

Efficient Parallelism for Mining Sequential Rules in Time Series Data: A Lattice Based Approach*

Biplab Kumer Sarker¹ and Kuniaki Uehara²

¹Faculty of Computer Science, University of New Brunswick, Fredericton, Canada

²Department of Computer and Systems Engineering, Kobe University, Japan

Summary

A parallel algorithm based on a lattice theoretic approach is proposed to find the rules among patterns that sustain sequential nature in the multi-stream time series data. The human motion data captured by motion capturing system considered as multi-stream multidimensional data is used as real time data set. The data set is transformed into sequences of symbols of lower dimension due to its complex nature. The relations among symbols are expressed as "rules". The proposed algorithm is implemented on a Distributed Shared Memory (DSM) multiprocessors system. The experimental results justify the efficiency of finding rules from the sequences of the patterns for time series data by achieving significant speed up comparing with the previous reported algorithm.

Key words:

Data Mining, Time series data, Parallel algorithm, Association rule and Multiprocessor system.

Introduction

Time series data mining has recently become an important research topic and is earning substantial interest from both academia and industry. It sustains the nature of multi-stream data due to its characteristics. An example of such a type could be "if company A's stock goes up on day 1, B's stock will go down on day 2 and C's stock will go up in day 3, i.e. some of the events of company A influences to occur some of the events of company B and C. If we analyze the multi-stream of time series for some stock prices and can discover correlations among all the streams, then the correlations can help us to decide better time to buy stocks. So, the correlations can be expressed as rules. It is quite effective and useful to analyze the underlying behavior of time series data to investigate the correlations among its multi-stream. Strong dependencies capture

structures in the streams because it indicates that there exists a relationship between their constituent patterns, that occurrences of those patterns are not independent. These correlations depicted as rules are useful to describe the sequential nature of time series data. Therefore, the dependencies can be termed as association rules in multi-stream.

Researchers have been concentrating to find out these dependency rules from the large and complex data sets such as prices of stocks, intensive weather patterns, astrophysics databases, human motion data etc. Basically, except human motion data, these are one dimensional data in nature. In this paper, we focus on human motion data deemed as high dimensional multi-stream due to its features [4]. The correlations discovered from multi-stream of human motions data characterize a specific motion data. Furthermore, those correlations become basic elements that can be used to construct motion with combinations of themselves, just as phonemes of human voice do. These basic elements are called primitive motions. As a result, we can use these primitive motions as indices to retrieve and recognize motion for creating SFX movies, computer graphics and animations.

The following section discusses the problem statement and various research findings related to the topics. Section 2 introduces the structure of human body as multi-stream and briefly discusses the way of converting high dimensional motion data into sequence of symbols of multi-stream representing as lower dimensional data. The next section discusses the discovery process of association rules from the symbols of multi-stream. It is very expensive, time consuming and computationally intensive task to discover association rules from these kinds of huge amount of time series data represented as symbols of multi-streams. Hence, section 3 illustrates the advantage of using lattice theoretic based approach over Distributed Shared Memory (DSM) Multiprocessor system. A parallel algorithm using the lattice theoretic approach is also presented here. The algorithm can efficiently find the sequence of correlations among the body parts that

Manuscript received July 5, 2006.

Manuscript revised July 25, 2006.

perform various motions in a small amount of time. The results are discussed elaborately in the section and compared with one of our previous reported algorithm. Finally section 4 concludes the paper by presenting the direction of future research.

1.1 Problem Statement

Let A be a set of distinct attributes. With no loss of generality, any subset of A can be represented as a sequence that is sorted according to the lexicographic order of attribute names. For instance, $\{a, c\}$ and $\{c, a\}$ represent the same subset of $\{a, b, c\}$ that is identified by the sequence ac . We term such a sequence as a *canonical attribute sequence (cas)* [12]. There exists a one-to-one mapping between the set of all *cass* and the power set, denoted 2^A , so that the set of *cass* can be identified with 2^A . 2^A can be treated as a Boolean lattice where \emptyset (i.e. empty *cas*) and A (i.e. the complete *cas*) are, respectively, the bottom and top elements. The order in 2^A is denoted as \leq , coinciding with set inclusion; where $s \leq u$ reads as s is a *partial cas* or a *subcas* of u .

