Attribute-Oriented Knowledge Discovery in Rough Relational Databases

Theresa Beaubouef
Southeastern Louisiana University
Computer Science Department, SLU 10847
Hammond, LA 70402 USA
Email: tbeaubouef@selu.edu

Frederick E. Petry
Naval Research Lab
Stennis Space Center MS USA
Email: fpetry@nrlssc.navy.mil

Abstract

The rough relational database model was developed for the management of uncertainty in relational databases. A particular type of knowledge discovery, attribute oriented induction of rules from generalized data is described in this paper.

Rough Relational Databases

In rough sets (Pawlak 1984) an approximation space is defined on some universe U by defining some equivalence relation that partitions the universe into equivalence classes called elementary sets, based on some definition of 'equivalence' as it relates to the application domain.

Any finite union of these elementary sets is called a definable set. A *rough set* $X \subseteq U$, however, can be defined in terms of the definable sets in terms of its lower ($\underline{R}X$) and upper ($\overline{R}X$) approximation regions: $\underline{R}X = \{x \in U \mid [x]_R \subseteq X\}$ and $\overline{R}X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}$.

We may refer to $\underline{R}X$ as the positive region, U- $\overline{R}X$ as the negative region, and $\overline{R}X - \underline{R}X$ as the boundary or borderline region of the rough set X. The lower and upper approximation regions, then, allow the distinction between certain and possible inclusion in a rough set.

The rough relational database model captures all the essential features of the theory of rough sets including the notion of indiscernibility of elements through the use of equivalence classes and the idea of denoting an indefinable set by its lower and upper approximation regions

The attribute domains in this model are partitioned by equivalence relations designated by the database designer or user. Within each domain, a group of values that are considered indiscernible form an equivalence class. The query mechanism uses class equivalence rather than value equality in retrievals. A user may not know the particular attribute value, but might be able to think of a value that is equivalent to the value required. For example, if the query requests "COLOR = 'BROWN'", the result will contain all colors that are defined as equivalent to BROWN, such as TAN, SORREL, or CHESTNUT. Therefore, the exact wording of a query is less critical.

The rough relational database (Beaubouef and Petry 2000) retains significant features of the ordinary relational database. Both models represent data as a set of *relations* containing *tuples*. The relations themselves are also sets. The tuples of a relation are its elements, and like the elements of sets in general, are unordered and nonduplicated. A tuple $\mathbf{t_i}$ takes the form $(d_{i1}, d_{i2}, ..., d_{im})$, where d_{ij} is a *domain value* of a particular *domain set* D_j. Let $P(D_i)$ denote the powerset(D_i) - $\emptyset$.

Definition. A *rough relation R* is a subset of the set cross product $P(D_1) \times P(D_2) \times \cdots \times P(D_m)$.

A rough tuple **t** is any member of *R*, which implies that it is also a member of $P(D_1) \times P(D_2) \times \cdots \times P(D_m)$. If $\mathbf{t_i}$ is some arbitrary tuple, then $\mathbf{t_i} = (d_{i1}, d_{i2}, ..., d_{im})$ where $d_{ij} \subseteq D_j$.

Definition. Tuples $\mathbf{t_i} = (d_{i1}, d_{i2}, ..., d_{im})$ and $\mathbf{t_k} = (d_{k1}, d_{k2}, ..., d_{km})$ are *redundant* if $[d_{ij}] = [d_{kj}]$ for all j = 1,..., m, and where $[d_{ij}]$ denotes the equivalence class to which d_{ij} belongs.

Attribute Oriented Generalization

Generalization of data is typically performed with utilization of concept hierarchies on an attribute-by-attribute basis, applying a separate concept hierarchy for each of the generalized attributes included in the relation of task-relevant data.

The basic steps / guidelines for attribute-oriented generalization in an object-oriented database are summarized below (Han and Kamber 2006):

1. An initial query to the database provides the starting generalization relation R which contains the set of data that is relevant to the user's generalization interest.

2. If there is a large set of distinct values for an attribute but there is no higher level concept provided for the attribute, the attribute should be removed in the generalization process.

3. If there exists a higher-level concept in the concept tree for an attribute value of a tuple, the substitution of the value by its higher-level concept generalizes the tuple.

4. Two generalized tuples may become similar enough to be merged, so we include an added attribute, Count, to keep track of how many objects have been merged to form the current generalized relation. The value of the count of a tuple should be carried to its generalized tuple and the counts should be accumulated when merging identical tuples in generalization.

5. The generalization is controlled by providing levels that specify how far the process should proceed. If the number of distinct values of an attribute in the given relation is larger than the generalization threshold value, further generalization on this attribute should be performed. If the number of tuples in a generalized relation is larger than their generalization threshold value, the generalization should proceed further. We can then extract characteristic rules from generalized data.

Our example is based on a herd management database for a large ranch housing various types and breeds of livestock. Some attributes have been partitioned into equivalence classes as shown. Tuples of a boundary region of a rough relation are shown in italics if there are any.

