Mobile User Data Mining: Mining Relationship Patterns

John Goh David Taniar

School of Business Systems
Monash University, Clayton Vic 3800 Australia
{Jen.Goh, David.Taniar}@infotech.monash.edu.au

Abstract. Mobile user data mining focuses on finding useful and interesting knowledge out from raw data collected from mobile users. Frequency pattern and location dependent mobile user data mining are among the algorithm used in this field. Parallel pattern, our previous proposed method, extracts how a group of mobile users makes similar decisions, such as by moving towards the similar direction, or by viewing similar contents at the same time. Parallel pattern is triggered group behaviour of mobile users. This paper reports our refinement work on parallel pattern which incorporated *refinement of the relationships among parallel patterns, or relationship pattern*, which shows how 'similarities of decisions' are related to each other. Effects found are such as conditional relationship, where one parallel pattern has to happen before the next one occurs. Other effects includes associative, sequential and loop pattern effects. Our performance evaluation reports how relationship pattern performs in real life dataset and synthetic dataset and discusses some potential implementation issues.

1 Introduction

With the penetrate rate [1] of mobile technologies into the consumer market industry, many consumers are using mobile equipments in different situations in their life. Since mobile users usually carries mobile equipments most of the time, and the mobile equipment stays in touch with the mobile network access point at all times, source data that encompasses the behaviour of mobile users could be captured. Capturing this large amount of source data helps producing quality knowledge [11, 13, 15].

Mobile user data mining [11-15, 24] is a domain of data mining which emphasizes on developing methods into finding useful and interesting patterns from source data generated by mobile users. Data mining [3, 4] has been studied in many domains such as time series environment [5, 16-18], web based environment [6, 9, 10], geographical based environment [7, 8, 21, 22, 25] and location dependent environment [12, 20, 22, 23]. Enhancements in the area of data mining in general are also being developed, such as security enhancements [27] and efficiency in rules generation enhancements [26, 28, 29].

Mobile user data mining [12, 15, 22, 24] focuses extensively on mobile users and the mobile equipment that they carry. Parallel pattern [13, 15] is our previously proposed method designed to find out similarities of decisions made by mobile users. In a mobile environment, there is a pattern that exists whereby similarities of decisions are taken among mobile users within a short period of time. The objective of this paper is to overcome the shortcomings of the previous proposed parallel pattern that only shows only one way relationship of similar change of states. In the interest of space, details on mobile user data mining is not provided.

2 Background

Parallel pattern [13, 15] finds out similarities of decision such as physical movements, or access to contents from the web, among mobile users. The word parallel in the parallel pattern reflects the event that happens at the same direction at the same time, such as two separation events from mobile user 1 and mobile user 2 to go from a fashion shop to a grocery shop within a specified period of time. The pattern in the word parallel pattern reflects the component to find out the pattern trend. If there is a pattern, 1) it is more likely to have reasons behind it 2) the pattern is likely to be repeated in the future. Therefore the study of parallel pattern provides a framework for understanding of mobile environment.

Definition 1 (General): The general definition of events, time, locations and mobile users in the mobile environment is as follows.
Let $E = \{e_1, e_2, \ldots, e_n\}$ be the set of events.
Let $T = \{t_1, t_2, \ldots, t_n\}$ be the set of time.
Let $L = \{l_1, l_2, \ldots, l_n\}$ be the set of location or theme.
Let $M = \{m_1, m_2, \ldots, m_n\}$ be the set of mobile users.

Definition 2 (Event): The mobile environment is a space over a time series where mobile users move freely as they wish and intends. Each event is recorded with the time and the mobile user identification and their movement from and movement to in a single event.

$$\sum_{time=0}^{time=n} [\sum_{m=0}^{m=\max} visits(from, to)]$$

Definition 3 (Window): The window is a specified timeframe, from *time_start* to *time_end* where events within this window are treated as occurred within the same window. Two events happening in two different window is considered as remote to each other such that: $E\{M, L(from, to), T]$, where $T >$ *time_start* and $T <$ *time_end.*

Definition 4 (Parallel Pattern): The parallel pattern is a trend evidenced by a higher than threshold volume of similar parallel activities occurring within the same window period of time such that: Parallel Pattern = $E(from, to)$, where $T >$ *time_start* and $T <$ *time_end* and $\forall E \ (from, to) : E_x(from, to) = E_y(from, to)$.

