Discovery and Application of Inter-Class Patterns in Database

Dong-Ha Lee, Dong-Yal Seo, Kang-Sik Moon, Jisook Chang, Do-Won Nam, Jeon-Young Lee
Intelligent Information Systems Laboratory
POSTECH Information Research Laboratories(PIRL)
Pohang University of Science and Technology
Nam-Gu Pohang Kyungbuk, 790-784, Korea
{dongha, dyseo, ksmoon, jihan, irene, jeon}@white.postech.ac.kr

Abstract

This paper presents an inter-class pattern discovery method in real world database. While data in conventional database has tuple structure, the data in pattern discovery has set-values or sequences. The structural difference between them may cause useless resulting patterns and may result in inefficient pattern discovery method.

To resolve those issues, we propose an inter-class pattern discovery methodology. The first step is to convert conventional database to set of objects. During the conversion process, a tuple in the original database is converted to a conceptual object and as another result, object generalization hierarchies are generated. From the object generalization hierarchies, interesting patterns of the conceptual objects can be extracted by applying multi-level pattern discovery algorithms. The resulting patterns are inter-class patterns of original conventional database.

We also show a pattern discovery query for our methodology and its application on intelligent query processing.

1. Introduction

Knowledge hidden in database can be extracted by KDD(knowledge discovery in database) techniques. The hidden knowledge, for example, 'students from Seoul and more than 30-year old are tend to get good grade in courses of computer department' can not be extracted by conventional database interface languages or using statistical tools. Moreover, the extraction processing should be tractable considering the amount of data in practical database.

Knowledge discovery systems or pattern discovery systems are motivated by the above problem. They extract previously unknown and possibly useful knowledge that helps decision-making. Two of the most useful patterns among knowledge discovery techniques are association rules and sequential patterns. Association rules are extracted from bucket database that is composed of set of buckets; each bucket has a set of items. An association rule finds out dependency relationships among items. Some items tend to appear at the same time with certain items. On the other hand, sequential patterns are defined by the sequence database that is composed of set of sequences. Each sequence is a list of items. The sequential patterns show temporal inter-relationship among item sets.

These kinds of knowledge from databases are very useful and have been applied to practical applications to get meaningful results [2][3]. The inputs of these algorithms are somewhat different form from the data stored in conventional database. While objects in conventional database have tuple structure, the target objects in pattern discovery are set values or sequences. The different format between data in conventional database and inputs for pattern discovery makes it irrelevant to apply the latter for exploiting the former.

To apply these knowledge discovery techniques for real-world database, we develop a well-defined mechanism to convert conventional databases into a set of items or sequences. During the transformation, a tuple in the original database is converted to a conceptual object and object generalization hierarchies are built. Using the object generalization hierarchies, interesting patterns of the conceptual objects can be extracted by applying previously developed multi-level pattern discovery algorithms

As a useful application of inter-class pattern discovery, we show intelligent query answering system in which user information and interaction patterns have the form of relational tables.

The rest of this paper is organized as follows. Previous researches on association rules and sequential patterns are introduced in section 2. Section 3 gives inter-class pattern discovery methods. 3 steps of inter-class discovery methods are object abstraction from relational table and target transaction generation and multi-level pattern

0-8186-8147-0/97 $10.00

discovery. Applications of inter-class pattern discovery are given in section 4. The examples of pattern discovery query and intelligent query processing are given. Section 5 is a conclusion.

2. Previous researches on association rules and sequential patterns

Association rule was introduced by [2]. It has the form of rules among items in retail database. An association rule of {a, b}→ {c} means that an item set {a, b} tends to be appeared with an item set {c}. Two important parameters for finding association rule are minimum support (minsup) and minimum confidence (minconf). Association rules among item set in database are found with given parameters of minimum support and minimum confidence.

Sequential pattern was introduced by [3]. Transactions in the database have time stamps. Sequential patterns are patterns of time sequences. An example of the sequential pattern ({a, b}, {c}) means when item set {a, b} tends to be followed by item set {c}. In this pattern, minimum support parameter should be given.

