How Does Knowledge Discovery Cooperate with Active Database Techniques in Controlling Dynamic Environment? *

Hirohiko Kawano¹, Shojiro Nishio², Jiawei Han³ and Toshiharu Hasegawa¹

¹ Department of Applied Math. and Physics, Kyoto University, Kyoto 606, Japan ({kawano,hasegawa}@kuamp.kyoto-u.ac.jp)
² Department of Information Systems Engineering, Osaka University, Osaka 565, Japan (nishio@isc.osaka-u.ac.jp)
³ School of Computing Science, Simon Fraser University, Burnaby, BC, Canada V5A 1S6 (han@cs.sfu.ca)

Abstract. A dynamic environment, such as a production process, a communication network, highway traffic, etc., may contain a huge amount of information, changing with time, which is a valuable resource for understanding the general behavior of the environment, discovering the regularities and anomalies currently happening in the environment, controlling an evolution process, and intelligent modeling or managing the environment. Unfortunately, the data generated in a dynamic environment are often expressed in low level primitives and in large volumes. Because of the dynamic, continuous and rapid changes of the information flow, it is difficult to catch the regularities and anomalies in a dynamic environment and react promptly for real-time applications. In this study, a knowledge discovery technique is integrated with data sampling and active database techniques to discover interesting behaviors of a dynamic environment and react intelligently to the environment changes. The discovery of the dynamics in a computer communication network and the application of the discovered knowledge for network management are taken as an example in our study. The study shows (1) data sampling is necessary in the collection of information for regularity analysis and anomaly detection; (2) knowledge discovery is important for generalizing low level data to high-level information and detecting interesting patterns; (3) active database technology is essential for real-time reaction to the changes in a dynamic environment; and (4) an integration of the three technologies forms a powerful tool for control and management of large dynamic environments in many applications.

* The research of the first author was supported in part by a scholarship from the Ministry of Education, Science and Culture of Japan, that of the second author was supported in part by the Ministry of Education, Science and Culture of Japan under Scientific Research Grant-in-Aid, and that of the third author was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada and the Centre for Systems Science of Simon Fraser University. The work by the first author was done during his visit to Simon Fraser University in 1993.
1 Introduction

Natural or human-controlled processes generate huge amount of information in various kinds of large, dynamic environments. It is challenging but truly necessary to analyze the general behavior of the information flow in such dynamic environments in order to understand and/or control the environments in scientific, business and industry applications.

First, in such a dynamic environment, data are generated rapidly, continuously, dynamically and in huge volumes. It is often unrealistic to store a complete set of raw data in the limited amount of memory of a database system and dynamically analyze and manage the data. This forces people to dump the generated data to tapes without timely analysis. Huge data can be handled by a data sampling technique which samples interesting pieces of information dynamically and systematically[13].

Second, most of the data/information in a dynamic environment are presented at low, primitive levels. There may not exist clear and concise relationships or regularities expressible by low level primitives. The discovery of clear and concise relationships or regularities among the collected data, can be handled by a knowledge discovery technique which performs efficient and effective data generalization to discover useful knowledge or regularity from the collected information [11, 12].

Third, process control and system management in a dynamic environment often require prompt, real-time, and intelligent reactions in response to situation changes in the environment. These reactions can be dealt with by application of active database technology[11] for automatic and prompt reaction and control of the environment.

In this paper, a technique is developed for knowledge discovery in dynamic environment by extension of the attribute-oriented induction technique from relational databases[3]. In order to discover the dynamic status of those systems, the cluster of actual data with dynamics is collected during the observation time using data sampling technique, and knowledge rules regarding to the status of dynamic environment are derived effectively at real time. Those discovered rules are stored in the rule base. Our proposed technique also makes new rules from the rules in the rule base during several sampling periods. The derived rules can be stored in the active database. The condition evaluator evaluates the current condition, compares it with that of the stored rules, discovers irregularities of the current status, if there exist, and executes actions to control the system.

