users have purchased i_1 and i_2, respectively. Based on our observations, we identify two basic heuristics to serve as the basis of our similarity inference model: 1) s_1 and s_2 are more similar, if Γ_{s_1} and Γ_{s_2} are more similar. For example, Mcdonalds and Burger King are similar since their provided items are similar, e.g., hamburgers, French fries, and Cokes. 2) i_1 and i_2 are more dissimilar, if Ω_{i_1} and Ω_{i_2} are more dissimilar. For example, the stores {Mcdonalds, Burger King, and KFC} and {Hang Ten, Giordano, and G2000} are dissimilar since the former is food-related stores and the latter is clothes-related stores. The items sold between these two kinds of stores are usually different, e.g., hamburger and shirt. Although SimRank which is similar to SIM has been proposed in [11], it is not applicable to the store similarity inference. In SimRank, the similarity between two given objects is measured based on the average similarities between other objects linked with the given two

(a) Store similarity inference   (b) Item similarity inference

Fig. 3. The basic concept of SIM.

TABLE 3
SID AND ISD.

Store	Items	Item	Stores
A	i_1, i_3	i_1	A, B, E
B	i_1, i_5	i_2	D, I, K
C	i_3, i_5	i_3	A, C, E, F
D	i_2, i_4, i_6, i_7	i_4	D, F
E	i_1, i_3	i_5	B, C, I, K
F	i_3, i_4	i_6	D, I
I	i_2, i_5, i_6, i_8	i_7	D
K	i_2, i_5	i_8	I

objects. As a result, two supermarkets selling a number of different items may be considered as dissimilar in SimRank. In SIM, we use two different inference heuristics for the similarity of stores and items because some stores, such as supermarkets, may provide various types of items. If we apply the same similarity inference heuristics to both of stores and items, various types of items may be seen as similar since different supermarkets are seen as similar. Based on our heuristics, if two stores provide many similar items, they are likely to be similar; if two items are sold by many dissimilar stores, they are unlikely to be similar. Since the store similarity and item similarity are inter-dependent, we compute them iteratively. In the following, we discuss the computational model.

For the store similarity, we consider that two stores are more similar if their provided items are more similar. Given two stores s_p and s_q, we compute their similarity $sim(s_p, s_q)$ by calculating the average similarity of item sets provided by s_p and s_q. For every item sold in s_p (and respectively s_q), we first find the most similar item sold in s_q (and respectively s_p). Then, the store similarity can be obtained by averaging all similar item pairs. Therefore, $sim(s_p, s_q)$ is defined as (4).
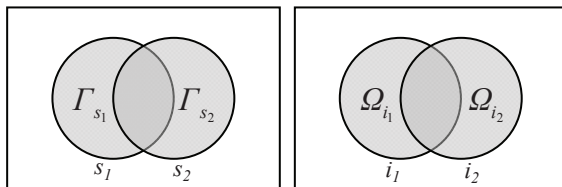
$$sim(s_p, s_q) = \frac{\sum_{\varphi \in \Gamma_{s_p}} MaxSim(\varphi, \Gamma_{s_q}) + \sum_{\gamma \in \Gamma_{s_q}} MaxSim(\gamma, \Gamma_{s_p})}{\left|\Gamma_{s_p}\right| + \left|\Gamma_{s_q}\right|} \quad (4)$$

where $MaxSim(e, E) = Max_{e' \in E} sim(e, e')$ represents the maximal similarity between e and the element in E. Γ_{s_p} and Γ_{s_q} are the sets of items sold in s_p and s_q, respectively.

On the other hand, for the item similarity, we consider that two items are less similar if they are sold by many dissimilar stores. Given two items i_x and i_y, we compute their similarity $sim(i_x, i_y)$ by calculating the average dissimilarity of store sets that provide i_x and i_y. For every store providing i_x (and respectively i_y), we first find the most dissimilar store that provides i_y (and respectively i_x) to obtain the item similarity by averaging all dissimilar store pairs. Therefore, $sim(i_x, i_y)$ is defined as (5).

$$sim(i_x, i_y) = 1 - \frac{\sum_{\omega \in \Omega_{i_x}} MinSim(\omega, \Omega_{i_y}) + \sum_{\psi \in \Omega_{i_y}} MinSim(\psi, \Omega_{i_x})}{\left|\Omega_{i_x}\right| + \left|\Omega_{i_y}\right|} \quad (5)$$

where $MinSim(e, E) = Min_{e' \in E} (1 - sim(e, e'))$ represents the minimal dissimilarity between e and the element in E. Ω_{i_x} and Ω_{i_y} are the sets of stores sell i_x and i_y, respectively.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

ERIC HSUEH-CHAN LU, WANG-CHIEN LEE, AND VINCENT S. TSENG: A FRAMEWORK FOR PERSONAL MOBILE COMMERCE PATTERN MINING AND PREDICTION

7

TABLE 4
STORE SIMILARITY AND ITEM SIMILARITY.

	A	B	C	D	E	F	I	K
A	1	0.6	0.653	0.185	1	0.656	0.173	0.253
B	0.6	1	0.739	0.281	0.6	0.303	0.52	0.764
C	0.653	0.739	1	0.326	0.653	0.563	0.517	0.759
D	0.185	0.281	0.326	1	0.185	0.494	0.649	0.461
E	1	0.6	0.653	0.185	1	0.656	0.173	0.253
F	0.656	0.303	0.563	0.494	0.656	1	0.202	0.243
I	0.173	0.52	0.517	0.649	0.173	0.202	1	0.68
K	0.253	0.764	0.759	0.461	0.253	0.243	0.68	1

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
i_1	1	0.19	0.584	0.276	0.39	0.164	0.124	0.168
i_2	0.19	1	0.222	0.346	0.665	0.608	0.387	0.505
i_3	0.584	0.222	1	0.549	0.372	0.201	0.156	0.164
i_4	0.276	0.346	0.549	1	0.227	0.421	0.509	0.221
i_5	0.39	0.665	0.372	0.227	1	0.467	0.197	0.463
i_6	0.164	0.608	0.201	0.421	0.467	1	0.51	0.666
i_7	0.124	0.387	0.156	0.509	0.197	0.51	1	0.256
i_8	0.168	0.505	0.164	0.221	0.463	0.666	0.256	1

As discussed above, the store similarity can be inferred if we know the item similarity, and vice versa. Based on the inference, we compute both the store similarity and the item similarity iteratively. Initially, the similarities between the same stores and the same items both are 1, otherwise, they are 0. In each iterative computation, SIM first uses the item similarity to compute store similarity, and then re-computes the item similarity from the store similarity. Then, all the values in the store or item similarity matrix are normalized to the value range between 0 and 1. The inference process stops when the computation reaches a stable state, which is decided by the differences of store similarity and item similarity. If the differences only change a little after an iteration step, SIM will stop. Since the numbers of stores or items may be large, the computation cost of SIM may be high. To reduce the running time, the efficiency of SIM can be improved by parallel processing techniques. Table 4 shows the final store similarity and item similarity from *SID* and *ISD* in Table 3.