There are two types of parallel pattern [13, 15], namely *physical parallel pattern* and *logical parallel pattern*. The principle behind both is the same but the application environments among them are different. Physical parallel pattern looks at the physical location that mobile user travels, which are identified by means of geographical coordinates. Logical parallel pattern, on the other hand, looks at how the current logical state of the mobile user. This can be described as such that mobile user being currently in a shopping centre, even when some are in a shopping centre in the north of the city, and some are in a shopping centre in the south of the city. Logical parallel pattern treats them the same.

3 Proposed Method: Relationship among Parallel Patterns

Relationship parallel pattern is our proposed method, which examines the relationships among parallel patterns – relationship patterns. There are four kinds of relationships among parallel patterns in a static node. These four relationships are namely *conditional*, *associative*, and *sequential* and *loop* relationship among parallel patterns.

3.1 Conditional Relationship

Definition 5 (Conditional Relationship): Let *P* be a list of parallel patterns found from a mobile user data mining episode. Let *C* be the condition for which two parallel patterns exists. Let $P = \{p_1, p_2, p_3, \ldots, p_n\}$. Let $C = \{c_1, c_2, c_3, \ldots, c_n\}$, where P is a list of parallel patterns, and C is a list of conditions. Let C_{ref} to be the reference condition which determines the existence of the relationship. Conditional relationship for two parallel patterns p_x, p_y is such that: C_n (p_1, p_2). Where $C_n = C_{ref.}$ Consider there is a conditional parallel pattern, *P*1 being Parallel Pattern 1 and *P*2 being Parallel Pattern 2, {If *P*1 = True Then *P*2} in the sample which consists of {*P*1, *P*2, *P*3, *P*4}. *P*1 must occur first before any *P*2 occurrence can occur. Once *P*2 have occurred, and if *P*1 occur after *P*2, then the statement no longer holds true. On the other hand, *P*2 must not occur before *P*1 have occurred. Finally, the statement must satisfy the confidence requirement, that is, must be sufficient frequent enough within the window sample in order to be qualified as a conditional parallel pattern.

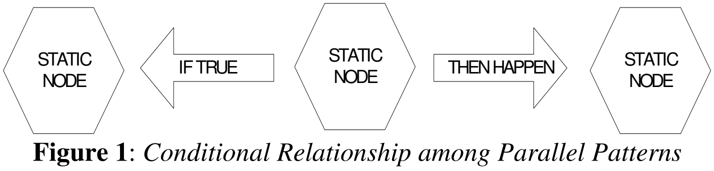

Figure 1: *Conditional Relationship among Parallel Patterns*

Figure 1 shows the diagrammatic representation of conditional parallel pattern. Each arrow represents a parallel pattern found from previous data mining exercise, and all parallel patterns were drawn from within a static node in the centre. In the diagram, the left arrow shows a *TRUE* sign and the right arrow shows a *THEN HAPPEN* signs. It represents a conditional relationship in between these two parallel patterns. For example, consider the parallel pattern on the left is *P*1 and the parallel pattern on the right is *P*2. If the algorithm finds out at most of the time, *P*2 only happens after *P*1 have occurred, then it can determine the confidence of conditional relationship among these two parallel patterns. Conditional parallel pattern occurs when one parallel pattern *B* occurs only after parallel pattern *A* occurred. It is not a conditional parallel pattern if there are instances in the window where parallel pattern *B* occurred before the occurrence of parallel pattern *A*. Therefore, this relationship must hold true at all times.