This paper is organized as follows. In Section 2, the ideas for knowledge discovery in dynamic environment are introduced, including a data sampling technique and the primitives for specification of generalization/learning tasks. In Section 3, an attribute-oriented induction algorithm with sampling is presented for learning several kinds of rules from a huge amount of data in a dynamic environment. The integration of knowledge discovery with active database technology is also discussed in this section. The application of knowledge discovery and active database techniques in network management is examined, and the study is summarized in Section 4.
2 Knowledge Discovery in Dynamic Environment

Generally, data for scientific discovery are objective and show the characteristics of natural phenomena; whereas discovery in business database reflect the real world and show the behaviors of human-artifacts. The latter one is well known as challenging and difficult problems. The volume of data in a dynamic environment could be created by artificial systems, which is similar to other business information systems. However, the huge volume of data are generated rapidly, continuously and possibly redundantly without a break in a dynamic environment. Such features are also similar to those in scientific discovery. Therefore, in order to develop the knowledge discovery in dynamic environment, it is necessary to fuse the two discovery domains. The technique of data sampling is based on the former one; whereas the technique of induction algorithm is based on the latter one.

To simplify our discussion, we consider several assumptions regarding to the data and learning tasks for knowledge discovery.

Assumption 1. A set of data for a learning task is collected by random sampling from a dynamic information system or a dynamic environment.

For the attribute-oriented induction algorithm, our previous study in [13] convinced us the accuracy of the discovered rules, including both characteristic rules and classification rules, using data sampling technique based on a statistical estimation theory. It shows the validity and credibility of knowledge discovery in database by the small number of sampled tuples. Therefore, meaningful rules satisfying the conditions can be derived within a short sampling period.

It is often desirable for large databases to have rules expressed at concept levels higher than the primitive ones. Therefore, we have,

Assumption 2. Generalized rules are expressed in terms of high level concepts.

Using a concept hierarchy, the rules learned can be represented in terms of generalized concepts and stated in a simple and explicit form, which is desirable to most users.

Assumption 3. Background knowledge is generally available for knowledge discovery process.

Following these assumptions, our mechanism for knowledge discovery in dynamic environment can be outlined as follows. First, a knowledge discovery process is initiated by a learning request, which is usually in relevance to only a subset of generating data. A data retrieval process is initiated to collect the set of relevant data using a data sampling technique. Second, generalization is performed on the set of retrieved data using the background knowledge and a set of generalization operators. Third, the generalized data is simplified and transformed into one of the following kinds of generalized rules, which may facilitate query answering and other applications.
1. **current status rules.** A *current status rule* summarizes the general characteristics of a set of sampled data at the present time which satisfies certain criteria, such as, the characteristics of the traffic flow on a network at the present time.

2. **stable rules.** A *stable rule* describes the general characteristics which remain stable over a period of time, such as, the rule that helps find out the heavy traffic on a network constantly or periodically.

3. **evolution rules.** An *evolution rule* describes the general characteristic of a set of patterns which evolve over several periods of sampling time, such as, how a network flow changes drastically over the past several sampling times.

3 **Attribute-Oriented Induction in Dynamic Environment**

3.1 **Attribute-oriented induction algorithm with data sampling**

A knowledge discovery process applies generalization operators to a large volume of sampling data and generates a set of generalized rules. In general, we have the following five basic techniques, *(1) Data focusing, (2) Attribute removal, (3) Attribute generalization, (4) Vote propagation, (5) Attribute generalization control*, for attribute-oriented induction [3].

Next, a stable rule shows the constant and periodical stable pattern of characteristic rules derived from different sampled data. The vote counts in characteristic rules can be used to find the stable pattern among a set of rules. Let \( r_k(t) \) be the \( k \)-th characteristic rules in the rule set derived from the sampling data at time \( t \), and \( v_k(t) \) be the vote count of \( r_k(t) \), normalized by the total number of sampling tuples. The numerical sequence of \( v_k(t) \) is examined in a way similar to a scientific discovery process. Since the volume of rules is also accumulated as the sampling period becomes large, it is usually assumed that the vote count is similar to the natural phenomena. Therefore, attribute-oriented induction with data sampling will be effective for rule acquisition in a dynamic environment.