4.3 Discovery of PMCPs

In this section, we describe the PMCP-Mine algorithm to mine the personal mobile commerce patterns efficiently. The PMCP-Mine algorithm is inspired by the TJ_{PF} algorithm [30] which is an Apriori-like algorithm. However, we observe that the TJ_{PF} algorithm does not consider user identification, which is essential for discovering *personal* mobile behaviors. In other words, the TJ_{PF} algorithm cannot be employed in our framework. The PMCP-Mine algorithm is performed in a bottom-up manner. We first discover frequent transaction behaviors in a single store, e.g., {Starbucks, Latte}. Then, these single patterns can be joined to form compound patterns, e.g., {Hang Ten, clothes} $\xrightarrow{Giordano}$ {Starbucks, Latte}. Eventually, the complete mobile commerce patterns can be obtained by the PMCP-Mine algorithm. The PMCP-Mine algorithm is divided into three main phases: 1) Frequent-Transaction Mining. A Frequent-Transaction is a pair of store and items indicating frequently made purchasing transactions.

TABLE5
FREQUENT TRANSACTIONS.

User ID	Store	Item Set	Itemset Mapping	Large Transaction	Path	Sup.
U_1	A	$\{i_1\}$	LI_1	(U_1, A, LI_1)	A	4
U_1	D	$\{i_2\}$	LI_2	(U_1, D, LI_2)	D	3
U_1	F	$\{i_3\}$	LI_3	(U_1, F, LI_3)	F	2
U_1	F	$\{i_4\}$	LI_4	(U_1, F, LI_4)	F	2
U_1	K	$\{i_5\}$	LI_5	(U_1, K, LI_5)	K	2
U_2	A	$\{i_1\}$	LI_1	(U_2, A, LI_1)	A	4
U_2	K	$\{i_2\}$	LI_2	(U_2, K, LI_2)	K	4
U_3	B	$\{i_1\}$	LI_1	(U_3, B, LI_1)	B	3
U_3	E	$\{i_3\}$	LI_3	(U_3, E, LI_3)	E	2
U_4	D	$\{i_4\}$	LI_4	(U_4, D, LI_4)	D	3
U_1	F	$\{i_3, i_4\}$	LI_6	(U_1, F, LI_6)	F	2

In this phase, we first discover all Frequent-Transactions for each user. 2) Mobile Transaction Database Transformation. Based on the all Frequent-Transactions, the original mobile transaction database can be reduced by deleting infrequent items. The main purpose is to increase the database scan efficiency for pattern support counting. 3) PMCP Mining. This phase is mining all patterns of length k from patterns of length k-1 in a bottom-up fashion.

4.3.1 Frequent-Transaction Mining. In this phase, we mine the frequent transactions (F-Transactions) for each user by applying a modified Apriori algorithm [2]. Table 2 shows the mobile transaction database. At first, the support of each (store, item) pair is counted for each user. The patterns of *frequent 1-transactions* are obtained when their support satisfies the user-specified minimal support threshold T_{SUP} (T_{SUP} is set as 2 in this example). A candidate 2-transaction, indicating that two items are purchased together in the transaction, is generated by joining two frequent 1-transactions where their user identifications and stores are the same. For example, the candidate 2-transaction (F, $\{i_3, i_4\}$) is generated by joining (F, $\{i_3\}$) and (F, $\{i_4\}$), because the user identifications and purchased stores of them both are U_1 and F, respectively. Thus, we keep the patterns as frequent 2-transactions, when their support is larger than T_{SUP}. Finally, the same procedures are repeated until no more candidate transaction is generated. The frequent transactions are shown in Table 5. In the table, we use an item mapping table to re-label item sets in order to present F-Transactions in Table 5. For each unique item set, we use a symbol LI_i (Large Itemset i) to represent it, where i indicates a running number. The mapping procedure can reduce the time required to check if a mobile commerce pattern is contained in a mobile transaction sequence. Finally, the frequent 1-PMCPs (same as the F-Transaction) are obtained in Table 6.

4.3.2 Mobile Transaction Database Transformation. In this phase, we use F-Transactions to transform each mobile transaction sequence S into a *frequent mobile transaction sequence S'*. According to Table 5, if a transaction T in S is frequent, T would be kept as an F-Transaction. Otherwise, the store of T is taken as part of a path. Table 7 shows the result of frequent mobile transaction database transformed from Table 2. Take the partial sequence (U_1, A, LI_1) $\xrightarrow{BC}$ (U_1, D, LI_2) in the first mobile transaction

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

8   IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING

TABLE 6
FREQUENT-1 PMCPs.

Frequent-1 PMCP	Path	Support
(U_1, A, LI_1)	A	4
(U_1, D, LI_2)	D	3
(U_1, F, LI_3)	F	2
(U_1, F, LI_4)	F	2
(U_1, K, LI_5)	K	2
(U_2, A, LI_1)	A	4
(U_2, K, LI_2)	K	4
(U_3, B, LI_1)	B	3
(U_3, E, LI_3)	E	2
(U_4, D, LI_4)	D	3
(U_1, F, LI_6)	F	2

TABLE 7
FREQUENT MOBILE TRANSACTION DATABASE.

T_{id}	Frequent Mobile Transaction Sequence
1	$(U_1,A,LI_1)\xrightarrow{BC}(U_1,D,LI_2)\xrightarrow{E}(U_1,F,LI_6)\xrightarrow{I}(U_1,K,LI_5)$
2	$(U_1,A,LI_1)\xrightarrow{BC}(U_1,D,LI_2)$
3	$(U_1,A,LI_1)\xrightarrow{BC}(U_1,D,LI_2)\xrightarrow{E}(U_1,F,LI_6)\xrightarrow{I}(U_1,K,LI_5)$
4	$(U_1,A,LI_1)\xrightarrow{DC}$
5	$(U_2,A,LI_1)\xrightarrow{EF}(U_2,K,LI_2)\xrightarrow{I}$
6	$\xrightarrow{B}(U_2,A,LI_1)\xrightarrow{EF}(U_2,K,LI_2)$
7	$(U_2,A,LI_1)\xrightarrow{EF}(U_2,K,LI_2)\xrightarrow{I}$
8	$(U_2,A,LI_1)\xrightarrow{EF}(U_2,K,LI_2)\xrightarrow{I}$
9	$(U_3,B,LI_1)\xrightarrow{A}(U_3,E,LI_3)\xrightarrow{DE}$
10	$\xrightarrow{BAED}(U_3,B,LI_1)\xrightarrow{D}$
11	$(U_3,B,LI_1)\xrightarrow{A}(U_3,E,LI_3)\xrightarrow{D}$
12	$(U_4,D,LI_4)\xrightarrow{BA}$
13	$\xrightarrow{IFE}(U_4,D,LI_4)$
14	$\xrightarrow{IFE}(U_4,D,LI_4)$

sequence in Table 2 as an example, (A, i_1) and (D, i_2) are transformed into F-Transactions (U_1, A, LI_1) and (U_1, D, LI_2) in Table 7. On the other hand, $(B, \varnothing)$ and $(C, \varnothing)$ are transformed into the path BC. The main objectives and advantages of the transformation are: 1) item sets are represented as symbols for efficiently processing, and 2) transactions with insufficient support are eliminated to reduce the database size.