Figure 2 shows the algorithm for conditional parallel pattern. The algorithm first parses the list of source data and checks one set of parallel patterns which consists of two parallel patterns. Conditional parallel pattern is found when Parallel Pattern *A* always occurs with Parallel Pattern *B* within the same or immediately after the same time point, and that Parallel Pattern *B* never exist without the existence of Parallel Pattern *A*. When both of the conditions are satisfied, the two parallel patterns are marked as conditional parallel pattern.

```
Boolean Conditional Parallel Pattern (Pattern 1, Pattern 2) {
    For I = 0 to countPatterns (Parallel Patterns) Do
        Boolean Result = False;
        For J = 0 to countPatterns (Parallel Patterns) Do
            X = Parallel Pattern 1;
            Y = Parallel Pattern 2;
            If (X < Y) && (X !> Y) Then Result = True;
            Else Result = False;
    Return Result;
}
```

Figure 2: *Conditional Parallel Pattern*

3.2 Associative Relationship

Definition 6 (Associative Relationship): Let *T* be a time sequence of $\{t_1, t_2, .., t_n\}$. Let *P* be a list of parallel pattern $\{p_1, p_2, .., p_n\}$. Consider the following list of input $\{t_1=p_1, t_2=p_1, p_2, t_3=p_3, t_4=p_4\}$ and the unit of buffer is 1 time unit. Therefore, since p_1 and p_2 occurred at t_2 itself, it qualifies as an associative parallel pattern because they both occurred at a precise time point. On the other hand, the fact that p_1 and p_2 occurred at t_1 and t_2 are considered relatively near as the time buffer is 1 unit, which is exactly what is happening right now. Therefore, the above fact also qualifies p_1 and p_2 as an associative parallel pattern: (E_x, E_y) : $time(E_x) = time(E_y)$, and $frequency\ (E_x, E_y) \geq frequency\_threshold.$

Figure 3 shows the diagrammatic representation of associative parallel pattern. It can be observed that during time 1, two parallel patterns occurred at the same time. During time 2, there are also two parallel patterns occurred at the same time. Figure 4 shows the algorithm for mining associative parallel pattern. The algorithm first receives the list of transactions and the conditions specified by the user. Conditional parallel pattern is designed to find out frequent instances that two parallel pattern occurring within the relatively same period of time, in this case, ± 5 seconds. Once they are found, they are marked as associative parallel pattern. Associative properties occur when two parallel patterns occurring at the same time. As the chances of two parallel patterns happening at precisely the same time, such as precision up to 1 millisecond is very rare, therefore, it should be adjusted so that parallel patterns that are happening at somewhere nearby timeframe will prevent the situation of too much precision returning too little result.

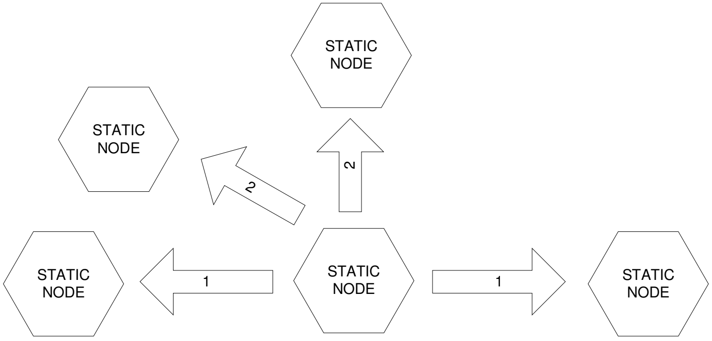

Figure 3: *Associative Parallel Pattern*

```
  Boolean Associative Parallel Pattern (Parallel Pattern List) {
      Boolean Result = False;
      For I = 0 to countSequence(Parallel Pattern List) Do
          If (countFrequency(I) > freq_threshold) Then
              List(I).Status = Frequent;
          Else List(I).Status = Infrequent;
      For All List.Status = Frequent Do
          L = formCombination(attribute_length);
          L.frequency = countFrequency(L, List);
          If (L.frequency > support) Then Result = True;
          Else Result = False;
  }
```

Figure 4: *Associative Parallel Pattern*

3.3 Sequential Relationship

Definition 7 (Sequence Relationship): The sequence relationship pattern $\{e_1, e_2, e_3, e_4\}$ represent a sequence of events, each item being an event, such that one event happens one after the previous event. In this case, $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4$. In this case, $e1$ depends on nothing, e_2 depends on e_1, e_3 depends on e_2 and e_1, e_4 depends on e_3, e_2 and e_1. Therefore, sequential parallel pattern can be seen as a sequence of conditional parallel pattern. Figure 5 illustrates: $frequency(E_{\text{time=1}} \rightarrow E_{\text{time=2}} \rightarrow E_{\text{time=3}} \rightarrow E_{\text{time=4}}) \geq frequency\_threshold$.