Given a threshold ratios σ and γ such that $0 \leq \sigma \leq 1$ and $0 \leq \gamma \leq 1$, mining a database D for association rules consists of extracting all pairs of *cass* s and u for which the relation $s \Rightarrow u$ is a $\sigma\gamma$ -valid association rule; i.e. $support(s \Rightarrow u) \geq \sigma$ and $confidence(s \Rightarrow u) \geq \gamma$. The problem can be solved with the two-step procedure below.

- 1) Find all σ -frequent *cass* v , that is to say all v such that $support(v) \geq \sigma$.
- 2) For all *cas* v found in step 1, generate all association rules $s \Rightarrow u$ such that
 1. $s < v$ and $u = v-s$ and
 2. $confidence(s \Rightarrow u) \geq \gamma$.

1.2 Related Works

A good number of serial and parallel algorithms have been developed for mining association rules for basket data (supermarket transaction data) [6-8]. Several researchers have applied data mining concepts on time series to find patterns including [1, 3, 5, 13, 16-19]. However, these algorithms are sequential in nature except [13, 16].

The problem of mining sequential patterns was introduced in [14]. Three algorithms have been proposed in this work for the purpose. However, in subsequent work [15], the same authors proposed *GSP* algorithm that outperformed *AprioriAll* by up to 20 times which is considered as the best in their earlier works. In [9], the algorithm *SPADE* was shown to outperform *GSP* by a factor of two by using

the advantage of lattice based approach. A parallel version of *SPADE*, was introduced in [10] for Shared Memory Multi-processor (SMP) systems. Sequence discovery can be essentially thought of as association discovery over a temporal database. While association rules discovery only applies to intra-relationship between patterns and sequence mining refers to inter-transactional patterns [14]. Due to this similarity, sequence mining algorithms like *AprioriAll*, and *GSP*, utilize some of the ideas initially proposed for the discovery of association rules [9]. Our algorithm proposed in section 3 is basically based on [10]. But we use prefix based classes rather than suffix based classes [10] and implemented the algorithm to discover the sequence of correlations over the multi-stream data. To the best of our knowledge there is no suitable algorithm proposed in the literatures for discovering sequential rules from multidimensional multi-stream time series data. Hence, our approach, discovering association rules from a large amount of time series data differs from the above algorithms in following ways:

- 1) We transform the large amount of multi-dimensional time series data into symbols of multi streams to make the data into lower dimensions. Because, it is very expensive, time consuming and computational intensive task to discover association rules from these kinds of huge amount of time series data represented as symbols of multi-streams.
- 2) We use lattice-theoretic based approach to decompose the original search space into smaller pieces i.e. into the prefix based classes which can be processed independently in main-memory of DSM multi-processors systems.

2 Transformation of Multi-dimensional Multi-Stream Data

The motion data captured by motion capturing system consists of various information of the human body parts. This means that, motion captured data can be represented by the 3 dimensional time series stream considering various positions of various body joints [4]. The following body parts data can be obtained as: lower torso, upper torso, the root of neck, head, the root of collar bones, shoulder, elbows, wrists, hips, knees and ankles. Moreover, body parts can be represented as the tree structure that is shown in Figure.1.

To get time series of 3-D motion data, we use an optical motion capture system. The system consists of six infrared cameras, 18 infrared ray reflectors, and a data processor. To get time series of motion data an actor puts 18 markers on the selected body joints and performs an action in the field surrounded by the installed cameras.

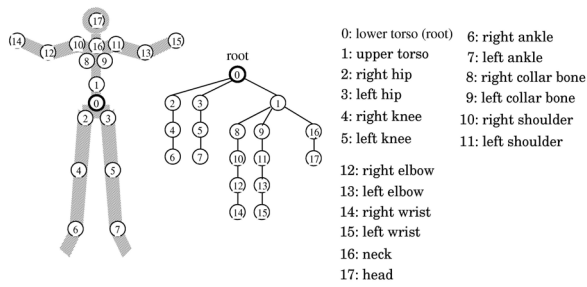


Fig. 1. Human body parts used for the experiments and considered as a tree structure

The recorded actions are processed in order to calculate the 3-D locations of the reflectors. Figure 2 shows an example of the motion that represents the correlation in “walking” such as “after one finished raising one’s right hand, one starts lowering the left hand”.