4.3.3 PMCP Mining. In this phase, we mine all the PMCPs from the frequent mobile transaction database. Frequent 1-PMCPs are obtained in the frequent-transaction mining phase, as described earlier. In the mining algorithm, we utilize a two-level tree, named *Personal Mobile Commerce Pattern Tree* (*PMCP-Tree*) to maintain the obtained PMCPs. Fig. 4 illustrates the PMCP-Tree. As shown, the upper level of the PMCP-Tree keeps track of the frequent mobile transactions. On the other hand, the lower level of the PMCP-Tree maintains the users and paths where PMCPs occurs. Thus, all the PMCPs are captured in the PMCP-Tree. Take the path <1> in Fig. 4(a) as an example, a PMCP $(A, LI_1) \xrightarrow{BCDEFI} (K, LI_5)$ of user
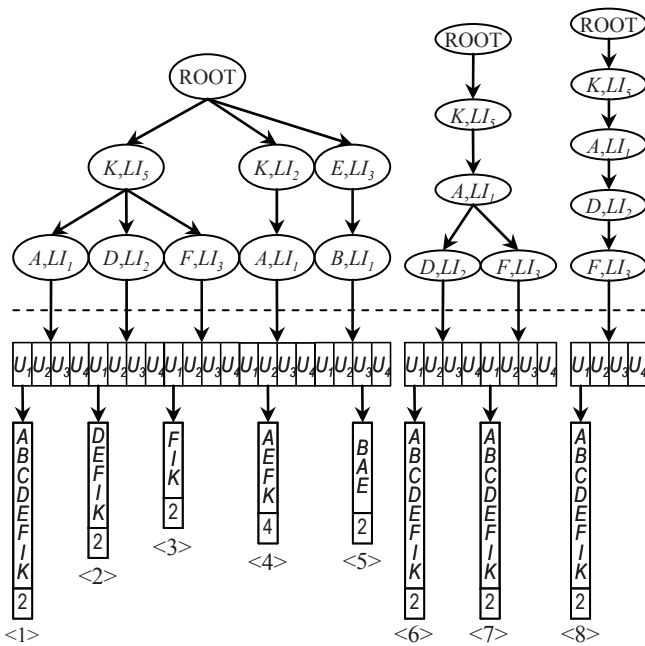
U_1 can be identified by the F-Transactions (A, LI_1) and (K, LI_5) along with the path $ABCDEFIK$. The procedure of the PMCP-Tree generation is as follows:

Step 1. PMCP-Mine generates candidate 2-PMCPs by hashing each combination of frequent transactions from the frequent mobile transaction sequence for each user, and then stores the results in the PMCP-Tree. Take Table 7 as an example. The two frequent transactions in MTS 9 and 11, i.e., (B, LI_1) and (E, LI_3), are the same and thus can be combined as candidate 2-PMCP, because their user identifications are also the same. The candidate 2-PMCP and its path are inserted to the path <5> of PMCP-Tree.

Step 2. To identify frequent 2-PMCPs, PMCP-Mine checks the candidate patterns whose support is larger than the minimal support threshold. Fig. 4(a) shows the part of frequent 2-PMCPs. Next, PMCP-Mine generates candidate 3-PMCPs with the path trimming technique [30] from frequent 2-PMCPs. A candidate 3-PMCP can be generated from two frequent 2-PMCPs, if one of moving paths in two frequent 2-PMCPs contains another. For example, the candidate pattern <6> in Fig. 4(b) is joined by <1> and <2> in Fig. 4(a), because the path $ABCDEFIK$ contains $DEFIK$ and their user identifications are the same.

Step 3. PMCP-Mine counts the support of candidate 3-PMCPs and identifies the frequent 3-PMCPs. There is an efficient method that generates candidate patterns in this step [30]. A candidate pattern CP_k: U_i: (X_1) (X_2) … (X_{k-1}) (X_k) and its path $m_1, m_2, …, m_q$, is joined by U_i: (X_1) (X_2) … (X_{k-3}) (X_{k-1}) (X_k) and U_i: (X_1) (X_2) … (X_{k-3}) (X_{k-2}) (X_k) and their paths are the same as CP_K. Explicitly, (X_2) (X_3) … (X_{k-2}) (X_{k-1}) is joined by centro-subtransactionsets (X_2) (X_3) … (X_{k-3}) (X_{k-2}) and (X_2) (X_3) … (X_{k-3}) (X_{k-1}) [30]. In Fig. 4(c), the pattern <8> is generated by joining the centro-subtransactionset (D, LI_2) and (F, LI_3), because the first and the last transaction of <6> and <7> are the same, and their moving paths and user identifications are the same. The goal of PMCP-Tree is to efficiently generate candidate mobile commerce patterns because PMCP-Tree can quickly compare two patterns to check whether they have the same first and last transactions.

Step 4. Repeat Step 3 until no more candidate patterns can be generated.

(a) The part of frequent 2-PMCPs.   (b) 3-PMCPs. (c) 4-PMCPs.

Fig. 4. PMCP-Tree.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

ERIC HSUEH-CHAN LU, WANG-CHIEN LEE, AND VINCENT S. TSENG: A FRAMEWORK FOR PERSONAL MOBILE COMMERCE PATTERN MINING AND PREDICTION

9

4.4 Mobile Commerce Behavior Predictor (MCBP)

In this section, we describe how to use the discovered PMCPs to predict the users' future mobile commerce behaviors which include movements and transactions. In existing pattern-based prediction models, the pattern selection strategy is based on exact matching, i.e., the similarity between different locations is treated as 0. Such prediction strategy may lead to prediction failures if there is no existing pattern to match. To overcome this problem, we integrate the similarities of stores and items which are obtained from SIM into the mobile commerce behavior prediction. We first define the *premise* and *consequence* of a PMCP. Let $P = (s_1, I_1) \xrightarrow{p_1} (s_2, I_2) \xrightarrow{p_2} ... (s_m, I_m) \xrightarrow{p_m} (s_{m+1}, I_{m+1})$ be a PMCP of length $m+1$, and the support of P be $sup(P)$. We call $(s_1, I_1) \xrightarrow{p_1} (s_2, I_2) \xrightarrow{p_2} ... (s_m, I_m) \xrightarrow{p_m}$ the *premise* and (s_{m+1}, I_{m+1}) the *consequence*. Let $P' = (s_1', I_1') \xrightarrow{p_1'} (s_2', I_2') \xrightarrow{p_2'} ... (s_n', I_n') \xrightarrow{p_n'}$ be the user's recent mobile commerce behavior (with length equal to n). The most basic pattern-based prediction strategy is to choose the pattern with highest support from all the patterns whose premise matches the user's recent mobile commerce behavior. Such a prediction strategy is named *Support Only (SO)* in this paper. For example, Let $P_1 = \{(A, I_1) \xrightarrow{B} (C, I_2) \xrightarrow{DE} (F, I_3)\}$, $P_2 = \{(C, I_2) \xrightarrow{DE} (I, I_4)\}$, and $P_3 = \{(D, I_1) \xrightarrow{E} (F, I_2) \xrightarrow{AB} (C, I_5)\}$ be three PMCPs and $sup(P_1) = 5$, $sup(P_2) = 8$, and $sup(P_3) = 10$. $P' = \{(A, I_1) \xrightarrow{B} (C, I_2) \xrightarrow{DE}\}$ be the user's recent mobile commerce behavior. Since the premise of P_3, i.e., $\{(D, I_1) \xrightarrow{E} (F, I_2) \xrightarrow{AB}\}$, does not match P', P_3 is not used for predicting the user next behaviors. Since P_1 and P_2 both match P', they are thus two candidates for behavior prediction. Since P_2 has a larger support than P_1, the consequence of P_2, i.e., (I, I_4), is used to predict the user's next behaviors. However, the *SO* strategy does not take into account the lengths of pattern matches. For instance, in the above example, although the support of P_2 is larger than that of P_1, the matching length of P_1 is longer than P_2 for user's recent mobile behaviors. In the mobile commerce behavior prediction, a longer pattern match may represent that this pattern is better matched for recent mobile commerce behaviors.