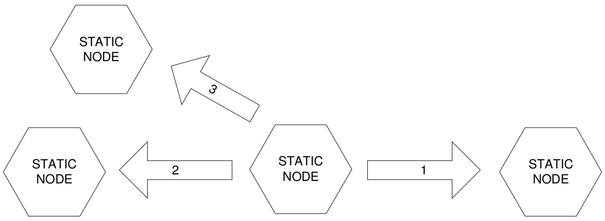

Figure 5: *Sequential Parallel Pattern*

Figure 6 shows the algorithm for mining sequential parallel pattern. The algorithm first receives the list of parallel pattern and the confidence threshold. After receiving these inputs, the list of transactions in the windows are examined by matching each pass of window with a sequence of possible sequential parallel pattern. The frequency of occurrence for each sequence are then recorded and divided by the window size to show the confidence of the sequence being a sequential parallel pattern. If the condition are satisfied, that is the sequence occurs frequent enough to be considered high confidence, then the sequence will be accepted as a sequential parallel pattern.

```
Boolean Sequential Parallel Pattern (Parallel Pattern List) {
    For I = 0 to formSequence(Parallel Pattern List) Do
        Boolean Result = False;
        SQ = currentSequence();
        If countOccurence(SQ) > sequence_threshold) Then
            If (last(SQ) = nextFirst(SQ) - 1) Then Result = False;
            Result = True;
        Else Result = False;
    Return Result;
}
```

Figure 6: *Sequential Parallel Pattern*

3.4 Loop Relationship

Definition 8 (Loop Relationship): Consider the sequential parallel pattern of {p_1, p_2, p_3, p_4, p_5}. It qualifies as a loop parallel pattern when p_1 occurs after p_5 have occurred and p_1, p_2, p_3, p_4 and p_5 follows a sequence relationship, that is, one occurs after the previous.

Let S be a list of sequence. S_{ref} is the reference sequence in which it has occurred frequently to be considered a sequential relationship pattern. Loop Relationship = $time1(S_{\text{ref}}) \rightarrow time2(S_{\text{ref}})$ By finding the circular relationships among parallel patterns, it can be said that each circular relationship refers to a phenomenon. A phenomenon is something reoccurs at various frequencies, but the whole cycle do repeats itself. Of course, the window size must be large enough to accommodate the possible loop size of the circular flow. Furthermore, there must exist for a few repetition of the circular relationship in order to suggest a high confidence of the statement.

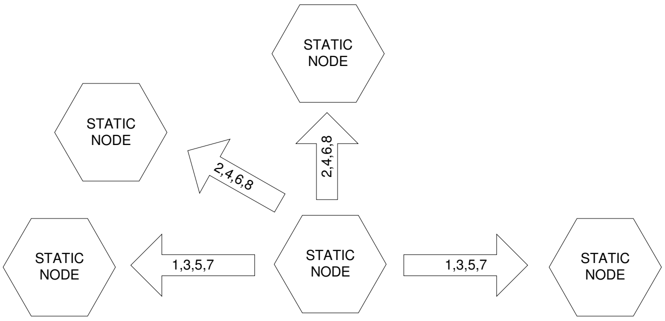

Figure 7: *Loop Parallel Pattern*

Figure 7 shows the diagrammatic representation of loop parallel pattern. It can be seen that there are two sets of parallel pattern that occurs over the same period amount of time, namely time series {1, 3, 5, 7} and {2, 4, 6, 8}. Figure 8 shows the algorithm to find the loop characteristics among the parallel patterns. The loop characteristic is an extension of multiple reoccurrence of sequential parallel pattern without any break. If a loop parallel pattern is found, it shows that there is a highly significant reoccurrence of patterns, in which during random events are highly unlikely. However, this algorithm is written to preserve the need in case this actually occurs so that it could be detected.

```
Boolean Loop Parallel Pattern (Parallel Pattern List) {
    For I = 0 to formSequence(Parallel Pattern List) Do {
        Boolean Result = False;
        SQ = currentSequence();
        If countOccurence(SQ) > sequence_threshold) Then {
            If (last(SQ) = nextFirst(SQ) - 1) Then Result = True;
            Else Result = False;
        Else Result = False;
    Return Result;
}
```