Both pattern discoveries need multiple scans of database. A big effort has been given to reduce the number of database scans or main memory requirement [10][4]. Other researches focused on expanding expressiveness of resulting rules with multi-level patterns.

{Korea, Foreign} ⊂ Any
{Seoul, Not_Seoul} ⊂ Korea
{Kyungbuk, Kyungnam, Kangwon,...} ⊂ Not_Seoul
{Chunchon, Wonju, ... } ⊂ Kangwon

Figure 1. Concept Hierarchy on values of birth_place

Recent database extensions introduce concept hierarchy on attributes. Figure 1 shows concept hierarchy for values of birth_place[1] attribute. Using given concept hierarchies on the database, generalized association rules or generalized sequential patterns can be extracted [12][7]. Generalized patterns give more abstracted information. For example, a generalized association rule of birth_place attributes is 'Foreign students tend to get good grade on English.'

Though useful results come out using association rule and sequential pattern finding in practical applications, there is still major obstacles using these mining techniques on real world conventional database. Because target objects in pattern discovery need to be item sets or lists of item sets but the data objects in real world database has tuple structure. Thus a database tuple should be merged into large item set to be used as an input for pattern discovery. Even after changing into large item set, there could be come out irrelevant result with pattern discovery on several tables or trivial association rules of dependant items.

To use these pattern discovery techniques on real world conventional database, we develop inter-class pattern discovery method. We describe it in next section.

Table 1. Tuples in table *student*

sid	name	age	birth_place
9325d07	Dong-Ha Lee	30	Seoul
9425d05	Dong-Yal Seo	29	Chunchon
9625m20	Do-Won Nam	25	Taegu
...	...	...	...

3. Inter-class association rules discovery

We show inter-class pattern discovery method in this section focused on discovery patterns on association rules. The results may be easily expanded to sequential patterns.

We describe the ICAR(Inter-Class Association Rules) algorithm, which is used to extract association rules among several tables (or classes in OODB). The 3 steps of this algorithm are tuple abstraction, target transaction composition, and multi-level association rule finding. First, tuples in relational database are abstracted as objects. The original target tables are called base tables. During the abstraction, object generalization hierarchies are generated. Second, target transaction databases are built by joining the resulting objects from the first step. Each target transaction has same number of attributes and each item represents a tuple in the base table. Finally, multi-level association rule finding algorithm is executed on the target transactions using the object hierarchy that was generated by step 1.

Table 2. Generalized tuples from table *student*

Age	birth_place
20>=	Seoul
20<, 25>=	Seoul
...	...
25<, 30>=	Not_Seoul
30<	Not_Seoul

3. 1. Tuple abstraction

[1] The attribute birth_place is defined in the relational schema of Figure 3.

This step abstracts tuples in base table (given table) into objects and build generalization hierarchy that conceives the resulting objects.

One of the most well known knowledge discovery techniques is attribute-oriented induction [5]. This technique results in generalized table from base table using given concept hierarchies. The processes of attribute-oriented induction are given in table 1, table 2, and table 3. The concept hierarchy of each attributes in student table, we can obtain the result of table 2(Note that we must choose target attributes before this abstraction step. In general, it is common that interesting attributes are predefined by administrator). More generalization can be applied to the table 2, and it generates table 3.

Table 3. Generalized table from table 2

Age	birth_place
30>=	Seoul
30 <	ANY
30>=	Not_Seoul

After getting the generalized relation, a tuple in student table can be viewed as being located in an abstraction hierarchy. This generalization information can be kept as hierarchy tree, which is similar to conceptual hierarchy defined on an attribute. Figure 2 gives the object generalization hierarchy in which tuples in base table are abstracted as objects and the hierarchy is used for next step. We used attribute-oriented induction to build the object generalization hierarchy in this paper but other kinds of hierarchical tuple clustering methods can be used. As the first step of inter-class association rule discovery or inter-class sequential pattern discovery, the table abstraction step results in the object generalization hierarchy.

