Vol. 38 No. 4 Transactions of Information Processing Society of Japan Apr. 1997

Regular Paper

A Multi-phase Process for Discovering, Managing, and Refining Strong Functional Relationships Hidden in Databases

NING ZHONG† and SETSUO OHSUGA††

Functional relationships are important regularities hidden in databases. Since erroneous data can be a significant problem in real-world databases and the contents of most databases are ever changing, functional relationships that can be discovered from databases are usually *strong* ones that hold qualitatively for the collected data. Moreover, the discovery process is a *multi-phase* process based on incipient hypothesis generation, evaluation, management, and refinement. In this process, it is necessary to perform multi-aspect intelligent data analysis and multi-level conceptual abstraction and learning by combining AI techniques with statistical methods in multiple learning phases. This paper describes a *multi-phase* process for discovering, managing, and refining strong functional relationships hidden in databases.

1. Introduction

Knowledge discovery in databases (KDD) is becoming an important topic in artificial intelligence and is attracting the attention of leading database researchers [4]. The purpose of KDD is to elicit knowledge (i.e., rules and regularities among attributes) from raw data in databases. Functional relationships are important regularities hidden in databases. Since erroneous data can be a significant problem in real-world databases (i.e., data in databases are generally uncertain and incomplete) and the contents of most databases are ever changing (i.e., data in databases are often deleted, added, or updated), functional relationships that can be discovered in databases are usually *strong* ones that hold qualitatively for the collected data [6),12),15),17]. Moreover, the discovery process is a *multi-phase* process based on incipient hypothesis generation, evaluation, management, and refinement, as shown in **Fig. 1**. In this process, it is necessary to perform multi-aspect intelligent data analysis and multi-level conceptual abstraction and learning in multiple learning phases [16),18].

In previous related work on machine discovery, many researchers have investigated the discovery of laws from scientific experimental data. Bacon, Fahrenheit and Abacus are well-known systems [3),7),8] that use machine-learning methods to discover scientific (numeric) laws from scientific (numeric) data in the domains of physics and chemistry. Their main advantage is that data-driven heuristics can be used to discover a wider class of numeric laws. These systems have led to some successes, and provide a good background for our work. However, their object is only scientific (numeric) data, their capabilities for handling more uncertain data and their search control are weaker, they cannot be integrated with the knowledge-driven method, they cannot handle changes in data, and they do not address management and refinement. Moreover, their capability for discovery is also limited. For example, it is difficult to discover approximate quadratic functions such as $Y = a_0 + a_1 X + a_2 X^2 + \varepsilon$ from a database.

We argue that the *multi-phase* process is an important methodology for knowledge discovery in databases. That is, a KDD process is usually a *multi-phase* process, which involves data preparation, preprocessing, search for hypothesis generation, knowledge evaluation, representation, refinement, and management. Furthermore, the process may repeat at different intervals when new/updated data come. Although the process-centered view has recently been accepted by many researchers in the KDD community, few KDD systems provide the capabilities that a more complete process should possess [4]. This paper describes a *multi-phase* process for discovering, managing, and refining *strong* functional relationships hidden in databases. In a sense, the KDD process described in this paper can be regarded as a demonstration of the process-centered KDD methodology that extends the Bacon system

† Department of Computer Science and Systems Engineering, Faculty of Engineering, Yamaguchi University

†† Department of Information and Computer Science, School of Science and Engineering, Waseda University

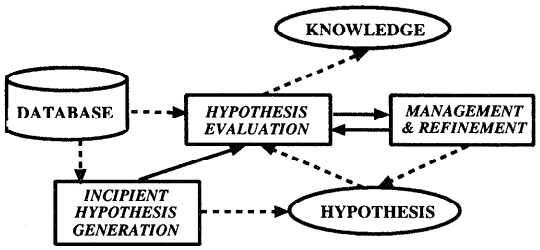

Fig. 1 The process of knowledge discovery in databases.

and its several successors to support qualitative and quantitative discovery, as well as to handle more uncertain data and changes in data[3),7),8)]. The key point of this extension is to enhance the capability of processing uncertainty systematically by extending heuristic search and search control, as well as combining AI techniques with statistical methods in the KDD process based on generation, evaluation, management, and refinement.

In the following sections, we will describe in detail the KDD process. The description will include how to generate and evaluate strong functional relationships by cooperatively using heuristic search and regression analysis, how to represent the discovered strong functional relationships as deductive rules and sets of data showing the errors of these rules in a knowledge-base, and how to manage and refine these rules by using quantitative inheritance, meta-reasoning, and so on. Finally, we offer some concluding remarks and discuss our future work.