The example in Fig. 2 shows that the motion data has features as multi-stream, such as: “unfixed temporal interval between events on each stream: consistent contents on each stream do not always occur in fixed temporal interval. For instance, “raising the right hand” and “lowering the left hand” occur twice in each stream, however, temporal interval of occurrence between “raising the right” and “lowering the left hand” are different ($T1 \neq T2$).

In order to find motion association rules with easy analysis with considerations of various occurrences and reduce the cost of the task, we convert the high dimension multi-stream motion into sequence of symbols of lower dimension i.e. from 3 dimensions to 1 dimension. That is to say, the motion data can be converted into a small number of symbol sequences. Each symbol represents a basic content and motion data that can be expressed as the set of the primitive motions. We call this process content-based automatic symbolization [4]. For the content-based automatic symbolization, we focused on each change in the velocity of a body part that can be considered as a break point. We divide motion data into segments at those breakpoints where velocity changes. However the variation of the curve for motion data between these breakpoints are small and include noise occurred by unconscious movements, which are independent from occurrence of the changes of contents. The unconscious movements are mainly caused by the vibrations of the body parts. These are too short in time and tiny movements and contain no significant content of motion. Thus they are discarded by evaluating time scale and displacement of positions considering the 3D distances

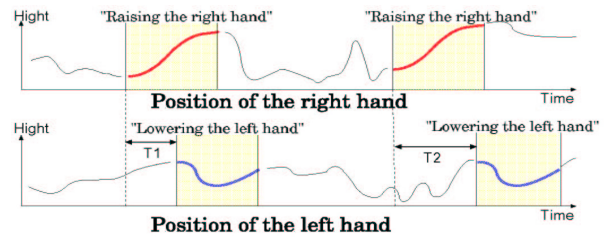


Fig. 2. An example of correlation of multi-stream of human motion

between points. Segmented data is clustered into groups according to their similarity. However, even segmented data with same content have different time lengths, because nobody can perform exactly in the same manner as past. In order to find out the similarity between time series data with different lengths, we employ Dynamic Time Warping (DTW) [1], which was developed in the speech recognition domain. By applying DTW on our motion data, the best correspondence between two time series was found. Human voice has fixed number of consistent contents, phonemes, but human motion does not have pre-defined patterns. Therefore, it is unknown that how many consistent contents exist in our motion time series data. For this reason a simple and powerful unsupervised clustering algorithm, Nearest Neighbor (NN) algorithm [2] is used with DTW to classify an unknown pattern (content) and to choose the class of the nearest cluster by measuring the distance.

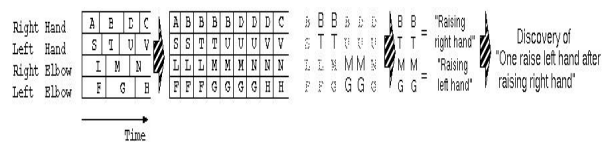


Fig. 3. Symbols of multi-stream

Thus, motion data is converted into symbol streams based on its content by using symbol that are given to clusters (Fig. 3). For more details of these processes please refer to [4]. After segmenting and clustering processes, a multi-stream that represents human motion is expressed as a multiple sequence of symbols that we call the sequence of symbols of multi-stream.

3 Mining Sequential Association Rules

Recently, lattice based sequence mining has been successfully employed by the researchers on supermarket

dataset [5]. The advantage lies in this approach is that it helps to decompose original search space into smaller pieces termed as the prefix based equivalence classes in our case, which can be processed independently in main-memory with the advantages of the DSM multi-processor systems. The decomposition is recursively applied within each parent class to produce even smaller classes in the next level. For example, in figure 4, the parent classes can be denoted as a, b, c, d, e and $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ represent the level of the search space.

3.1 Search Space Partition

As the first step to find the association rules, the step essentially consists of enumerating the frequent item sets. The database is very large, leading to a large search space. In order to lower the main memory requirements and perform enumeration in parallel, the search space is splitted into several parts that can be processed independently in parallel. This can be accomplished via prefix-based equivalence classes. It is possible that each class is a sub-lattice of the original sequences lattice and can be processed independently. For example, in the figure 4, it is shown that the effect of decomposing the frequent sequence lattice for the sample database, by collapsing all sequences with the same 1-length prefix into a single class. For details about lattice theory and structure please refer to [9].