Figure 8: *Loop Parallel Pattern*

4 Performance Evaluation

Performance evaluation is performed under an IBM PC Pentium IV 384MB memory environment. Dataset used are *R*, *R3* and *R5* derived from dataset generator, with random seed derived from generation of integers with repetition but fixed within a specified range. The objective is to examine the relationship among the window size towards the responsiveness of finding relationships among parallel pattern. Therefore, data for *window size* 10 is towards *time* 10, *window size* 20 is towards *time* 20, *window size* 30 is towards *time* 30. Figure 9 shows the chart for random source data from *random.org* [19]. The randomness is sourced from atmospheric noise and fed into the seed for random number generator.

Variable	Description
window	Size of window used to perform data mining.
frequency	Frequency of occurrence for a certain pattern.

Dataset	Description
R	Fully random dataset.
R3	Sequence repeated after 3 units of time interval.
R5	Sequence repeated after 5 units of time interval.

Figure 10: D*escription for R, R3 and R5 Datasets*

Figure 10 shows the description of variables involved in the performance evaluation and description of individual dataset used for performance evaluation, namely {*R*, *R*3 and *R*5}. Due to lack of real life dataset, performance evaluation is conducted using synthetic dataset. These synthetic dataset are obtained from a random number generator with the seed being replaced via atmospheric noise. *R*, also known as *Random* or *R*1 is a truly random dataset, and it sets as a baseline of a data where no pattern can be found as the whole dataset is random. *R*3 and *R*5 are repeated dataset, where *R*3 have a repeat every 3 units of time interval and *R*5 have a repeat every 5 units of interval. The reason for doing so is to create some form of artificial patterns, so that the algorithms can extract the patterns out.

