International Journal of Innovation and Scientific Research
ISSN 2351-8014 Vol. 5 No. 1 Jul. 2014, pp. 30-39
© 2014 Innovative Space of Scientific Research Journals
http://www.ijisr.issr-journals.org/

Efficiently Mining the Frequent Patterns in Mobile Commerce Environment

S. Jacinth Evangeline[1], K.M. Subramanian[1], and Dr. K. Venkatachalam[2]

[1]Computer Science and Engineering, Erode Sengunthar Engineering College,
Anna University Chennai, Tamilnadu, India

[2]Electrical Communication and Engineering, Vellalar College of Engineering and Technology,
Anna University Chennai, Tamilnadu, India

ABSTRACT: Nowadays, a rapid development in the communication technology and increasing the usability of powerful portable devices, mobile users can use their mobile devices to access the information. One of the active areas is the mining and prediction of users' mobile commerce behaviors such as their movements and purchase transactions. The important issue is to mine the rare frequent items from database to satisfy the user needs. In this paper, we propose a technique that can efficiently satisfy the user needs. It predicts the frequent item based on the user selection. Systolic tree implementation is used to predict the frequently moved item in the database. The main aim is to recommend the stores and items previously to unknown user. We evaluate our system in real world and deliver good performance in terms efficiency and scalability.

KEYWORDS: Data Mining, Frequent Pattern, Mobile Commerce, Prediction.

1 INTRODUCTION

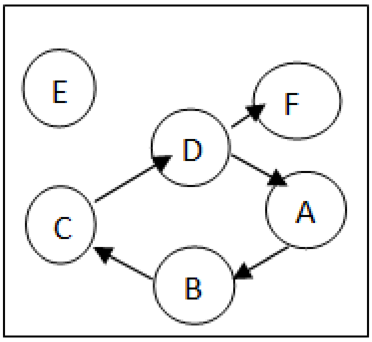
The advance of powerful portable devices with wireless communication technologies, has made the mobile services available at anywhere at any time. In future, it is expected that hundreds of millions of users will carry their mobile phones that use wireless connection to access the information making the mobile commerce [9] a reality. Mobile Commerce [19] is a new emerging technology with greater scope. Mobile commerce is the buying and selling of goods and services through wireless handheld devices. Mobile devices mainly smart phones overcome laptops and desktops in many perspectives. Its size, portability, convience and so on. It is advantage to the customers during purchasing, customers usually carry a mobile device mainly a smart phone than laptops because of its smaller size and portability. Mobile commerce [9] has several applications, in that localization of products and services plays a major role. It is used to know user locations and the services requested by the user.

Association rule mining [15] is a popular and well researched method for discovering interesting relations between variables in large databases[12]. It is used to identify strong rules discovered in databases (e.g. Basket data analysis, clustering, classification). The association rule mining can be of two types:

1. Frequent item sets: The items that frequently occur in the database and satisfies the minimum support count.
2. Generate strong association rules from the frequent item sets: Satisfy the minimum support and minimum confidence based on the rules.

For example:- A user request a service A1 in the location L1 and request a service B1 in the location L2. This is an example of locations and service requested by the user in that location. (i.e., when a user goes to the location L1, he requests the service A1, and when he goes to the location L2, he requests the service B1)

This paper is aimed to satisfy the user needs based on their selection. It is mainly used to mine the rare frequent item from the database. In the past, only the frequently moved item was predicted to the user. It does not satisfy the user needs. The user behavior must be predicted efficiently based on both the movements and purchase transactions. In the past, the Association rule mining[10] is used to predict the frequent item from the database, but it takes more time. To overcome this, systolic tree implementation is proposed to predict the frequent item from the database in a short time. In the past, the mobile commerce behavior was predicted based on multiple users, but we are going to mine the mobile commerce behavior for individual user. Fig. 1a shows a moving sequence, where store labels indicate some transactions takes place. Fig. 1b shows the transaction records of a user, where item i1 was purchased when the user is in store A. For example, the mobile transaction generated by the user is {(A,{i1}), (B, {i2,i3}), (C, ∅), (D, {i4}), (E, ∅), (F,i6)}



Shop	items
A	i1
B	i2, i3
C	-
D	i4
E	not visited
F	i6

(a) (b)

Fig. 1. An example for mobile transaction sequence. (a) Moving trajectory (b) Transactions

The main contribution of our approach is to discover the user behavior efficiently and satisfy the user needs. In addition, a systolic tree implementation was proposed, to discover the frequent item in an efficient and time consuming manner. Based on the movements and purchase transactions, the user behavior can be predicted for each and every individual user.