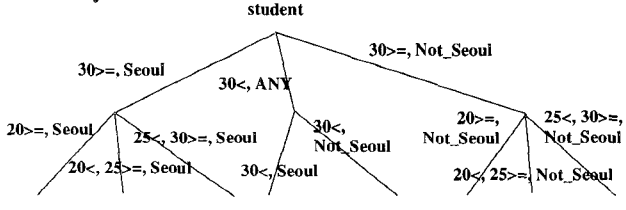

Figure 2. Object generalization hierarchy on table *student*

A tuple in student table can be generalized on the object hierarchy in Figure 2. When ***sid*** is tuple identifier of the tuple, a student ***9325d07*** can be generalized as a class that is represented as (25<, 30>=, Seoul) and the student can be classified to as more generalized class of (30>=, Seoul).

3. 2. Target transaction composition

Target transaction database is constructed by joining the participating base tables. Pattern discovery algorithm is performed on a set of items in target transaction database. A target transaction is an object that represents a tuple in a base table.

student(sid, name, status, major, gpa, birth_date, birth_place, address)
course(cid, name, dept)
register(rid, sno, cno, grade)

Figure 3. Example schema

For example, a relational schema is given in Figure 3. Target transaction is compose of 3 items, ***sid, cid,*** and ***rid*** which represent each base tables.

After defining target transaction scheme, the target transaction database should be built by joining the base tables. The resulting table has not all attributes but only tuple identifiers. The number of target transactions should be same as the number of tuples in the result table.

After building the target transaction database, multi-level association rule finding algorithm [7][12] should be executed. The resulting association rules are inter-class association rules.

Algorithm ICAR
Input: Tables T, Interesting attributes IA
Output : Set of Association Rule

filter out non-interesting attributes from T if it is neither primary or foreign key ;
foreach table t in T do
generalize table t using attribute-oriented induction and remember the generalization path ;
build target transaction schema collecting the representative attributes ;
build target transaction database by join the encapsulated objects
multi-level association rule mining;
convert the result to inter-class association rules

Figure 4. Algorithm ICAR

3. 3. Representative attributes

An attribute selected from base table is referred as representative attribute that is an abstraction of the tuple. In the previous section we used tuple identifiers as the representative attribute. The representative attribute should be chosen carefully. These considerations make more efficient the result for the inter-class pattern discovery.

For the base table has no interesting attributes that are not foreign keys, the table has no representative attribute. This table is used with a connection of other tables. No meaningful association rule can be extracted. On such tables, generalization step can be omitted.

Second, when base table has only one interesting attribute that is not foreign key, the only interesting attribute becomes the representative attribute of the table. The attribute is the only attribute in the table that can be contributed for inter-class association rules as a meaningful concept. In this case, the generalization step is also omitted. The concept hierarchy defined on the attributes plays the role of object generalization hierarchy of the table.

Finally, when the only one interesting attribute is set type, all the values of the attributes should be appeared in target transaction. The resulting target transactions will have different sizes. Recently published multi-level pattern discovery techniques give efficient results for set data type. In this case, object generalization step omitted also.

For previous example, the target transaction schema of the example relational database in Figure 3 is (***sid, dept, grade***). The ***dept*** attribute came from ***course*** table.

3. 4. Performance gain

Let the tuple numbers of n tables be $t_1, t_2, ..., t_n$ and attribute numbers $a_1, a_2, .., a_n$. Let the number of bytes for attributes be same.

In conventional method, we need to join all the participating base tables and that requires space of $O(\Sigma a_k \Pi t_k)$. And we need time for finding association rules(or sequential rules) from the transaction database of attribute size Σa_k and the size of transaction of $O(\Pi t_k)$.

But the proposed techniques requires space of $O(n \Pi t_k)$ and time for association rule(or sequential rule) finding from the transaction database of attributes size n and the size of transaction of $O(\Pi t_k)$ and the time for table generalization which is $O(\max a_i t_i)$ and multi-level association rule(or sequential rule) finding in target transaction database. The time requires to generalize the base table is linear to the size of base table. Both cases require additional time for table joins.