Based on the above discussion, we propose the second prediction strategy named *Integration of Support and Matching length (ISM)*. The idea of *ISM* is to incorporate both the pattern support and matching length into the mobile commerce behavior prediction. In *ISM*, we design a scoring function $sf(P, P')$ to compute the matching score between the *premise* of a PMCP and user's recent mobile commerce behavior. The *consequence* of PMCP with the highest score is used to predict the next mobile commerce behavior. The scoring function $sf(P, P')$ is defined as (6), where the function $matching\_length(P, P')$ represents the length of pattern matching.

$$sf(P,P') = matching\_length(P,P') \times sup(P) \quad (6)$$

For example, Let $P_1 = \{(A, I_1) \xrightarrow{B} (C, I_2) \xrightarrow{DE} (F, I_3)\}$ and $P_2 = \{(C, I_2) \xrightarrow{DE} (I, I_4)\}$ be two PMCPs with $sup(P_1)$

$= 5$ and $sup(P_2) = 8$. Let $P' = \{(A, I_1) \xrightarrow{B} (C, I_2) \xrightarrow{DE}\}$ be the user's recent mobile commerce behavior. The matching lengths are 2 between P_1 and P' and 1 between P_2 and P'. Hence, the consequence of P_1, i.e., (F, I_3), is used to predict the user's next behavior, since the pattern score of P_1 is $2 \times 5 = 10$ which is larger than that of P_2 $(1 \times 8 = 8)$. However, the predictions fail if there is no pattern to match in pattern based predictions. To overcome this problem, we incorporate the store and item similarities into the mobile commerce behavior prediction.

Accordingly, we propose *Mobile Commerce Behavior Predictor (MCBP)*, which measures the similarity score of every PMCP with a user's recent mobile commerce behavior by taking store and item similarities into account. In *MCBP*, three ideas are considered: 1) the premises of PMCPs with high similarity to the user's recent mobile commerce behavior are considered as prediction knowledge; 2) more recent mobile commerce behaviors potentially have a greater effect on next mobile commerce behavior predictions; and 3) PMCPs with higher support provide greater confidence for predicting users' next mobile commerce behavior. Based on the above ideas, we propose a weighted scoring function to evaluate the scores of PMCPs. The weighted scoring function $wsf(P, P')$ is defined as (7).

$$wsf(P,P') = \left(\sum_{j=0}^{max(m,n)-1} ps(P_{m-j}, P_{n-j}') \times w_{max(m,n)-j} \right) \times sup(P) \quad (7)$$

where the function $ps(P_u, P_v')$, defined as (8), represents the pattern similarity for $P_u = \{(s_u, I_u) \xrightarrow{p_u}\}$, $P_v = \{(s_v', I_v') \xrightarrow{p_v'}\}$ and w_k, defined as (11), denotes the weight of a pattern element at position k.

$$ps(P_u, P_v') = \begin{cases} 0, \text{if } u < 1 \text{ or } v < 1 \\ sim(s_u, s_v') + sim_I(I_u, I_v') + sim_P(p_u, p_v'), \text{otherwise} \end{cases} \quad (8)$$

where $sim(s_u, s_v')$ is the store similarity of s_u and s_v', $sim_I(I_u, I_v')$ is defined as (9) which is the item set similarity of I_u and I_v', and $sim_P(p_u, p_v')$ is the path similarity of p_u and p_v'.

$$sim_I(I_u, I_v') = \frac{1}{|I_u| \times |I_v'|} \times \sum_{\forall i_u \in I_u} \sum_{\forall i_v' \in I_v'} sim(i_u, i_v') \quad (9)$$

where $sim(i_u, i_v')$ is the item similarity of i_u and i_v'.

The function $ps(P_u, P_v')$ is used to measure the similarity between two pattern elements. If the index u or v is less than 1, the function returns 0 because either P_u or P_v' is an empty element. Otherwise, we measure the store similarity $sim(s_u, s_v')$, the item set similarity $sim_I(I_u, I_v')$, and the path similarity $sim_P(p_u, p_v')$ for p_u and p_v'. The values of store similarity and item similarity can be obtained by SIM as described in Section 4.2. For the similarity of paths, we apply the sequence alignment strategy [4] to measure the similarity of two paths. Dynamic programming is effective in solving sequence alignment problems. Let $p_u = \{s_1, s_2, ..., s_x\}$ and $p_v' = \{s_1', s_2', ..., s_y'\}$ be two paths, the alignment score of p_u and p_v' can be calculated by $M_{x, y}$ which is defined as (10).

$$M_{i,j} = \begin{cases} 0, \text{if } i = 0 \text{ or } j = 0 \\ Max(M_{i-1,j-1} + sim(s_i, s_j'), M_{i-1,j}, M_{i,j-1}), \text{otherwise} \end{cases} \quad (10)$$

where $sim(s_i, s_j')$ is also obtained from SIM in Section 4.2. Since the value range of $sim(s_i, s_j')$ is within 0 to 1, $M_{x, y}$ is

within 0 to $max(x, y)$. To normalize the value, $simP(p_u, p_v')$ is defined as $M_{x,y} / max(x, y)$.

$$w_k = \frac{k}{\sum_{i=1}^{max(m,n)} i} \tag{11}$$

Equation (11) gives more weight to recent movement. For all PMCPs, we can calculate their pattern score by the weighted scoring function. The *consequence* of PMCP with the highest score is used to predict the next mobile commerce behavior.

5 EXPERIMENTAL EVALUATION

We conduct a series of experiments to evaluate the performance of the proposed framework MCE and its three components, i.e., SIM, PMCP-Mine, and MCBP under various system conditions. The synthetic data generator is described in Section 5.1. In the experiments, we evaluate the precision of mobile behavior prediction under a number of examined prediction methods. All of the experiments were implemented in Java on a 3.0 GHz machine with 1 GB of memory running windows XP.