The remainder of this paper is organized as follows. In Section 2, we briefly review some related work. Section 3, discuss the system architecture in detail. Section 4, describes about the proposed work and algorithms. Section 5, concludes this paper and discuss the future work.

2 RELATED WORK

Efficient algorithms for finding the frequent item set or sequences in very large database have been one of the key success of data mining research. In order to discover patterns from two-dimensional mobility data, Tseng et al. [2] first studied the problem of mining associated services in mobile web environments. It is mainly used to understand the behavior of mobile users. The patterns are mined based on two kinds of hierarchies, the location and service hierarchies.

In [3], Yun et al, proposed Mobile Sequential Pattern (MSP) to take moving path into consideration and add the moving path to the left hand and right hand. Mobile sequential pattern[3] takes both the moving and purchase patterns of the customers. The goal of this paper is to mine the user behaviour efficiently.

In [1], Y. Zheng et al, proposed a method for mining the interesting locations and travel sequences in a given geospatial region, in that TBHG is used to mine the multiple individuals' location histories. In this, they are creating a geo-related web community, in that they can upload GPS logs. Based on TBHG[1], they propose a HITS based model to know the users travel experience and interesting location within a region.

In [7], J. Pei et al, proposed a method, namely WAP-Mine, for discovering the web access patterns from web logs by using a tree-based data structure without candidate generation. Weblogs is like a storage medium, which contains the information about the accesses are recorded, including the URL, origin of request and timestamp. Weblogs is divided into several pieces by pre-processing technique. With the each piece of web log [7] they can't able to predict the frequent sequential pattern.

In [4] V.S. Tseng et al. modelled an efficient mobile behavior prediction system. When users moves between the stores in the mobile network, their location and services are stored in a mobile transaction database. It has an offline mechanism for mining and online engine for mobile behavior prediction.

In [8], D. Xin et al, overcome the sheer size in the frequent pattern mining with a tightness measure and representative pattern. The patterns are clustered with a tightness measure and representative pattern can be selected for each cluster. Several techniques are proposed[8], in that RPglobal is used to mine frequent patterns. RPlocal is much more efficient and quite good compression quality.

To overcome the difficulty in predicting the user behavior, [5] V.S. Tseng and K.W. Lin proposed SMAP-Mine to discover mobile users' sequential movement patterns with the service that was requested. By using the mobile,we can get various kinds of services in any time.To predict the user behavior is an difficult task,so they proposed a data mining method called SMAP-Mine[5], is used to predict the sequential mobile patterns,from that we can efficiently discover the user behavior.

In [11], Eric Hsueh-Chan Lu Et al. proposed a framework called MCE framework for mining and prediction of mobile users movements and purchasing transactions. Its goal is to predict the behavior of individual users. The mobile network database maintains detailed store information which includes the location. The mobile users moves between the stores and purchase the items, and all these information are stored in mobile transaction database.