Time requirements both pattern discoveries are not yet analyzed exactly. Considering that time requirement of pattern discovery is higher than linear, linearity of time requirement of object generalization step promises that the performance of proposed method is better.

4. Applications

In this section, we present a pattern discovery language as an extension of SQL and give intelligent query answering system that is an application for inter-class pattern discovery and the pattern discovery language. Inter-class pattern discovery enables variety of application of pattern discovery.

Query pattern: (ts, u_id, {at_id}, {cond}, {at_id_pair})
Answer pattern: (ts, u_id, at_id, {val})
ts: time stamp
u_id: user identifier
at_id: selected attribute id
cond: constant condition
at_id_pair: join condition
val: retrieved values

Figure 5. Patterns query

```
discover        RULE_TYPE
from            TABLE_LIST
[where          CONDITION_LIST]
[in relevance to
                    INTERESTING_ATTRIBUT
                E_LIST]
[with           DISCOVERY_PARAMETER_LIST]
[during         DISCOVERY_OPTION]

discover        association rules
from            student s, course c, register r
where           s.birth_place ="Seoul" and s.sid =
                    r.sid and c.cid = r.cid
in relevance to s.status, s.major, s.address,
                s.birth_date, c.dept
with            support=0.01, confidence=0.4
during          every winter

discover        sequential patterns
from            student s, course c, register r
with            confidence=0.4, window size = 5,
                    constraint 4
during          every winter
```

Figure 6. Pattern discovery query syntax and examples

4. 1. Pattern discovery query language

The development of query language for knowledge discovery is one of hot issues in data mining research area. We design a pattern discovery language that has similar syntax with SQL. To get specific inter-class patterns, target base table and discovery parameters should be expressed using pattern discovery query.

The designed pattern discovery query language is given in Figure 5. Inter-class pattern discovery enables pattern discovery from multiple tables and the pattern discovery query can be expressed as structured form shown in the examples. We are considering other discovery options. One of the discovery options is temporal constraint such as 'during winter.' By considering specific time period, more interesting patterns can be found which are ignored during the process of global pattern discovery.

The results of pattern discovery query can be expressed as a set of rules in association rules, or a set of lists of items in sequential patterns.

4. 2. Intelligent query answering

Recently, a research on intelligent query answering using knowledge discovery techniques was introduced [8]. Intelligent query answering is an attempt to give more relevant information for user query. Related studies are cooperative query answering [6] and intensional answering [11]. Techniques used in those researches can be classified as generalization, summarization, query rewriting using associated or neighborhood information, comparison of answers with those of similar queries, and intensional answers or explanation of answers [9].

Han and others [8] focus on their knowledge-rich data model and depends generalization technique among various knowledge discovery techniques.

We suggest that more cooperative behavior of database can be served by analyzing user access pattern. User access pattern can be expressed in list of items. Therefore, recent research on association rule and sequential pattern discovery enables efficient and meaningful user analysis.

Observing user interactions between database system and user query, we model the interaction as query step and answer step. At each step, target pattern can be generated. We give possible patterns at Figure 6. We refer the pattern comes from query step as query pattern and the other as answer pattern.

A query pattern is a tuple of an extended relational model with set item extension. Query patterns will be abstracted as objects and object generalization hierarchy will be generated using tuple abstraction method of section 3-1.

In this intelligent query answering system, it is assumed that user information is given as a table, which named ***user*** table. The ***user*** table has relational information such as affiliation, age, department, and classified group. During preprocessing, the ***user*** table is abstracted using tuple abstraction method described in section 3.1. As the result of tuple abstraction, object generalization hierarchy for ***user*** table is generated. Using the two object generalization hierarchies, we can give pattern discovery query, such as in Figure 7. The result of the pattern discovery query in Figure 7 can be used in intelligent query answering technique of neighborhood query.

```
discover    association rules
from        query q, user u
where       u.birth_place ="Seoul" and u.u_id =
              q.u_id and ts in [97/01/01,
              97/03/01]
with        support=0.01, confidence=0.4
```

Figure 7. A pattern discovery query from query pattern

4. 3. More intelligent query answering

More intelligent query answering is possible. Pattern discovery from query pattern can show the tendency of user group. The group tendency information can be used in query rewriting to give interesting neighborhood query. In other case, comparative group tendency can be used for comparison query. For example, given graduate student's query, it is more informative to give not only the answer of the query but also the answer of undergraduate student's frequent query.