Discovery of functional relationships in numerical data has been studied in the programs such as Bacon[8]. Generalization (or abstraction) is also an essential technique in such programs to grasp the knowledge about the status of a complex system. For numerical values, aggregation of variables is based on the eigenvalues of the system matrix [5]. On the other hand, in our proposed induction algorithm, aggregation of variables is done by climbing up the conceptual hierarchy. The vote counts in the derived rule describe the characteristics of a dynamic system. Based on such values, one can derive various functional relationships depending on the abstraction level of hierarchical concepts.

To derive the functions which describe the stability or variance of a dynamic system, it is important to examine the set \( \mathcal{V}_k = \{ v_k(t_i) | t_i = t + L \cdot i, (i \geq 0) \} \), where \( v_k \) is the vote count regarding to the \( k \)-th rule, \( t_i \) is the starting time of the \( i \)-th sampling operation, and \( L \) is the sampling period.

Generally, scientific discovery applies heuristic search in an infinite space of potential relations or, equivalently, the combination of possible functional
forms, in order to find an appropriate functional property of \( \mathcal{V}_h \). This strategy of infinite searching causes difficulties for function finding in the real time. However, in the most cases of finding characteristics of information flow in a dynamic environment, search is limited to several typical functions. In order to keep the system stable, the system with the characteristics of the stable functions in the long range of time must be constructed. Therefore, constant values or periodical functions play an important role in those stable systems. Such periodic properties will be stored as stable rules in a rule base. Moreover, if the same rules are discovered during different periods, the weight of those stable rules will increase. On the other hand, if anomalous values are observed in short/long periods, evolution rules in short/long periods can be extracted based on quantitative measurements of such anomaly.

Further, several vote counts can be examined in order to derive the complex stable/evolution rules in short/long periods. A sequence of vote counts of several rules may satisfy the following equation, \( y = f(v_1(t), v_2(t), \ldots, v_i(t), \ldots, v_M(t)) \), the way to derive rules is similar to the above method. However, the search space will become huge depending on the abstraction level when the number of rules increases. Such mathematical functions can be considered as another kind of generalization operators.

Therefore, we have two techniques. (5) Stability Criterion: , if the sequence of vote counts \( v_k(t) \) satisfy the stable condition, the rule \( r_k(t) \) should be stored into the set of stable rules. And (6) Variant Criterion: , if the sequence of vote counts \( v_k(t) \) satisfy the evolutorial condition, the rule \( r_k(t) \) should be stored into the set of evolution rules.

The six basic learning strategies can be summarized into the following generalization algorithm which extracts generalized characteristic, stable and evolution rules from a large volume of data using sampling technique. The algorithm is an extension of the basic attribute-oriented induction algorithm [3, 4, 10] for learning rules in dynamic environment.

Algorithm: Attribute-oriented induction with random sampling in a dynamic environment

*Discovery of a set of generalized characteristic, stable and evolution rules in a dynamic environment based on a user’s learning request.*

**Input:** (i) A large volume of data in dynamic environment \( D \), (ii) a set of concept hierarchies or generalization operators on attributes \( a_i \), and (iii) \( T \), a relation threshold, and \( T_i \), a set of attribute thresholds for attributes \( a_i \).

**Output:** A characteristic rule, stable rule and evolution rule based on the learning request.

**Method:** Attribute-oriented induction with random sampled data with density \( d \) from the time \( t \) to \( (t + \ell) \) consists of the following steps:

1. Step 1. Collect a set of task-relevant sampling data \( T \) into an initial relation \( r_0 \) (possibly by a relational query).
2. Step 2. Perform basic attribute-oriented induction on \( r_0 \) as shown below.

   ```
   begin
   ```
for each attribute $a_i$ ($1 \leq i \leq n$) in $r_0$ do % number of attributes
if $a_i$ has not reached the desired concept level $T_i$ then % attribute
begin
if $a_i$ cannot be further generalized
then remove $a_i$ % attribute removal
else generalize $a_i$ to the (minimal) desired level; % concept tree
ascension
merge identical tuples % vote propagation
end
end. (Basic attribute-oriented induction)

Output characteristic rules and store them into the rule base.