3 SYSTEM ARCHITECTURE FOR MOBILE LOG PROCESS

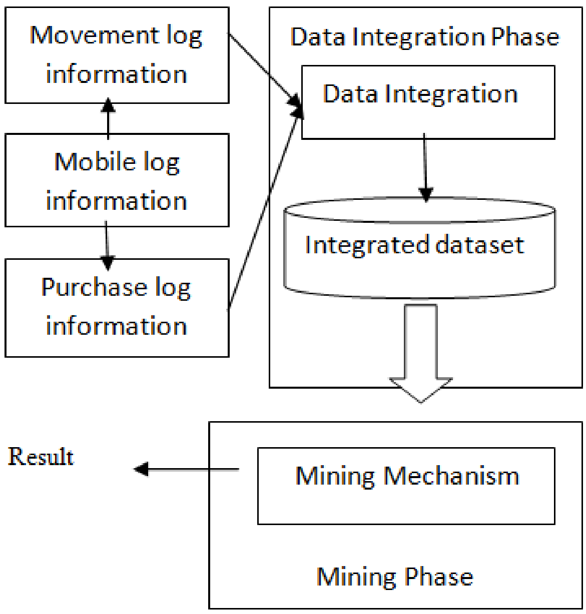


Fig. 2. System architecture for mobile log process

In this section, we represent the system architecture with the data mining module for mobile log pattern. It is conducted to extract the movement and purchase information based on the mobile log information. Fig. 2 describes the data mining system architecture for mobile log process. The work flow of the system is divided into two phases. This system is associated with the logs of users' movements and users' purchase transactions and all are stored in database. The first phase of the system architecture, Data integration phase, is to collect and integrate users' log into one dataset to efficiently discover the user behaviour information. For this phase, the attributes related to mobile user's behaviour will be extracted from the dispersed log files and joined to from an integrated log file. Next, Mining phase is an novel data mining method to discover the frequently moved item from the integrated log dataset. Finally, the best results are delivered, associated with the mobile log information and produces efficient and accurate result.

Table 1. Mobile Transaction Database

T_{id}	U_{id}	Mobile Transaction Sequence
1	1	$(A,\{i_1\}), (B,\emptyset), (C,\{i_3\}),(D,\{i_2\}), (E,\emptyset), (F,\{i_3,i_4\}), (I,\emptyset), (K,\{i_5\})$
2	1	$(A,\{i_1\}), (B,\emptyset), (C,\emptyset), (D,\{i_2\})$
3	1	$(A,\{i_1\}), (B,\emptyset), (C,\emptyset), (D,\{i_2\}), (E,\emptyset), (F,\{i_3,i_4\}), (I,\emptyset), (K,\{i_5\})$
4	1	$(A,\{i_1\}), (D,\{i_6\}), (C,\{i_5\})$

4 PROPOSED WORK

In this section, we discuss about the proposed work. Mobile commerce is used to predict the mobile user behaviour such as their movements and purchase transactions. It predicts the frequently moved item to the user based on their selection.

4.1 PROBLEM STATEMENT

When the user enters the store, the system predicts the frequently moved item to the user wireless handheld devices. It is a predetermined target, so it does not satisfy the user needs because the user may have a thought of buying another item. And also the time to retrieve information from the database is more. To overcome the issue, we are going to mine the rare association rule to satisfy the user needs. In this, after entering the store, the system will not predict the frequently moved item, the user can purchase the items based on their choice. Then the system will predict the possible items related to user choice.

4.2 FREQUENT PATTERN MINING

Frequent patterns are patterns that appear in a database frequently (e.g. a set of items,such as iphone and headset, that appear frequently in a transaction data set is a frequent itemset). A set is called frequent if its support is no less than a given absolute minimal support[10].Two measures are used they are, 1. Support and 2. Confidence.In support, the rule holds with support *sup* in *T* (the transaction data set) if sup% of transactions contain $X \cup Y$. In confidence, the rule holds conf in *T* if conf% of tranactions that contain *X* also contain *Y.* In the following, we describe the methods for mining frequent itemsets.

In this phase, we mine the frequent transactions (FTransactions) for each user by applying a modified Apriori algorithm [15]. Table 1 shows the mobile transaction database . At first, the support of each (store, item) pair is counted for each user. The patterns of frequent 1-transactionsare obtained when their support satisfies the user-specified minimal support threshold TSUP.

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; fi3; i4g) is generated by joining (F; fi3g) and (F; fi4g), because the user identifications and purchased stores of them both are U1 and F, respectively. Thus, we keep the patterns as frequent 2-transactions, when their support is larger than TSUP.