5.1 Simulation Model

The simulation model has been adopted by Yun *et al.* [30] as well. Table 8 summarizes the major parameters in the simulation model and their default values. In the base experiment model, we use a $|W| \times |W|$ mesh network [16] to model the stores. The atomic temporal observation unit of life is "day". In each store, the number of items is randomly generated based on a uniform distribution within a given range N_I. The advancing probability P_a of each neighbor for each store is the probability to move to a given neighboring store from the store and purchase some items sold there. In other words, each directed edge between two neighbor stores is assigned with an advancing probability. In the model, the advancing probability is defined as the ratio of the number of items in each neighbor to those numbers of other neighbors. The backward moving represents that a user will move from the current store back to the store from which he came. The backward probability is denoted by $P_0 = P_a \times W_b$, where W_b is a backward weight. For example, Fig. 5(a) shows a 3×3 mesh network. There are four neighbor stores N_1, N_2, N_3, and N_4 for store Y and the corresponding advancing probabilities are P_{a1}, P_{a2}, P_{a3}, and P_{a4}, respectively. In addition, N_{I1}, N_{I2}, N_{I3}, and N_{I4} indicate the numbers of items in stores N_1, N_2, N_3, and N_4 and $P_{ai} = N_{si} / (N_{I1} + N_{I2} + N_{I3} + N_{I4})$, $\forall 1 \leq i \leq 4$. When a user visits store Y from store N_1, the probabilities for moving to the neighbors are shown in Fig. 5 (b).

In this experiment, there are N_U users in this network. When a user moves between stores, the mobile transaction sequence is generated by this user. The method for generating mobile transaction sequences is referred to [17]. For each user, the number of mobile transaction sequences which is generated by this user is N. In each mobile transaction sequence, the length is determined from a Poisson distribution with mean equal to L_T. When a user moves to a store, the probability that this user purchases the items in this store is denoted by P_b. In [30], the paper

TABLE 8
MAJOR PARAMETERS OF SIMULATION MODEL.

Parameter	Description	Default Value
$\|W\|$	$\|W\| \times \|W\|$ mesh network	6
N_U	The number of users	30
N	Average size of transaction log (a user)	10,000
L_T	Average length of a transaction	20
N_I	The number of items	500
P_b	The probability of item transaction	0.7
P_E	Event probability	0.7
W_b	The weight of backward movement	0.5
T_{SUP}	The minimal support threshold	0.1%

mentioned that people tend to buy sets of items together, which are called potential maximal frequent sets. A transaction may contain one or more of such frequent sets. Therefore, in each user, some potential events are generated to represent the mobile behaviors of this user. The potential events indicate the preferred item sequences. Each user may purchases such item sequences adhering to potential events with probability P_E or randomly [24].

The followings are the main measurements for the experimental evaluation. The Precision, Recall, and F-measure are defined as (12), (13), and (14), where p^+ and p^- indicate the number of correct predictions and incorrect predictions, respectively, and $|R|$ indicates the total number of item transactions. In addition, we use the average improvement rate to measure how many percentage of our proposed method is better than other methods. The average improvement rate is defined as (15), where m_{ours} and $m_{baseline}$ are the measured result of our proposed method and that of the compared method, respectively.

$$Precision = \frac{p^+}{p^+ + p^-} \tag{12}$$

$$Recall = \frac{p^+ + p^-}{|R|} \tag{13}$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{14}$$

$$Average\ Improvement\ Rate = \frac{m_{ours} - m_{baseline}}{m_{baseline}} \tag{15}$$

The experiments are divided into two parts: i) internal evaluation; and ii) external comparison. The internal evaluation focuses on evaluation of proposed techniques within the MCE framework. We first compare the pro-
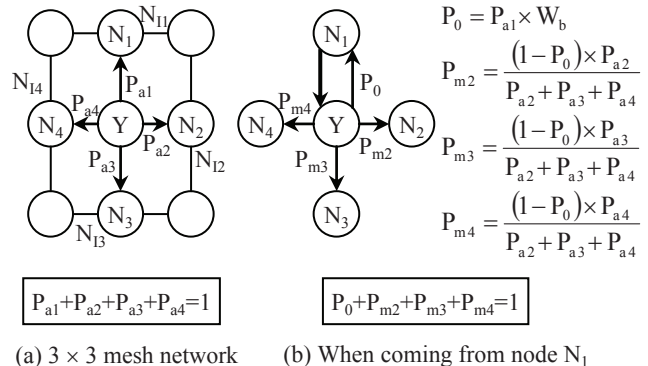

$$P_0 = P_{a1} \times W_b$$

$$P_{m2} = \frac{(1 - P_0) \times P_{a2}}{P_{a2} + P_{a3} + P_{a4}}$$

$$P_{m3} = \frac{(1 - P_0) \times P_{a3}}{P_{a2} + P_{a3} + P_{a4}}$$

$$P_{m4} = \frac{(1 - P_0) \times P_{a4}}{P_{a2} + P_{a3} + P_{a4}}$$

$$P_{a1} + P_{a2} + P_{a3} + P_{a4} = 1 \qquad\qquad P_0 + P_{m2} + P_{m3} + P_{m4} = 1$$

(a) 3×3 mesh network   (b) When coming from node N_1

Fig. 5. Simulate a mobile commerce environment using mesh network.

posed *Similarity Inference Model* (*SIM*) with *Set Similarity* (*SET*) described in (1) in terms of precision, recall, and F-measure. Next, we compare the three proposed prediction techniques described in Section 4.4, i.e., *Support Only* (*SO*), *Integration of Support and Matching length* (*ISM*), and *Mobile Commerce Behavior Predictor* (*MCBP*) in terms of precision, recall, and F-measure. Finally, we evaluate the efficiency of the proposed algorithms including SIM, PMCP-Mine, and MCBP. For external comparison, we evaluate the proposed framework *Mobile Commerce Explorer* (*MCE*) as a whole against three prediction frameworks: 1) *Markov Model* (*MM*) [10], 2) *Hybrid Prediction Model* (*HPM*) [12], and 3) *Mobile Sequential Pattern* (*MSP*) [30] under various parameters, including the minimal support threshold, the event probability, and the network size, in terms of precision, recall, and F-measure. In this series of experiments, the main goal is to measure the precisions of mobile commerce behavior predictions by the examined methods. For the synthetic data generated from the data generator, 70% of the data are used for training, and the remaining 30% are used for prediction.

5.2. Comparison of various similarity measures

The goal of this experiment is to compare SIM with SET as the similarity measure component of MCE in terms of precision, recall, and F-measure. Fig. 6(a) shows that SIM outperforms SET in all metrics. The main reason is that the similarity inference among stores and items for SIM is more accurate than that for SET. The similarity between two stores can be accurately measured by SIM, even if there is no common item sold in these two stores. As the figures show, SIM outperforms SET by 41.03% in terms of precision, 7.19% in terms of recall, and 31.39% in terms of F-measure.