The following way describes how the database can be partitioned. For example, given the number m in which search space is to be partitioned by satisfying the condition that k is the smallest integer such that $m \leq 2^k$. Here, $A = \{a, b, c, d, e\}$ and k are integers. For example, for $m = 4, k = 2$ which satisfies the above condition. For this value, it is possible to generate 4 sets of classes for splitting $(abc), (abd), (abe), (ac), (ad), (ae), (bc), (bd), (be), (c), (d), (e)$. The splitted parts are shown in the Fig. 4 by dotted lines. For $m = 8$, the value of k is 3, then the search space is splitted in 8 parts considering their corresponding prefixes. This is how the search space can be splitted by prefix based approach among the multiple processors.

3.2 Lattice Decomposition-Prefix Based Classes

In figure 4, there are five resulting prefix classes, namely, $[RightLeg], [LeftLeg], [RightArm], [LeftArm], [Trunk]$ in our case, which are referred to parent classes. It is to mention that for simplicity purpose to show the figure, the example using only 5 parts of the body is presented here. They are denoted in the figure as a, b, c, d and e

respectively. However, the figure for 17 parts also can be built in the same way. Each class is independent in the sense that it has complete information for generating all frequent sequences that share the same prefix.

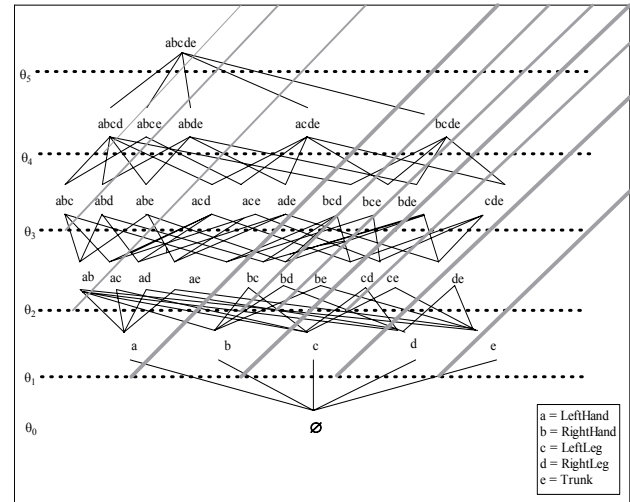


Fig. 4. Partitioning search space with the prefix based equivalence classes

For example, if a class $[a]$ has the elements $[b] \rightarrow [a]$, and $[c] \rightarrow [a]$. The only possible frequent sequences at the next step can be $[b] \rightarrow [c] \rightarrow [a]$ or $[c] \rightarrow [b] \rightarrow [a]$ and $[bc] \rightarrow [a]$. It should be obvious that no other parts such as y can lead to a frequent sequence with the prefix $[a]$, unless (ya) or $y \rightarrow x$ is also in $[x]$. The method decomposes the sequences at each new level into smaller independent classes. The figure shows the effect of using 2-level prefixes i.e. ab, ac, bd, bc etc. The figure also shows the 3-level, 4-level and 5-level prefixes. For all the levels, it can be obtained as a tree like structure of independent classes. This composition tree is to be processed in a breath-first manner, within each parent classes. In other words, parent classes are processed one-by-one, but within a parent class we process the new classes in a breath-first search. Frequent sequences are generated for the databases list by considering their support threshold min_sup . The sequences are being found to be frequent at the current level from the classes for the next New_Level . The level-wise process is repeated until all frequent sequences have been enumerated.

In terms of memory management, it is easy to see that we need memory to store intermediate lists for the most 5 consecutive levels within a parent class (Fig. 4). Once all the frequent sequences or the next level have been generated, the sequences at the current level can be deleted.