Query pattern and answer pattern also can be extended. Possible extension of pattern is considering positive and negative answer [13]. Query result is a table and that is positive answer. Negative answer is obtained when all table participating the query are joined and projected as query result and excluded the positive answer. Patterns from negative answers represent the tendency of data which have been excluded from retrieval.

5. Conclusion

We have presented an inter-relation pattern discovery method that has improved efficiency and relevance. This inter-class pattern discovery method makes it possible to expand the application areas of knowledge discovery. As we know, no research on association rule or sequential pattern did work on real-world database. Database should be preprocessed for those pattern discovery.

The proposed method has been illustrated by the application to the design of an intelligent query answering system. To help the application, a structured query has been designed. Association rules and sequential patterns discovered by the proposed inter-class pattern discovery method from the database have shown the significantly improved the intelligence of the system.

This preliminary work is being extended in several directions. We are extending the discovery pattern to be found in real-world database. Object generalization hierarchy extraction process can be optimized (in a certain point of view) or can be effected by user feedback. Also, we want to find more 'large' grain pattern from database or user-database interactions.

6. References

[1] R. Agrawal, S. Ghosh, T. Imielinski, B. Iyer and A. Swami, "An Interval Classifier for Database Mining Applications," Proc. 18th Int. Conf. VLDB pages 560-573, 1992.

[2] R. Agrawal T. Imielinski and A. Swami, "Mining association rules between sets of items in large database," Proc. 1993 Int. Conf. ACM SIGMOD, pages 207-216, May 1993.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," Proc. 11th Int. Conf. Data Engineering, pages 3-14, 1995.

[4] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," in U. M. Fayyad et. al. eds, Advances in Knowledge Discovery and Data Mining, Menlo Park, CA, AAAI/MIT Press, pages 307-328, 1996.

[5] Y. Cai, N. Cercone, and J. Han, "Attribute-Oriented Induction in Relational Databases," in G. Piatetsky-Shapiro and W. Frawley, Editors, Knowledge Discovery in Databases, Menlo Park, CA, AAAI/MIT Press, pages 213-228, 1991.

[6] F. Cuppens and R. Demolombe, "Cooperative Answering: A Methodology to Provide Intelligent Access to Databases," Proc. 2nd Int. Conf. Expert Database Systems, pages 621-643, Apr. 1988.

[7] J. Han and Y. Fu, "Discovery of Multiple-Level Association Rules from Large Databases," Proc. 21th Int. Conf. VLDB, pages 420-431, 1995.

[8] J. Han, Y. Huang, N. Cercone, and Y. Fu, "Intelligent Query Answering by Knowledge Discovery Techniques," IEEE Trans. Knowledge and Data Engineering, Vol. 8. No. 3, pages 373-390, Jun. 1996.

[9] T. Gaasterland, P. Godfrey and J. Minker, "An Overview of Cooperative Answering," J. Intelligent Information Systems, Kluwer Academic Pub. Vol. 1, No. 2, pages 123-157, 1992.

[10] A. Savasere, E. Omiecinski and S. Navathe, "An Efficient Algorithm for Mining Association Rules in Large Databases," Proc. 21th Int. Conf. VLDB, pages 432-444, 1995.

[11] C.-D. Shum and R. Muntz, "Implicit Representation for Extensional Answers," Pro. 2nd Int. Conf. Expert Database Systems, pages 497-522, Apr. 1988.

[12] R. Srikant and R. Agrawal, "Mining Generalized Association Rules," Proc. 21th Int. Conf. VLDB, pages 407-419, 1995.

[13] J. P. Yoon and L. Kerschberg, "A Framework for Knowledge Discovery and Evolution in Databases," IEEE Trans. Knowledge and Data Engineering, Vol. 5. No. 6, pages 973-979, Dec. 1993.