5.3. Comparison of various prediction techniques

This experiment analyzes the precision, recall and F-measure of examined prediction techniques, including Support Only (SO), Integration of Support and Matching length (ISM), and Mobile Commerce Behavior Prediction (MCBP). Fig. 6(b) shows that MCBP is slightly better than SO and ISM in terms of precision, but significantly outperforms them in terms of recall and F-measure. This is because that MCBP considers not only the pattern supports but also the weighted matching score between pattern premises and users' recent mobile commerce behaviors in the behavior prediction. Since the evaluation of weighted matching score is based on store and item simi-
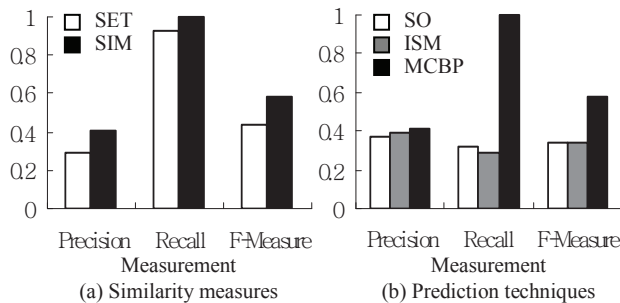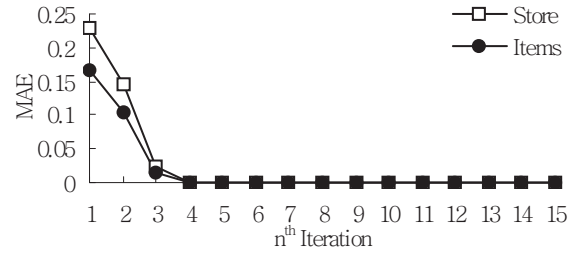

Fig. 7. Convergence of similarity inference model.

larities, the recall of MCBP can achieve near 100%. The average improvement rates of MCBP over SO and ISM are 10.18% and 3.57% for the precision, 215.36% and 240.34% for the recall, and 69.76% and 72.4% for the F-measure, respectively.

5.4. Convergence of similarity inference model

This experiment aims to demonstrate that the proposed SIM converges in various settings. In this experiment, 1,000 different mobile transaction databases under various settings (the network size is set from 6x6 to 14x14 and the number of items is set from 200 to 500) are generated. For each dataset, we execute SIM to observe whether the iterative process converges or not. For each store or item similarity inference, we use Mean Absolute Error (MAE) to measure how close current similarities are to the last similarities. Fig. 7 shows that the average MAE of 1,000 datasets in every inference. We observe that the SIM inference converges quickly under all the tested datasets, i.e., the average MAE is near 0 after only the 5th iteration. In this work, although we do not formally prove that SIM always converge, we empirically show that all the inferences under our test-case datasets do converge.

5.5. Performance of the proposed algorithms

This experiment evaluates the performance of the proposed algorithms in the MCE framework, including SIM, PMCP-Mine, and MCBP in terms of execution time. Fig. 8(a) shows that the execution time for SIM remains constant while the execution time for both PMCP-Mine and MCBP increases as the support threshold decreases. Since the number of stores and items is constant after the mobile transaction database is transformed into SID and ISD, the execution time of SIM is constant. As the support threshold decreases, the number of discovered PMCPs increases. Hence, the PMCP-Mine needs more time to mine the PMCPs. In the phase of behavior prediction, the
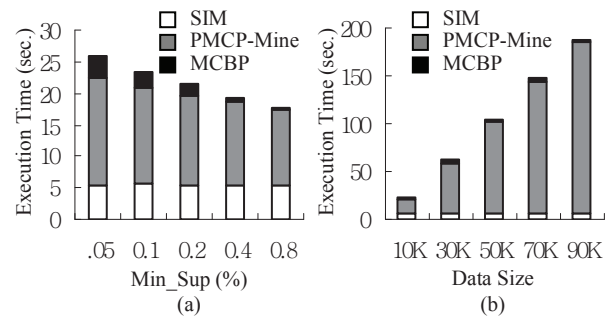

Fig. 6. Comparison of various techniques.

Fig. 8. Performance evaluation of the proposed algorithms.

MCBP needs more time to match users' recent mobile commerce behaviors and PMCPs under a lower support threshold, too. In Fig. 8(b), with the data size increases, only the execution time of PMCP-Mine increases. Although the PMCP-Mine needs more time to discover the PMCPs when the data size increases, it is acceptable since both of SIM and PMCP-Mine are offline processes. For the behavior prediction, we observe that MCBP is efficient (within 3.5 seconds) even when a lower support threshold is used.

5.6. Impact of the minimal support threshold T_{SUP}

Here we start our external comparison of the proposed MCE with MM, HPM and MSP. This experiment analyzes their precision, recall and F-measure by varying the minimal support threshold from 0.05% to 0.8%. Fig. 9(a), 9(b) and 9(c) show that MCE outperforms MM, HPM, and MSP in terms of precision, recall and F-measure, respectively. We observe that the recall keeps 100% for MCE, while that for MSP decreases as the support threshold increases. Since MCBP is the prediction technique used in MCE, the recall of MCE achieves 100% under various support thresholds (as discussed in Section 5.3). For MSP, the number of discovered patterns is sensitive to the support threshold. In the behavior prediction phase, the number of completely matched patterns decreases, when the support threshold is larger than 0.2%. Hence, the recall of MSP decreases significantly with the support threshold increases. Besides, we observe that the precision slightly decreases for MCE, while that for MSP increases as the support threshold increases. The reason is that MCE ranks all mobile patterns based on their weighted matching scores and selects the pattern with the highest score to predict the next behavior, even when all the pattern scores are very low. It causes the precision of MCE to decrease slightly when the support threshold increases, even though fewer noisy mobile patterns are obtained under the larger support threshold setting. For MSP, since the prediction technique is based on exact pattern matching, few noises result in high precision and low recall. Therefore, setting a suitable support threshold is important, which can be verified by experiments. In all experiments, the support threshold is set as 0.1%. The average improvement rate of MCE over the three generalized frameworks is 109.06% for the precision, 26.79% for the recall, and 143.7% for the F-measure.
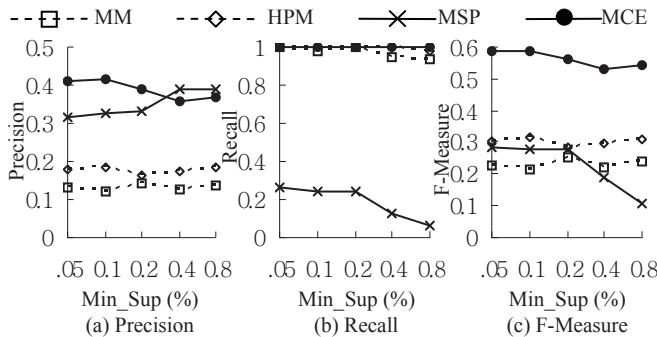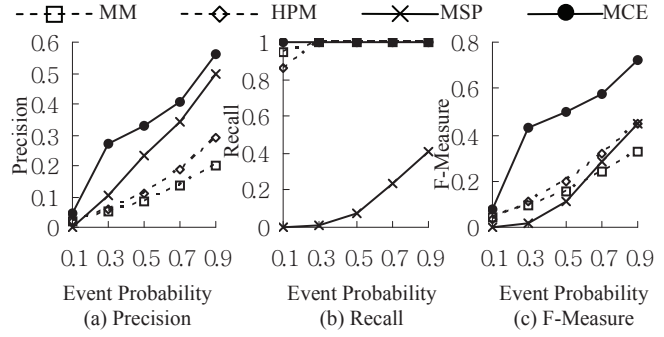

Fig. 10. Comparison of various event probabilities (P_E).