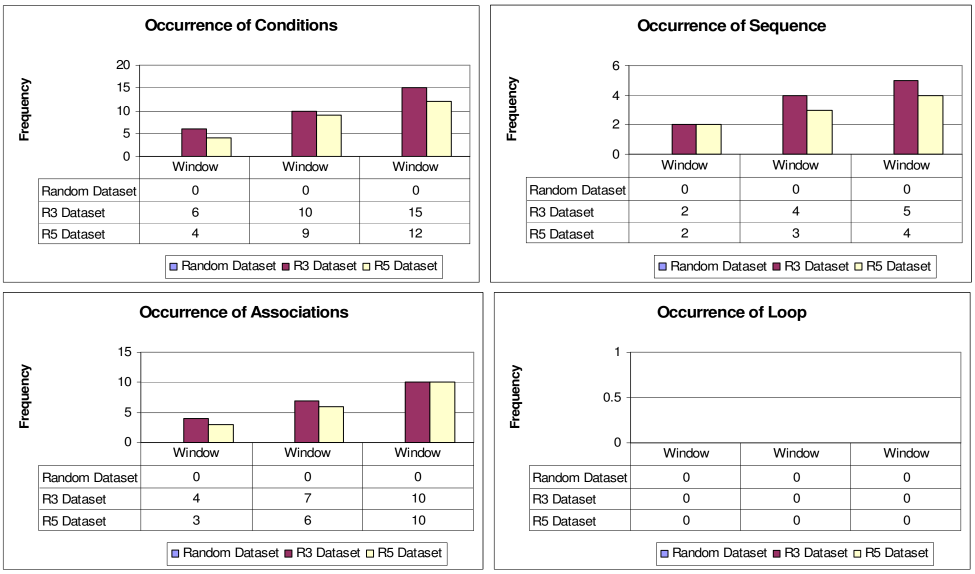

Figure 11: *Performance Result using R, R3 and R5 Datasets*

Figure 11 shows the performance result for the 4 different mining algorithms under the 3 different dataset configuration under 3 different window sizes. For conditional relationship pattern, it was observed that during the course of testing Dataset *R*3 always returns higher result than Dataset *R*5. For sequential relationship pattern, it was observed that the trend of graph is growing gradually, and this is caused by increasing window size in the *x* axis. *R*3 is always higher than *R*5 because there are more complete repetitions in *R*3 than *R*5 dataset. For associative relationship pattern, it was observed that the reading is growing up because window size is increasing in the *x* axis. *R*3 and *R*5 happen to be equal number of reading because of coincidence. For loop relationship pattern, it was observed that no loop parallel pattern is found, due to its rare occurrence.

5 Conclusion and Future Work

This paper reports our work on finding relationships among parallel pattern - relationship pattern. *Conditional relationship* represents a movement patterns happening one after another such as if users move from *A* to *B*, then it is likely that users will move from *C* to *D* as well. *Sequence relationship* represents a movement patterns happening in sequence such as if users move from *A* to *B*, then it is likely for users to move from *C* to *D*, and the it is likely that users will move from *E* to *F* after this. *Association relationship* represents movement patterns happening at the same time such as users will move from *A* to *B* or *C* to *D* at the same time. *Loop relationship* represents patterns that repeat itself once finished and it is a sequential relationship where the last occurrence of pattern connects back to the first occurrence of pattern, such as users will move to *A* to *B*, then *C* to *D*, then *E* to *F*, and then pattern of users moving from *A* to *B* will occur again.

It represents a progress towards a higher level of mobile user data mining from the previously found patterns in the lower level. With higher level of knowledge being found, higher level knowledge represents knowledge that are rich and simplified as higher level knowledge is extracted based on the summary of previously found knowledge. It has been concluded that the ability for the proposed method being able to find knowledge at a higher level, it is recommended for mobile user data mining. Our future work for this research area is to extend the concept of high level quality knowledge mining over other areas such as *frequency pattern* [11], *group pattern* [24], *association rules* [3], and *sequential patterns* [4].

References

1. Statistics for Telecom Services for 2004. http://www.ida.gov.sg/idaweb/media/infopage.jsp?infopagecategory=factsheet:factfigure&versionid=1&infopageid=I2703 Accessed: 20 June 2004
2. University of California (Irvine) Library - Knowledge Discovery in Database Archive. http://kdd.ics.uci.edu/ Accessed: 1st Dec 2004
3. R. Agrawal and R. Srikant. Fast Algorithms for Mining Association Rules. In Proc. *20th International Conference on Very Large Data Bases*, pp. 487-499, 1994.
4. R. Agrawal and R. Srikant. Mining Sequential Patterns. In Proc. *11th International Conference on Data Engineering*, pp. 3-14, 1995.
5. S. Y. Chen, X. Liu, Data mining from 1994 to 2004: an application-oriented review, *Int Journal of Business Intelligence and Data Mining*, vol. 1, no. 1, pp. 4-21, 2005.
6. V. Christophides, G. Karvounarakis, and D. Plexousakis. Optimizing Taxanomic Semantic Web Queries using Labeling Schemes. *Journal of Web Semantics*, **1**(2):207-228, 2003.
7. I. Claude, J.-L. Daire, and G. Sebag. Fetal Brain MRI: Segmentation and Biometric Analysis of the Posterior Fossa. *IEEE Transactions on Biomedical Engineering*, **51**(4):617-626, 2004.
8. D. E. Cooper, P. Ezhilchelvan, and I. Mitrani. High Coverage Broadcasting for Mobile Ad Hoc Networks. *Lecture Notes in Computer Science,* vol. 3042, pp. 100-111, 2004.
9. D. L. Lee, M. Zhu, H. Hu, When location-based services meet databases, *Mobile Information Systems*, vol. 1, no. 2, pp. 81-90., 2005.
10. M. Eirinaki and M. Vazirgaiannis. Web Mining for Web Personalization. *ACM Transactions on Internet Technology*, vol. 3, no. 1, pp. 1-27, 2003.
11. Goh, J and Taniar, D., Mining Frequency Pattern from Mobile Users. *Knowledge-Based Intelligent Information & Engineering and Systems, Lecture Notes in Computer Science Part III,* vol. 3215, pp. 795-801, 2004.
12. Goh, J and Taniar, D., Mobile Data Mining by Location Dependencies. *5th International Conference on Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science,* vol. 3177, pp. 225-231, 2004.