3.3 The Algorithm

As for the data set, each motion consists of repetition of 2 or 3 times for one kind. Test data are about 5 to 20 seconds long and the sampling frequency is 120 times/second. The motions are performed 23 times in 6 different types each of which lasts for about 6-12 sec. The database consists of 50 different types of motion such as walking, running, dancing, pitching etc. All of the experiments were performed on a 64-node SGI Origin 3400 DSM multi-processor system. The database is stored on an attached 6 GB local disk. The system runs IRIX 7.3.

There are two main paradigms that may be utilized in the implementation of parallel data mining: a data parallel approach and a task parallel approach. In data parallelism, P processors work on distinct portions of the database, but synchronously process the global computation tree. The parallelism is available within class. In task parallelism, the processors share the database, but work on different classes in parallel, asynchronously processing the computation tree. We need to keep the temporary list of all newly generated candidates both infrequent and frequent since we cannot say if a candidate is frequent until all processors have finished the current level. In the task parallelism all processors have access to one copy of the database, but they work on separate classes.

We use the task parallel approach to solve our problem. The pseudo code of the algorithm is given below. This is done within the level of the computation tree (see fig. 4). In other words, at each new level of the computation tree (within a parent class like a , b , c , etc.), each processor processes all the new classes at that level, performing intersections for each candidate, but only over its local block (steps 7-11 in algorithm *Mining_Sequential_Parts*). The local supports are sorted in a local array to prevent false sharing among processors (step 10 in procedure *FindNewClasses*). After barrier synchronization signals that all processors have finished processing the current level, a sub-reduction is performed in parallel to determine the global support of each candidate. The frequent sequences are then retained for the next level, and the same process is repeated for other levels until no more frequent sequence are found (steps 10-12 in algorithm *Mining_Sequential_Parts*). The level-wise task parallelism requires modifications by performing local intersection for all classes at the current level, followed by a barrier before the next level can begin.

In our case, with regards to 17 parts of the body, we used 17 processors to process the each prefix based class independently in the memory. It is to mention that as reported earlier, the fig.4 represents only prefix based search space with 5 parts. However, in case of 17 parts the

structure will be of similar type using 17 individual (parent) classes. After implementing the proposed algorithm for 17 body parts as 17 classes using 17 processors, we found many sequential associations rules for the body parts that took part in performing a motion efficiently. As an example of such discovered rules for performing "walking" is "raising the *RightHand* forward, lowering the *LeftHand* back, raising the *RightLeg* forward, directing the *LeftLeg* backward and move forward the *trunk*". We can find such kind of sequential rules for other sets of motion data like running, pitching, dancing etc. The running time presented in figure 6 indicated by "time using lattice based approach" justifies that good speed up is obtained by our approach.

Algorithm *Mining_Sequential_Parts*

Input: P , min_sup

Output: Plist

begin

1. $P = \{\text{Total body parts representing the parent classes } P_i, \text{ where } i=1,2,\dots,17\}$
2. for each $P_i \in P$
3. {
4. while (Previous_level $\neq \emptyset$);
5. do in parallel for all processors p
6. {
7. New_Level=New_Level \cup FindNewClasses (Previous_Level.bodyparts());
8. Previous_Level= Previous_Level.next()
9. }
10. barrier;
11. New_Level= $\cup_{p \in P} \text{New_Level}_p$
12. Repeat steps 5-7 if (New_Level $\neq \emptyset$)
13. }

end

Procedure *FindNewClasses*(P , min_sup)

begin

1. for all the sequences in $P_i \in P$
2. {
3. for all the sequences $P_j \in P$, with
4. $j \leq i$
5. {
6. $L = P_i \cup P_j$;
7. do in parallel for all processors
8. \mathcal{D}
9. $X = P_i \cap P_j$
10. }
11. if ($\sigma(L) \geq \text{min_sup}$) then
12. $P_i = P_i \cup \{L\}$;
13. }
14. return Plist=Plist $\cup P_i$