5.7 Impact of the event probability P_E

This experiment analyzes the precision, recall and F-measure of examined prediction frameworks by varying the event probability from 0.1 to 0.9. The event probability is the probability of a user adheres to potential events to move and to purchase potential items. Fig. 10(a), 10(b) and 10(c) show that MCE outperforms MM, HPM, and MSP in terms of precision, recall and F-measure with varied event probability. We observe that the precision, recall and F-measure increase significantly with the increase in event probability. The reason is that the data noise rate for mobile user movement and transaction decreases, when the event probability of mobile network environment increases. Hence, the collected data from a mobile network with a higher event probability reveal more precise mobile patterns. In the prediction phase, these patterns are used to predict the next mobile behavior of each user, and thus achieve a better performance. The average improvement rate of MCE over the three generalized frameworks is 144.49% for the precision, 28.71% for the recall, and 170.95% for the F-measure.

5.8. Impact of the network size |W|

This experiment analyzes the precision, recall and F-measure of examined prediction frameworks when the network size is varied from 6x6 to 14x14. Fig. 11(a), 11(b) and 11(c) show that MCE outperforms MM, HPM, and MSP in terms of precision, recall and F-measure under various network sizes. We observe that the precision, recall and F-measure decrease by increasing the network size. We observe that the precision is nearly constant as the network size increases. Both the recall and F-measure decrease gradually as the network size increases. This is because that the movements of mobile users are more
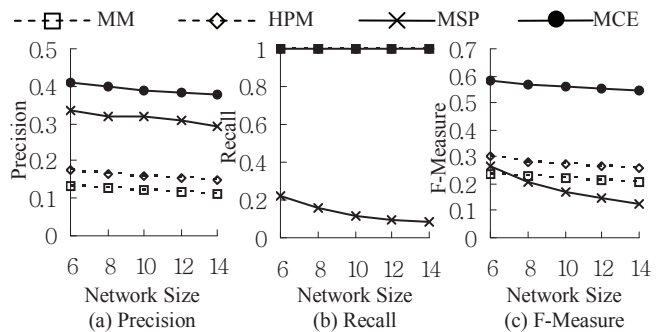

Fig. 9. Comparison of various minimal supports (T_{SUP}).

Fig. 11. Comparison of various network sizes (|W|).

ERIC HSUEH-CHAN LU, WANG-CHIEN LEE, AND VINCENT S. TSENG: A FRAMEWORK FOR PERSONAL MOBILE COMMERCE PATTERN MINING AND PREDICTION

13

complicated in a large mobile network. Hence, the mobile users create varied mobile behaviors in terms of movement between stores, paths, and purchased items in transactions. The average improvement rate of MCE over the other three frameworks is 125.41% for the precision, 23.42% for the recall, and 140.9% for the F-measure.

5.9. Summary of experimental results

The above experiments can be divided into two parts: 1) internal evaluations for store/item similarity measurement and mobile commerce behavior prediction in MCE; and 2) external evaluations for MCE and three other prediction frameworks. For the first part, the experimental results show the following two conclusions: 1) using SIM to inference the store and item similarities is more precise than using SET. The obtained similarity knowledge can improve the recall of behavior prediction when a user moves to stores or buys items previously unknown to the user. 2) Using MCBP prediction technique can achieve higher precision than using SO or ISM, because MCBP considers not only the pattern supports but also the weighted matching score between pattern premises and users' recent mobile commerce behaviors in behavior prediction. Hence, MCBP can capture a user's recent mobile commerce behavior and select the best PMCP to prediction the user's next behavior precisely. 3) The performance of the MCE framework is efficient.

For the second part, the experiments consist of three measurements: 1) the studies on precision, 2) the studies on recall, and 3) the studies on F-measure. For the studies on precision, it is observed that MCE outperforms the other three frameworks under various system conditions. For the studies on recall, MM, HPM, and MCE, are close to 100%. MM is a Markov model method, the transition probabilities decide the prediction result. A location cannot be predicted if this location has never been visited by any user. With sufficient training data, the recall can be close to 100%. HPM is a hybrid method. When the pattern based prediction fails, the regression based prediction would be started to predict the next behavior of the moving object. The prediction technique of MCE, i.e., MCBP, is to choose the pattern with the highest weighted matching score for predicting the next behavior of mobile users. Hence, the recall of MCE can achieve near 100%. Although the recalls of MM, and HPM are close to 100%, their precisions are significantly lower than MCE. In mobile commerce applications, low precision in predictions may lead to high penalty in business cost. Hence, the precision is more important than the recall for behavior prediction. For the studies on F-measure, it is observed that MCE performs the best under various system conditions.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel framework, namely *Mobile Commerce Explorer* (*MCE*), for mining and prediction of mobile users' movements and transactions in mobile commerce environments. In the *MCE* framework, we have proposed three major techniques: 1) *Similarity Inference Model* (*SIM*) for measuring the similarities among stores and items; 2) *Personal Mobile Commerce Pattern Mine* (*PMCP-Mine*) algorithm for efficiently discovering mobile users' *Personal Mobile Commerce Patterns* (*PMCPs*); and 3) *Mobile Commerce Behavior Predictor* (*MCBP*) for predicting possible mobile user behaviors. To our best knowledge, this is the first work that facilitates mining and prediction of personal mobile commerce behaviors that may recommend stores and items previously unknown to a user.

To evaluate the performance of the proposed framework and three proposed techniques, we conducted a series of experiments. The experimental results show that the framework MCE achieves a very high precision in mobile commerce behavior predictions. Besides, the prediction technique MCBP in our MCE framework integrates the mined PMCPs and the similarity information from SIM to achieve superior performs in terms of precision, recall, and F-measure. The experimental results show that our proposed framework and three components are highly accurate under various conditions.

For the future work, we plan to explore more efficient mobile commerce pattern mining algorithm, design more efficient similarity inference models, and develop profound prediction strategies to further enhance the MCE framework. In addition, we plan to apply the MCE framework to other applications, such as object tracking sensor networks and location based services, aiming to achieve high precision in predicting object behaviors.

REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rule between Sets of Items in Large Databases," *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 207-216, May 1993.

[2] R. Agrawal and R. Srikant, "Fast Algorithm for Mining Association Rules," *Proc. Int'l. Conf. on Very Large Databases*, pp. 478-499, Sept. 1994.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," *Proc. Int'l. Conf. on Data Engineering*, pp. 3-14, Mar. 1995.

[4] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic Local Alignment Search Tool," *J. of Molecular Biology*, vol. 215, no. 3, pp. 403-410, Oct. 1990.

[5] M.-S. Chen, J.-S. Park, and P. S. Yu, "Efficient Data Mining for Path Traversal Patterns," *IEEE Trans. on Knowledge and Data Engineering*, vol. 10, no. 2, pp. 209-221, Apr. 1998.

[6] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules in Large Database," *Proc. Int'l Conf. Very Large Data Bases*, pp. 420-431, Sept. 1995.

[7] J. Han and M. Kamber, "Data Mining: Concepts and Techniques, 2nd Edition," *Morgan Kaufmann*, Sept. 2000.

[8] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 1-12, May 2000.