13. Goh, J and Taniar, D., Mining Physical Parallel Pattern from Mobile Users. *International Conference on Embedded and Ubiquitous Computing, Lecture Notes in Computer Science,* vol. 3207, pp. 324-332, 2004.

14. Goh, J and Taniar, D., An Efficient Mobile Data Mining Model. *2nd International Symposium on Parallel and Distributed Processing and Applications, Lecture Notes in Computer Science,* vol. 3358, pp. 54-59, 2004.

15. Goh, J and Taniar, D., Mining Parallel Pattern from Mobile Users. *International Journal of Business Data Communications and Networking,* vol. 1, pp. 50-76, Jan - Mar 2005.

16. J. Han, G. Dong, and Y. Yin. Efficient Mining of Partial Periodic Patterns in Time Series Database. In Proc. *International Conference on Data Engineering*, pp. 106-115, 1999.

17. J. Han, W. Gong, and Y. Yin. Mining Segment-Wise Periodic Patterns in Time Related Databases. In Proc. *In Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*, pp. 214-218, 1998.

18. J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In Proc. *In Proc. Int. Conf. Special Interest Group of Management of Data*, pp. 1-12, 2000.

19. M. Harr. Random.org. http://www.random.org Accessed: 1st August 2004

20. G. Kastaniotis, N. Zacharis, T. Panayiotopoulos, and C. Douligeris. Intelligent Web Prefetching Based upon User Profiles - The WebNaut Case. *Lecture Notes in Artificial Intelligence,* vol. 3025, pp. 54-62, 2004.

21. K. Koperski and J. Han. Discovery of Spatial Association Rules in Geographical Information Databases. In Proc. *4th International Symposium on Advances in Spatial Databases*, pp. 47-66, 1995.

22. D. L. Lee, J. Xu, B. Zheng, and W.-C. Lee. Data management in location-dependent information services. *Pervasive Computing, IEEE*, vol. 1, no. 3, pp. 65-72, 2002.

23. J. Li, B. Tang, and N. Cercone. Applying Association Rules for Interesting Recommendations Using Rule Templates. *8th Pacific-Asia Conference on Knowledge Discovery and Data Mining 2004, Lecture Notes in Artificial Intelligence,* vol. 3056, pp. 166-170, 2004.

24. E.-P. Lim, Y. Wang, K.-L. Ong, and et al. In Search of Knowledge About Mobile Users. *ERCIM News*, vol. 1, no. 54, pp. 10, 2003.

25. W.-C. Lin, D.-Y. Liao, C.-Y. Liu, and Y.-Y. Lee. Daily Imaging Scheduling of an Earth Observation Satellite. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 35, no. 2, pp. 213-223, 2005.

26. T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda1. Discovery of Maximally Frequent Tag Tree Patterns with Contractible Variables from Semistructured Documents. *8th Pacific-Asia Conference on Knowledge Discovery and Data Mining 2004, Lecture Notes in Artificial Intelligence,* vol. 3056, pp. 133-144, 2004.

27. S. R. M. Oliveira, O. R. Zaiane, and Y. Saygin. Secure Association Rule Sharing. *8th Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2004, Lecture Notes in Artificial Intelligence,* vol. 3056, pp. 74-85, 2004.

28. H. C. Tjioe, D. Taniar, Mining Association Rules in Data Warehouses, *Intl. J. of Data Warehousing and Mining*, vol. 1, no. 3, pp. 28-62.

29. P. K. C. Tse, W. K. Lam, K. W. Ng, C. Chan, An Implementation of Location-Aware Multimedia Information Download to Mobile System, *Journal of Mobile Multimedia*, vol. 1, no. 1, pp. 33-46.