end

3.4 Comparison of the Results

To justify the effectiveness of our algorithm we compared with our proposed algorithm presented in [13, 16]. For ready reference here, we present a brief overview of our previous approach. In previous works [13, 16], we introduced two parallel algorithms for the same data sets that we used here for mining association rules based on well known *apriori* algorithm [6] in data mining for super market set data; one of them for mining association rules from 17 body parts and the other one from 5 parts of the body by reducing the search space due to the large number of combinations that used to occur during the search process and the complexity of search process itself. Note that the algorithm that used the 5 parts of the body regarded as the better solution between the proposed two algorithms [13, 16]. But it was identified as a limitation to our goal as our motivation was to discover rules using all of the parts of the body i.e. from 17 parts. So, as a new approach (in this paper), we propose the lattice based parallel algorithm for this purpose. Hence, the approach that we have used in this paper prevailed over with our previous problem. As described in the above section, we can discover rules from our datasets consisting of 17 parts and thus fulfill our objective using the algorithm presented here. So, we present the comparison between the results using our previous approach and the present approach, to figure out the efficiency of our present proposed algorithm. The results shown in Fig. 6 indicate that our present lattice based parallel algorithm can effectively reduce the time required for discovering rules in terms of scalability whereas the time taken by our previously reported algorithm does not show that kind of trend with the number of processors.

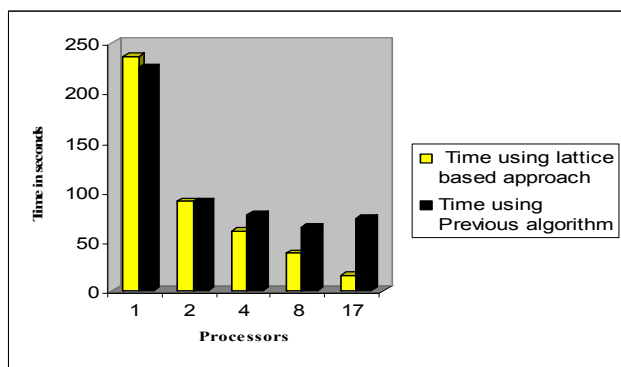


Fig. 6. The comparison results of the time required for discovering the rules between the two algorithms for the same data set

4 Conclusion and Future Work

In this paper, we have presented a parallel algorithm for finding association rules from sequences of the body parts that performs different kinds of motion from multi-stream time series data such as human motion data. The algorithm has considered a large number of combinations and the depth of the sequence of parts that perform motions (such as walking, dancing, pitching etc.). The extraction technique of motion data into symbols of multi-stream has also been discussed briefly. The experimental results have demonstrated that the algorithm can efficiently determine the rules of sequence of body parts that performs motion in our case by using the advantage of lattice based structure. It also has outperformed the result of our previously reported algorithm.

It is to mention that the structure of the algorithm and the way of implementation is not specific to the problem of searching the sequence of patterns (in our case body parts). It infers that our motivation was to reduce the time and find the sequence of the body parts using various combinations efficiently, and hence we achieved that for multidimensional data. This technique can be implemented with other data sets in the multidimensional multi-stream time series domain for finding interesting sequential rules for patterns in the domain of medicine and business. As a future work we aim to use the techniques for such data sets.

References

1. Berndt D.J. and Clifford J., Finding Patterns in Time Series: A Dynamic Programming Approach, in Proc. of Advances in Knowledge Discovery and Data Mining, pp. 229- 248, 1996.
2. Bay S.D., Combining Nearest Neighbor Classifiers Through Multiple Feature Subsets, in Proc. of 15th International Conference on Machine Learning, pp. 37-45, 1998.
3. Oates T. and Cohen P.R., Searching for Structure in Multiple Stream of Data, in Proc. of 13th International Conference on Machine Learning, pp. 346-354, 1996.
4. Shimada M. and Uehara K., Discovery of Correlation from Multi-stream of Human Motion, Lecture Notes in Artificial Intelligence, Vol. 1967, pp. 290-294, 2000.
5. Das G., Lin K., Mannila H., Renganathan G. and Smyth P., Rule Discovery from Time Series, in Proc. of Fourth International Conference on Knowledge Discovery and Data mining, AAAI Press, pp. 16-22, 1998.
6. Agrawal R. and Shafer J., Parallel Mining of Association Rules, IEEE Transactions on Knowledge and Data Engineering, Vol. 6, No.8, pp. 962-969, 1996.
7. Agrawal R. and Srikant R., Fast Algorithms for Mining Associations Rules, in Proc. of 20th VLDB Conference, pp. 487-499, 1994.