[9] J. L. Herlocker, J. A. Konstan, A. Brochers, and J. Riedl, "An Algorithm Framework for Performing Collaborative Filtering,"

Proc. Int'l. ACM SIGIR Conf. on Research and Development in Information Retrieval, pp. 230-237, Aug. 1999.

[10] Y. Ishikawa, Y. Tsukamoto, and H. Kitagawa, "Extracting Mobility Statistics from Indexed Spatio-Temporal Datasets," *Proc. Workshop on Spatio-Temporal Database Management*, pp. 9-16, Aug. 2004.

[11] G. Jeh and J. Widom, "SimRank: A Measure of Structural-Context Similarity," *Proc. Int'l. Conf. on Knowledge Discovery and Data Mining*, pp. 538-543, Jul. 2002.

[12] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou, "A Hybrid Prediction Model for Moving Objects," *Proc. Int'l. Conf. on Data Engineering*, pp.70-79, Apr. 2008.

[13] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," *New York: Wiley*, Mar. 1990.

[14] S. C. Lee, J. Paik, J. Ok, I. Song, and U. M. Kim, "Efficient Mining of User Behaviors by Temporal Mobile Access Patterns," *Int'l. J. of Computer Science Security*, vol. 7, no. 2, pp. 285-291, Feb. 2007.

[15] Y. Lu. "Concept Hierarchy in Data Mining: Specification, Generation and Implementation," *Master 's thesis, Simon Fraser University*, 1997.

[16] E. Modiano and A. Ephremides, "Efficient Algorithms for Performing Packet Broadcasts in a Mesh Network," *IEEE/ACM Trans. on Networking*, vol. 4, no. 4, pp. 639-648, Aug. 1996.

[17] J.-S. Park, M.-S. Chan, and P. S. Yu, "An Effective Hash Based Algorithm for Mining Association Rules," *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 175-186, May 1995.

[18] J. M. Patel, Y. Chen, and V. P. Chakka, "Stripes: An Efficient Index for Predicted Trajectories," *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 635-646, Jun. 2004.

[19] J. Pei, J. Han, B. Mortazavi-Asl, and H. Zhu, "Mining Access Patterns Efficiently from Web Logs," *Proc. Pacific Asia Conf. on Knowledge Discovery and Data Mining*, pp. 396-407, Apr. 2000.

[20] ShopKick, http://www.shopkick.com/index.html, 2010.

[21] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu, "Prediction and Indexing of Moving Objects with Unknown motion patterns," *Proc. ACM SIGMOD Conf. on Management of Data*, pp. 611-622, Jun. 2004.

[22] Y. Tao, D. Papadias, and J. Sun, "The tpr*-tree: An Optimized Spatio-Temporal Access Method for Predictive Queries," *Proc. Int'l. Conf. on Very Large Data Bases*, pp. 790-801, Sept. 2003.

[23] V. S. Tseng and W. C. Lin, "Mining Sequential Mobile Access Patterns Efficiently in Mobile Web Systems," *Proc. Int'l. Conf. on Advanced Information Networking and Applications*, pp. 867-871, Mar. 2005.

[24] V. S. Tseng and K. W. Lin, "Efficient Mining and Prediction of User Behavior Patterns in Mobile Web Systems," *Information and Software Technology*, vol. 48, no. 6, pp. 357-369, Jun. 2006.

[25] V. S. Tseng, H. C. Lu, and C. H. Huang, "Mining Temporal Mobile Sequential Patterns in Location-Based Service Environments," *Proc. Int'l. Conf. on Parallel and Distributed Systems*, pp. 1-8, Dec. 2007.

[26] V. S. Tseng and C. F. Tsui, "Mining Multi-Level and Location-Aware Associated Service Patterns in Mobile Environments," *IEEE Trans. on Systems, Man and Cybernetics: Part B*, vol. 34, no. 6, Dec. 2004.

[27] U. Varshney, R. J. Vetter, and R. Kalakota, "Mobile Commerce: A New Frontier," *IEEE Computer*, vol. 33, no. 10, pp. 32-38, Oct. 2000.

[28] J. Veijalainen, "Transaction in Mobile Electronic Commerce," *Proc. Int'l. Workshop on Foundations of Models and Languages for Data and Objects*, pp. 203-227, Sept. 1999.

[29] D. Xin, J. Han, X. Yan, and H. Cheng. "Mining Compressed Frequent-Pattern Sets," *Proc. Int'l. Conf. on Very Large Data Bases*, pp. 709-720, Aug. 2005.

[30] C. H. Yun and M. S. Chen, "Mining Mobile Sequential Patterns in a Mobile Commerce Environment," *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, vol. 37, no. 2, pp. 278-295, Mar. 2007.

[31] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining Individual Life Pattern Based on Location History," *Proc. Int'l. Conf. on Mobile Data Management Systems, Services and Middleware*, pp. 1-10, May 2009.

[32] X. Yin, J. Han, P. S. Yu. "LinkClus: Efficient Clustering via Heterogeneous Semantic Links," *Proc. Int'l. Conf. on Very Large Data Bases*, pp. 427-438, Aug. 2006.

[33] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining Interesting Location and Travel Sequences from GPS Trajectories," *Proc. Int'l. World Wide Web Conf.*, pp. 791-800, Apr. 2009.

Eric Hsueh-Chan Lu received the B.S. degree in computer science and information engineering from National Taiwan University of Science and Technology, Taiwan, R.O.C., in 2003, and the PhD degree from the Department of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C., in 2010. His research interests include data mining, location-based services, mobile computing, social networking, object tracking in sensor networks, and intelligent transport systems.

Wang-Chien Lee is an Associate Professor of Computer Science and Engineering at Pennsylvania State University. He received the BS degree from the National Chiao Tung University, Hsinchu, Taiwan, R.O.C., the MS degree from the Indiana University, Bloomington, and the PhD degree from the Ohio State University, Columbus. He is an associate professor of computer science and engineering at Pennsylvania State University, University Park, where he also leads the Pervasive Data Access Research Group to perform cross-area research in database systems, pervasive/mobile computing, and networking. Most of his research results have been published in prestigious journals and conference proceedings in the fields of databases, mobile computing, and networking. He has served as a guest editor for several journal special issues on mobile database-related topics, including the IEEE Transactions on Computer, IEEE Personal Communications Magazine, ACM MONET, and ACM WINET. He was the founding program committee co-chair for the International Conference on Mobile Data Management. He is a member of the IEEE and the ACM.

Vincent S. Tseng is currently a professor at Department of Computer Science and Information Engineering at National Cheng Kung University (NCKU), Taiwan, R.O.C.. Before this, he was a postdoctoral research fellow in Computer Science Division of University of California at Berkeley during January 1998 and July 1999. He has acted as the director for Institute of Medical Informatics of NCKU since August 2008. During February 2004 and July 2007, he had also served as the director for Informatics Center in NCKU Hospital. Dr. Tseng received his Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C., in 1997. Dr. Tseng has a wide variety of research interests covering data mining, biomedical informatics, mobile computing and Web technologies. He is on the editorial board of International Journal of Data Mining and Bioinformatics and has also served as chairs/program committee for a number of premier international conferences related to data mining and databases.