COLOR = {[BROWN, TAN, SORREL, CHESTNUT], [GRAY, GREY], [WHITE, BEIGE], [BLACK]}
BREED = {[BEEFMASTER, BM], [ANGUS, BLACK ANGUS], …}

The first step in the knowledge discovery process is to gather together in one rough relation that data that is relevant for the task at hand.

Tag	Breed	Cate-gory	Color	Problem/ Note
001	{ANGUS, JERSEY}	COW	BLACK	NONE
002	ANGUS	COW	BLACK	NONE
003	{ANGUS, BM}	CALF	BLACK	GETS_OUT
004	{ANGUS, BM}	CALF	BLACK	ESCAPES
005	BEEFMASTER	COW	TAN	HORNED
006	ANGUS	BULL	BLACK	DANGER
007	JERSEY	STEER	BROWN	STERILE
008	HOLSTEIN	COW	{BLACK, WHITE}	{STERILE, HORNED}
009	CHAROLAIS	COW	BEIGE	HORNED
020	BOER	KID	WHITE	{HORNED, GETS_OUT}
021	BOER	KID	BROWN	ESCAPES
022	NUBIAN	NANNY	GRAY	HORNED
…				

Figure 1. Relevant Relation of Herd Management Database

In the rough relational database, the generalization process is performed separately on both the "certain" tuples found in the lower approximation of the rough relation and the "possible" tuples found in the boundary region of the upper approximation of the rough relation. When the process is completed, rules can be ascertained with probabilities, keeping in mind that generalization involving the original uncertain tuples will result in rules that may or may not accurately represent truth—they are possible. However, they are useful in that they might provide insight for a domain expert, guiding further experimentation.

We will use concept hierarchy based on purpose of the attribute breed:

{meat-animal, dairy-animal} $\prec$ *livestock*
{dairy-goat, dairy-cattle} $\prec$ *dairy-animal*
{beef-cattle, meat-goat} $\prec$ *meat-animal*
{beefmaster, angus, charolais} $\prec$ *beef-cattle*
{ jersey, Holstein} $\prec$ *dairy-cattle*
{Boer, African } $\prec$ *meat-goat*
{Nubian, Saanen, Toggenburg } $\prec$ *dairy-goat*
{ Angora, Cashmere } $\prec$ *wool-goat…*

We start with data at lower levels of the concept hierarchy. For every tuple, replace that attribute value with the concept that generalizes it (its immediate parent) and accumulate counts for generalized tuples so that when duplicates are eliminated when merging redundant tuples, a count of the number of tuples merged is retained.

The tuples in the generalized relation are formulated into characteristic rules, assertions characterized by a majority of the data in the target class of the database, that data upon which the generalization was based. Quantitative information is also important as probabilities can be ascribed based on counts of tuples resulting from each generalization.

Breed	Cate-gory	Color	Problem/ Note	Count
{BEEF, DAIRY}	COW	BLACK	NONE	1
BEEF	COW	BLACK	NONE	1
BEEF	CALF	BLACK	GETS_OUT	2
BEEF	*COW*	*TAN*	*HORNED*	1
BEEF	BULL	BLACK	DANGER	1
DAIRY	STEER	BROWN	STERILE	1
DAIRY	COW	{BLACK, WHITE}	{STERILE, HORNED}	1
BEEF	COW	BEIGE	HORNED	1
MEAT	*KID*	*WHITE*	*{HORNED,*	1
MEAT	*KID*	*BROWN*	*GETS_OUT}*	1
			ESCAPES	
MEAT	*KID*	WHITE	*{HORNED, STERILE}*	1
DAIRY	NANNY	GRAY	HORNED	1

Figure 2. Final Stage of Rough Relation Generalization

The generalized relation that results represents some assertion of facts about the database. Within a tuple, the conjunction of attribute/value meaning is taken, and disjunction connects all the rules. Since we normally do not desire too many disjunctions, we generalize only to a threshold value, such as some minimal number of resultant tuples. For our example, the following non trivial rules are discovered: *Beef calves escape,* and *It is possible that white kid goats are horned.* From this simplified example we can see how generalization and rule formation occur.

We plan to investigate extensions to this approach by including fuzzy set uncertainty in the data as well. Additionally considerations of overlapping categories in a hierarchy will be studied.

Acknowledgment

The authors would like to thank the Naval Research Laboratory's Base Program, Program Element No. 0602435N for sponsoring this research.

References

Beaubouef, T. and Petry, F. 2000. Fuzzy Rough Set Techniques for Uncertainty Processing in a Relational Database. *International Journal of Intelligent Systems*, vol. 15, Issue 5, pp. 389-424, April, 2000.

Han J. and Kamber M. 2006. *Data Mining: Concepts and Techniques*. 2nd ed San Diego, CA Morgan Kaufmann.

Pawlak, Z. 1984. Rough Sets. *Int. Journal of Man-Machine Studies*, vol. 21, 1984, pp. 127-134.