Finally, the same procedures are repeated until no more candidate transaction is generated. We use an item mapping table to relabel item sets in order to present F-Transactions for each unique item set, we use a symbol LIi (Large Item set 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.

4.3 PATTERN MINING

Pattern mining is a data mining method that involves finding existing patterns in data. In this context *patterns* often means association rules. The original motivation for searching association rules came from the desire to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. For example, an association rules "beer ⇒ potato chips (80%)" states that four out of five customers that bought beer also bought potato chips.

4.4 METHODOLOGY

4.4.1 USER INTERFACE

The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals—what is often called user-centered design. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it. Graphic design may be utilized to support its usability. The design process must balance technical functionality and visual elements (e.g., mental model) to create a system that is not only operational but also usable and adaptable to changing user needs.

Interface design is involved in a wide range of projects from computer systems, to cars, to commercial planes; all of these projects involve much of the same basic human interactions yet also require some unique skills and knowledge.

4.4.2 ENROLLMENT PROCESS

Enrollment process is adding the information in database. The enroll process involves shop information, customer information and item information.

Shop information is used to maintain the shop details like shop name and the shop position like latitude and longitude information. In this we simulate the shop and location based on X and Y position.

Item enrollment process is used to enroll the item details in the shop. The item details contain the item number, item name, and item cost.

The customer enrollment process is used to enroll the customer details that have process the mobile commerce. The customer details have customer username, name, city, mail id, mobile number. All these information are stored in the database.

4.4.3 MOBILE MOVEMENT PROCESS

Manage information about objects moving in two- (or higher) dimensional spaces are important for several emerging applications including traffic supervision, flight control, mobile computing, etc. In order to avoid frequent location updates, the database stores the items purchased and its location. M-commerce services will be able to capture the moving trajectories and purchase transactions of users.

Mobile behaviour predictions can be used by nonlinear models. The nonlinear models capture objects' movements with sophisticate regression functions. Thus, their prediction accuracies are higher than those of the linear models. Recursive Motion Function (RMF) is the most accurate prediction method in the literature based on regression functions.

4.4.4 SHOPPING PROCESS

After completing all the process, customer gets the option to move various shop based on mobile movement process. If the customers reach the shop he/she view the products from the shop. 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.

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.

4.5 GENERATION OF FREQUENT PATTERN

Although serval methods have been developed for mining frequent patterns and closed patterns, such mining frequently generates a huge number of frequent patterns. People would like to see only interesting ones. In our work we mine frequent patterns from transactional log based on systolic tree implementation.

4.5.1 SYSTOLIC TREE

A systolic tree is an arrangement of pipelined processing elements (PEs) in a multidimensional tree pattern. The goal of our architecture is to mimic the internal memory layout of the FP-growth algorithm while achieving a much higher throughput. The role of the systolic tree as mapped in FPGA hardware is then similar to the FP-tree as used in software. The formal definition of FPtree can be found in supplementary.

The design principle of the WRITE mode algorithm is that the built-up systolic tree should have a similar layout with the FP-tree given the same transactional database. The software sends a candidate pattern to the systolic tree. After some clock cycles, the systolic tree sends the support count of the candidate pattern back to the software. The software compares the support count with the support threshold and decides whether the candidate pattern is frequent or not. After all candidate patterns are checked with the support threshold in software, the pattern mining is done. The approach to get the support count of a candidate pattern is called candidate item set (pattern) matching.

4.5.2 CREATION OF SYSTOLIC TREE

A systolic tree is an arrangement of pipelined processing elements (PEs) in a multidimensional tree model. The goal of the architecture is to copy the internal memory layout of the FP-growth algorithm while achieving a much higher throughput. The function of the systolic tree is to mapped in FPGA hardware is then similar to the FP-tree as used in software. A simple example for the systolic tree implementation is shown in Fig 2.