Step 3. Based on the stability criterion, derive the stable rules by the evaluation of stored $v_k$ regarding to characteristic rules.

Step 4. Based on the variant criterion, derive the evolution rules by the evaluation of stored $v_k$ regarding to characteristic rules.

Step 5. Repeat the induction algorithm periodically based on the new set of sampling data in the new period.

3.2 Intelligent reactions to dynamic environments

Active database [1, 2, 9, 14] is one of the interesting subfields in database research. Since a dynamic environment requires prompt, real-time reaction to the changes of the environment, it is important to explore the integration of active database technology with machine learning techniques.

Fig. 1 illustrates an architecture of active database in a dynamic environment, which is conceptually divided into rule base, knowledge base and conditional evaluator. Our proposed algorithm derives characteristic rules, stable rules and evolution rules. Those derived rules should be stored in the active database selectively and be applied in the control of the environment.

To react intelligently to dynamic environments, the active database techniques and knowledge discovery processes can be integrated in the following five aspects, regularity extraction, regularity updates, knowledge-assisted active rule specification, dedicated knowledge discovery, generalized triggering. These aspects are further analyzed below.

First, regularity extraction needs the integration of both techniques. A large number of rules which summarize the current status or the stable and evolving regularities of a system can be extracted by a knowledge discovery process. However, some of these discovered regularities could be less interesting or redundant to the system. An active database method may act as knowledge discovery initiator which triggers a knowledge discovery process based on the importance or freshness of the knowledge to the system. The importance is related to some critical or sensitive aspects of an environment, such as the potential crisis of a production process, the critical condition of a chemical reaction, etc.; whereas the freshness is related to whether similar knowledge is already in the system.
Fig. 1. Architecture of active knowledge database in dynamic environment

Furthermore, if the dynamic data is too large to be stored in a database but constantly monitoring is more preferable than data sampling, the database may store data summaries (extracted by knowledge discovery) at a level slightly higher (thus less voluminous) than the primitive data.

Second, it is important to verify, modify or invalidate the existing generalized rules stored in the rule base in a dynamic environment. Such a regularity updating task can also be performed by integration of knowledge discovery and active database technology. When a rule is discovered by a knowledge discovery process, the rule could be in one of the following cases: (1) it may enrich an existing rule by consolidating it in an extended period, or extending its condition or conclusion, (2) it may invalidate an existing rule because of the changed condition or conclusion, (3) it may not be interesting or fresh enough for inclusion in the rule base, or (4) it may violate certain integrity constraints, thus need to invoke some warning messages or perform appropriate actions. Knowledge rule verification, modification, invalidation or other appropriate actions can be specified as actions of triggers in an active database, which can be invoked when certain conditions are detected in the knowledge discovery process.

Third, the integration of the both techniques may facilitate knowledge-assisted active rule specification. It is often necessary to express active rules at the concept level higher than the primitive data in dynamic environments for comprehension and debugging by human programmers/operators. Such specification needs the help of knowledge discovery process. Moreover, the specification of appropriate conditions and actions should be based on the analysis of general characteristics in the current system, the stable and evolving regularities, and the execution history of the active rules. Obviously, such specification, refinement, and assessment of the appropriate conditions and actions of active rules need the application of knowledge discovery tools.

Fourth, dedicated knowledge discovery need the use of both knowledge discov-
ery and active database mechanisms. In general, data sampling and knowledge discovery can be classified into general and dedicated processes. The former is adopted for regular environment checking and knowledge discovery; whereas the latter need to be invoked for a dedicated, detailed, frequent, and specialized data sampling knowledge discovery when certain condition happens. For example, when a production environment reaches a critical condition, an emergency data collection and knowledge discovery process should be invoked for close observation. Such an invocation of a dedicated sampling and discovery process can be performed by specifying conditions and actions for a dedicated process using the active database technology. When some unusual situation was detected by a knowledge discovery process, more focused and refined knowledge discovery can be initiated, and such a process can be refined progressively based on the discovered results. Such progressively refined processes can be specified by active database rules.