8. Park, J.S. Chen M.S. and Yu P.S., Efficient Parallel Data mining for Association Rules, in Proc. of CIKM, pp. 31-36, 1995.
9. Zaki M.J., Efficient Enumeration of Frequent Sequences, in Proc. of Intl. Conf. on Information and Knowledge Management, pp. 68-75, 1998.
10. Zaki M.J., Parallel Sequence Mining of Shared Memory Machine, in Proc. of Workshop on Large-Scale Parallel KDD Systems, SIGKDD, pp. 161-189, 1999.
11. Rosenstein, M.T. and Cohen, P.R., Continuous Categories for a Mobile Robot, in Proc. of 16th National Conference on Artificial Intelligence, pp. 634-640, 1999.
12. Jean-Marc Adamo: Data Mining for Association Rules and Sequential Patterns- Sequential and Parallel Algorithms, Springer-Verlag, Berlin Heidelberg New York (2001).
13. Sarker, B. K., Mori, T., Hirata, T. and Uehara, K. Parallel Algorithms for Mining Association Rules in Time Series Data, Lecture Notes in Computer Science, LNCS-2745, pp. 273-284, 2003.
14. Agrawal R. and Srikant R., Mining Sequential Patterns, in Proc. of 11th ICDE Conf., pp. 3-14, 1995.
15. Agrawal R. and Srikant R., Mining Sequential Patterns: Generalization and Performance Improvements, in Proc. of 5th Intl. Conf. on Extending Database Technology, pp. 3-14, 1996.
16. Sarker B.K., Hirata T., and Uehara K., Parallel Mining of Associations Rules in Time Series Multi-stream Data, Journal of Information, vol.9, no.2, pp.265-282.
17. Zhu Y. and Shasha D, StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time, Proc. of the 28th VLDB Conf., (2002), pp. 358-369.
18. Roddick J.F. and Spiliopoulou M., A Survey of Temporal Knowledge Discovery Paradigms and Methods, IEEE Transactions on Knowledge and Data Engineering, 14(4) (2002), pp.750-767.
19. Honda R. and Konishi O., Temporal Rule Discovery for Time Series Satellite Images and Integration with RDB, Proc. 5th European Conf. PKDD, (2001), pp. 204-215.

in Time Series Data, Buyers-Sellers Matching in E-Business and Semantic Knowledge Management in E-Learning. He is a member of IEEE and IEEE computer society.



Kuniaki Uehara received the B.Eng., M.Eng. and D.Eng. degrees in Information and computer sciences from Osaka University, Osaka, in 1978, 1980 and 1984, respectively. From 1984 to 1990, he was an Assistant Professor in the Institute of Scientific and Industrial Research, Osaka University. He was an Associate Professor in the Department of Computer and Systems Engineering, Kobe University from 1990 to 1997. From 1997 to 2002, he was a Professor with the Research Center for Urban Safety and Security of Kobe University. Currently he is a Professor with Graduate School of Science and Technology, Kobe University. He is also a Professor with the Department of Computer and Systems Engineering, Kobe University. From 1989 to 1990, he was a visiting assistant professor of Oregon State University, USA. From 1994 to 1996, he was held the position of Associate Director of Information Processing Center, Kobe University. His current research interests include machine learning, data mining, natural language processing, and multimedia processing. He is a member of the Information Processing Society of Japan, Japan Society for Software Science and Technology, and Japan Society of Artificial Intelligence.



Biplab Kumer Sarker received M.Sc. in Computer Engineering from Vinnitsa State Technical University, Ukraine (Ex.USSR) in 1995 and Ph.D in Computer Engineering from Institute of Technology, Banaras Hindu University, India in 2003. From December 2001, he was a Visiting Researcher in the Research center for Urban Safety and Security, Kobe University, Japan. He was held as Assistant Professor in the

Graduate School of Science and Technology, Kobe University, Japan from April, 2003 to March 2004. He joined KDE laboratory, Institute of Information Science and Electronics, University of Tsukuba, Japan from July, 2004. From February 2005, he worked at the faculty of Computer Science, University of New Brunswick, Canada as a Postdoctoral fellow. From April 2006, he joined Innovatia Inc. as a researcher under NSERC, Canada IRDF. His present research interest includes Task Allocation & Scheduling in Distributed Real Time System, Mining Association Rules