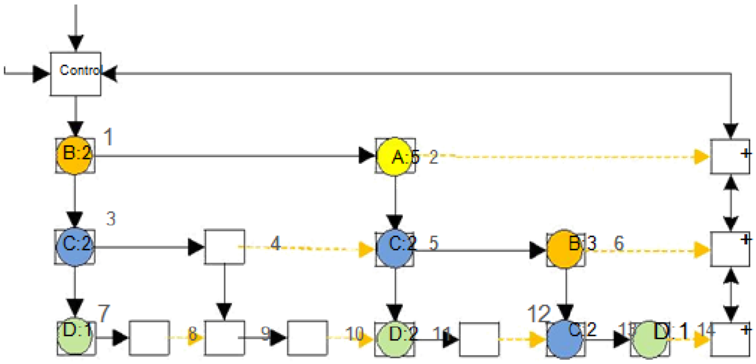


Fig. 3. The Systolic Tree Architecture

Figure 2 shows the static systolic structure where *K*=2 and *W* =3. Each node in the systolic tree architecture is also referred to as a processing element (PE). Each PE has its local data structure and corresponding operations upon re-ceiving signals from outside. The Fig. 2 denotes the three kinds of processing elements. The root PE is the control node discussed. The PEs in the rightmost column is the counting nodes which are specifically used for frequent item set dictation. The third kind of processing elements are the general PEs.

Each node has a counterpart in the general processing elements in Fig. 2, but the converse does not hold. Each general PEs has one input from its parent and two outputs to its child and siblings respectively. Each PEs has a connection with its leftmost child only. It sends the data to its rightmost child, then the data will be passed to its leftmost child and then through its siblings and through all children on the way. The general processing elements which do not contain any item are empty. The items in one transaction are transferred into the architecture in an ascending order, any PEs must contain a smaller item than that of its children.

Each PE in the systolic tree has three modes. In that WRITE mode is discussed as follow, During the building phase of tree, PEs are in WRITE mode. An item is loaded into the control PEs each cycle which in turn transfer each item into the general PEs. If the item is already present in a PE, the corresponding count value will be increased. Otherwise, an suitable empty PEs will be located for it. The algorithm for WRITE mode in each PEs is given in Fig. 4. The input of the algorithm is an item *t*. The *match* flag is set when the item in PEs matches *t*. The *Inpath* flag is not set when the PEs does not contain any item of the current transaction.

For example, the PEs under the control of PE in Fig. 2 must not contain the item B in a new transaction *{*A,B,C*}*. After all items are sent to the systolic tree, a control signal that state the termination of an old transaction and the start of a new one is sent to the control PE. The signal will be broad-casted to all PEs which reinitialize themselves for the next transaction. The initialization includes resetting of *match* and *Inpath* flags in the first line of Fig. 3.

```
Algorithm: WRITE mode(item  t)
match:= 0;  InPath:= 1;
(1)if  P E is empty  then
store the item t;
count:= 1;
match:= 1;
stop forwarding;
(2)if(t is in PE)and ( InPath=1) then
match:= 1;
count++;
stop forwarding;
(3)if(match=0) then
forward t to the sibling;
InPath:= 0;
else
forward t to the children
```

Fig. 4. WRITE mode Algorithm

Let's illustrate the creation of systolic tree with an exam-ple shown in Fig. 3. In order to clearly differentiate PEs a number in light scale is placed in the top-right corner. Sup-pose a new transaction {A,B,D} is to be added into the systolic tree. The control signal which indicates WRITE mode is first broadcasted from the control PEs. Then the transaction {A,B,D} is send to control PEs sequentially. When PE1 receives A, the step (3) in Fig.4 is triggered and A is forwarded to PE2. The *Inpath* flag in PE1 is set to zero. Step (2) is triggered in PE2. The *count* value in PE2 is increased by 1. The *match* flag is set to 1. PE2 stops forwarding A to other neighbors. The item A is sent to PE2 by PE1, then the second item B is sent to PE1 by the control node. Step (3) is triggered in PE1. Item B is sent to PE2. Since the *match* flag is set to 1, the item B is sent to PE5. Next, step (3) is triggered in PE5. B is then sent to PE6. The *count* value is increased by one in PE6. In the similar way, the item D is sent to PE14.