Finally, the integration may facilitate generalized triggering. Since the general status and evolution regularities of a system can be discovered by a knowledge discovery process and summarized at a high concept level, the conditions and appropriate actions of an active rule can be specified at a high level to communicate with both human and the system. Moreover, the actions of an active rule can also be specified at a high concept level. Appropriate mappings can be performed to transform high level actions into low level primitives to trigger the detailed actions and update the environment.

3.3 Management of Communication Networks by KDD

As an example to our study, the management of interconnecting communication network using data sampling[6], knowledge discovery[3], and active database techniques[14] is analyzed. The status of the network changes over time, and the difficulty of effective and stable operating of the complex system is evident. In order to operate the communication network, it is essential to find the real time characteristics for load balancing or the connecting status between several network resources.

Example. Let the learning task be to discover a stable rule in relevance to Source, Destination, Size, Type and Port. The learning task can be represented in a pseudo query language in Fig. 2.

LEARN Stable rule
FROM Sampling period from 1 to 100
WHERE P.SrcAddress in Domain(‘kuamp.kyoto-u.ac.jp’) and
P.DestAddress not in Domain(‘kuamp.kyoto-u.ac.jp’) and
P.Size <= 300
IN RELEVANCE TO P.SrcAddress, P.DestAddress, P.Size, P.Type, P.Port

Fig. 2. Query for stable rule

Active database techniques can be applied to the system management based on the rules discovered in the above example. Suppose that the following def-
inition of actions by pseudo description language, as shown in Fig. 3, is in the active database. The rule states that if the periodical length between different domains is less than 5 minutes, the system displays an alert message on the console, initiates the termination of connection-less packets, and checks the processes in the both domains. The descriptions of actions for evolution rules can be defined in a similar way.

Event: Update Stable Rules in Rule Base
Condition: Period(P.Vote) < 5
Query:
SELECT P.Vote
FROM P Stable Rules
WHERE P.SrcAddress != P.DestAddress and P.Type=CU
Action:
Operation: begin
  DisplayConsole('code red', Stable Rules P)
  Activate(Kill_CL in Code_Red_Rules)
  Signal(Check_Processes_in (P.SrcAddress, P.DestAddress)
end

Fig. 3. Pseudo definition in active database

This example shows that by integration of a knowledge discovery method with an active database technology, the interconnecting network can be managed intelligently and dynamically at a high level, which may effectively control the information network across remote distance.

4 Conclusion

In this paper, we studied knowledge discovery in dynamic environments. First, a cluster of data is collected by a data sampling technique. An attribute-oriented induction technique is then applied which integrates the learning-from-examples methodology with set-oriented database operations and extracts generalized data from actual data in databases. Attribute-oriented induction and data sampling substantially reduce the computational complexity of a database learning process. The attribute-oriented induction algorithm with data sampling discovers three kinds of rules: characteristic rule, stable rule and evolution rule. Moreover, using the technology of active database, generalized conditions can be evaluated and compared with the generalized rules for the control of the dynamic environment. Our study shows that the integration of attribute-oriented induction algorithm with data sampling technique and active database technology will substantially enhance the power and increase the flexibility of data and knowledge discovery and utilization in dynamic environments.

There are many issues which should be studied further. Both data sampling and knowledge discovery processes can be triggered by active database rules. It is not clear how close interactions should be maintained between active database rules and knowledge discovery algorithms with data sampling. A tight control by
active rules may restrict the possibilities of discovery of some unexpected events; however, a loose interaction may result in the discovery of a large number of uninteresting rules. The balance of different mechanisms needs much study and experiment work. We plan to examine the effectiveness of the techniques developed in this paper and implement such mechanisms in a dynamic production control environment. The experimental results will be reported in the near future.

References

3. J. Han, Y. Cai, and N. Cercone, “Knowledge Discovery in Databases: An Attribute-Oriented Approach.” In Proc. 18th Int’l Conf. on Very Large Data Bases, Vancouver, Canada, pp.547-559, August 1992.

This article was processed using the LATEX macro package with LLNCS style