5 Experimental Evaluation

In this section, we conducted a series of experiments to evaluate the performance based on systolic tree implementation under various system conditions. All the experiments are implemented in Java on a 3.0GHz machine with 1GB of memory running windows XP.

5.1 Simulation Model

In the base experimental model, the network is modelled as a mesh network with size W, and there are 10,000 users in this network. The total number of services that the users may request is modeled by parameter N. The number of mobile transaction D is generated. When a customer moves among the cells for shopping, the mobile transaction sequences consists of a moving path and a set of transactions for each and every customer. To build the systolic tree the items in transactional database are sent to the tree one by one in each clock cycle. The time to build the systolic tree is less than the Apriori Algorithm.

$$\text{Precision} = \frac{P+}{P+ + P-}$$

$$\text{Recall} = \frac{P+ + P-}{R}$$

$$\text{F\_measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5.2 Generation of Mobile Transaction Sequences

In the experiments, the moving scenario with transaction is predicted and it is simulated. The mobile commerce service is a new application in the near future, we believe that the customers have the similar behaviours to those of them in the current data network when they first use this service. After the service is used by customers, the behaviour will then be changed according to their usage experiences. Thus, in this paper, the simulation model for generating synthetic mobile sequences is in the fact similar to that in the companion papers [13][14]. Explicitly the method for generating moving patterns is similar to that in [14] and the method for generating transactions is similar to that in [13].

In the experiments, *|D|* is the number of mobile transaction sequences generated. When a customer moves among cells for shopping in the MC environment, the mobile transaction sequence completed by this customer consists of a moving path and a set of transactions made in the corresponding cells. The starting position of each mobile sequential pattern can be either vistor location register (VLR) or home location register (HLR) and is randomly selected among these cells [2]. A moving path consists of cells moved by a user. The size of each moving path is determined from a Poisson distribution with mean equal to *|P|*. When a customer moves to a cell, the probability that this customer makes the transaction in this cell is denoted by *Pb*. For each cell, once the number of items is determined, the items that could be purchased in each cell are fixed. The method for generating transaction data in each cell is similar to the one in the prior work [13]. In the mobile commerce environment, people tend to buy sets of items together, which are also called potential maximal frequent sets.

In the Fig. 5, shows the support count value between the two algorithms in that, s varies from 1.5% to 0.25%. As the minimum support decreases, the execution time of the existing algorithms increases because of the increases in the total number of frequent patterns.

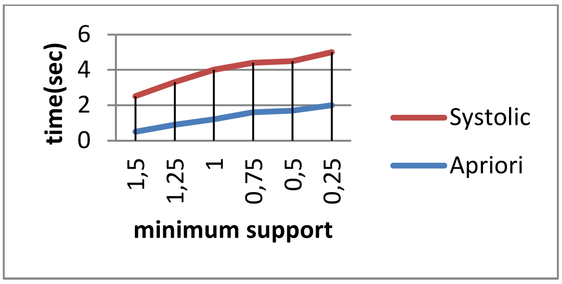


Fig. 5. Minimum support

5.3 Location Prediction by Mobile Commerce

This series of experiments show the effects on the prediction for location , services, and L&S with varying confidence value.we observe that it can be fortuitous to predict the correct location by the probability of 1/25 when the networksize jWj is set to 5*5 (i.e. there are 25 nodes in the mesh network). Take a small network for explanation. Let the network size be 2*2Z4 nodes, namely a, b, c, and d. We now want to predict the next location of a mobile user who is going to location b with the past behavior (a, I). Using (a, I) as the LHS to predict the next location means that the user is following the discovered event. In this case, if the user is following the discovered event, the prediction will exactly be correct. Otherwise, if the user moves randomly, the prediction will still have ¼ probability to be correct although it is fortuitous. We name this kind of prediction as fortuitous prediction. As can be seen, the probability of fortuitous prediction is relatively low. But in mobile commerce it is relatively higher than the other methods.

Based on the simulated result, we compare the mining time of systolic tree with Apriori algorithm in Fig. 6. The mining time of the software algorithm is collected from a PC with Pentium D 3GHz CPU, 2GB RAM. The benchmark is several transactional dataset which is collected and integrated from the Keel repository datasets. This database has several transactions. In our experiments, we change the support count threshold to get different numbers of frequent items. Note that the run time of the Apriori algorithm is closely related to the size of the database while the run time of the systolic tree

implementation is only determined by the number of frequent items. It can be observed that the threshold size of the systolic tree must be no more than 11 in order to be faster than Apriori algorithm. When the size of the systolic tree is 10, the mining speed is 24 times faster than Apriori.

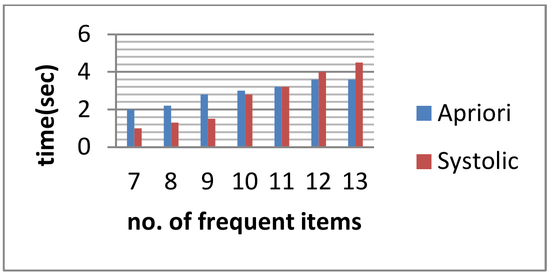


Fig. 6 Execution time comparison

6 CONCLUSION

In this system, a novel framework, namely Mobile Commerce, is used for mining and prediction of the mobile user behavior such as their movements and purchase transactions. It is used to recommend the stores and items to the unknown user. When mobile users move between the stores, the system predicts the frequently moved items to the user based on their selection, which includes the mobile information such as user identification, stores, and items purchased are stored in the mobile transaction database. In that a new systolic tree-based algorithm is an efficient and effective one to mine the frequent item sets. It produces better results in performance than the other algorithms. Our future work is to implement it in real world.

REFERENCES

[1] 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.

[2] V.S. Tseng and C.F. Tsui, “Mining Multi-Level and Location- Aware Associated Service Patterns in Mobile Environments,” IEEE Trans. Systems, Man and Cybernetics: Part B, vol. 34, no. 6, pp. 2480-2485, Dec. 2004.

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

[4] V.S. Tseng, H.C. Lu, and C.H. Huang, “Mining Temporal Mobile Sequential Patterns in Location-Based Service Environments,” Proc. Int’l Conf. Parallel and Distributed Systems, pp. 1-8, Dec. 2007.

[5] 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, June 2006.

[6] 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. Computer Science Security, pp. 285-291, Feb. 2007.

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

[8] D. Xin, J. Han, X. Yan, and H.Cheng, “Mining Compressed Frequent-Pattern Sets,” Proc. Int’l Conf. Very Large Data Bases,pp. 709-720, Aug. 2005.

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

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

[11] Eric Hsueh-Chan Lu, Wang-Chien Lee and Vincent S.Teng, “A Framework for Personal Mobile Commerce Pattern Mining and Prediction,” IEEE Transactions on Knowledge and Data Engineering, vol.24, no.5, May 2012.

[12] J. Han and M. Kamber, Data Mining: Concepts and Techniques,second ed. Morgan Kaufmann, Sept. 2000.

[13] J.-S. Park, M.-S. Chen, and P.S. Yu, “ An Effective Hash based algorithm for mining association rules,” in Proc. ACM SIGMOD Conf., May 1995, pp 175-186.

[14] W.-C. Peng and M.-S. Chen, “Developing data collection schemes by incremental mining of user moving patterns in a mobile computing system,” IEEE Trans. Knowl. Data Eng., vol. 15, pp. 70-85, Feb 2003

[15] R. Agrawal and R. Srikant, “Fast Algorithm for Mining Association Rules,” Proc. Int’l Conf. Very Large Databases, pp. 478-499, Sept. 1994.

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

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

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

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

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

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

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

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

[24] J. Han and M. Kamber, Data Mining: Concepts and Techniques, second ed. Morgan Kaufmann, Sept. 2000.

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