2. Generation and Evaluation

In order to find *strong* functional relationships in databases, we extended and revised the heuristic search method developed in Bacon[7),8)] by using the process of *generation* and *evaluation* shown in Fig. 1, in which Qualitative Mathematics in qualitative reasoning[5),9)], heuristic search, domain/meta-knowledge, and statistical methods are used cooperatively.

2.1 Generation by Heuristic Search

Here, we first define several concepts and terms, and then describe how to generate the hypothetical functional relationships.

Definition 1. Monotonicity between two attributes.

We say that there is *monotonicity* between the attributes X_I and X_J if X_I increases or decreases as X_J increases.

Let $X_I \propto_Q X_J$ denote that there is monotonicity between X_I and X_J, and $X_I \overset{not}{\propto}_Q X_J$ denote that there is no monotonicity between X_I and X_J. □

Definition 2. Qualitative values in an attribute.

Let *qualitative values* in an attribute be ranges of values in this attribute that are generated by using *landmark*[5)], domain knowledge, or other some method.

Let $[X_I]$ denote the qualitative values in the attribute X_I. □

Definition 3. Sets in an attribute.

Let *sets* in an attribute be groups of data in this attribute corresponding to qualitative values in another attribute.

Let $\{X_J\}_I$ denote the sets that correspond to the qualitative values $[X_I]$ in the attribute X_J. □

Definition 4. Contradictory values.

We say that *contradictory values* are values that destroy the monotonicity between the attributes X_I and X_J, or between the qualitative values $[X_I]$ and the sets $\{X_J\}_I$. □

Definition 5. Probability of contradictory values.

Let the *probability* of contradictory values be the ratio of the number of contradictory values to the total number of data in an attribute. □

Definition 6. Approximate monotonicity between two attributes.

We say that there is *approximate* monotonicity between two attributes if the probability of contradictory values is smaller than the threshold value.

Let $X_I \dot{\propto}_Q X_J$ denote that there is approximate monotonicity between the attributes X_I and X_J, and let $[X_I] \dot{\propto}_Q \{X_J\}_I$ denote that there is approximate monotonicity between the qualitative values $[X_I]$ and the sets $\{X_J\}_I$. □

On the basis of these definitions, we have developed several heuristics for finding *strong* functional relationships[15)]. Here, as an example, we would like to describe how to execute one of these heuristics,

- If $X_I \overset{not}{\propto}_Q X_J$, but $X_I \dot{\propto}_Q X_J$ or $[X_I] \dot{\propto}_Q \{X_J\}_I$, then hypothesize that there is a strong functional relationship between X_I and X_J.

When this heuristic is called, it is executed by

the following method:

> First, define the qualitative values $[X_I]$ for the attribute X_I, and form the sets $\{X_J\}_I$ for the attribute X_J;
> Then, evaluate the uncertainty of contradictory values that destroy the monotonicity between the qualitative values $[X_I]$ and the sets $\{X_J\}_I$;
> If the probability of contradictory values is smaller than the threshold value, then hypothesize that there is a strong functional relationship between X_I and X_J.

Example 1. By using the heuristic stated above for a sample database of information on stars[13], which consists of several attributes such as the cluster designations of stars, their V magnitude, color indexes B-V and U-B, effective temperature, luminosity, mass, and so on, as shown in **Table 1**, we can find several strong functional relationships between the attribute *effTemp* and some other attributes such as *luminosity*, and the color indexes *B-V* and *U-B* in the clusters by the following steps:

Step 1: From our general knowledge of astronomy, we know that in comparison with field stars, members of an open cluster are more suitable objects for such calculations, because of the homogeneous chemical composition in the clusters and the reliability of luminosity and temperatures determined from the cluster UBV data[13]. Hence, we assume that (1) a user wants to investigate whether there are some functional relationships between the attribute *effTemp* and some other attributes such as the *luminosity* and the color indexes *B-V* and *U-B* in the clusters; and (2) the user provided background knowledge for defining the qualitative values of *effTemp*:

$[effTemp]_1 = [5.0 \sim 4.5]$
$[effTemp]_2 = (4.5 \sim 4.0]$
$[effTemp]_3 = (4.0 \sim 3.5]$
$[effTemp]_4 = (3.5 \sim 3.0]$.

That is, these qualitative values are used as the criterion for controlling the search for the *strong* functional relationships.

Table 1 DB: stars.

cl.	v-Mag.	B-V	U-B	effT.	lum.	...
1	13.18	0.30	−0.33	4.19	2.19	
1	14.62	0.34	0.12	4.02	1.32	...
1	14.37	0.67	−0.06	3.87	1.11	
⋮	⋮	⋮	⋮	⋮	⋮	⋮
2	12.22	0.83	0.48	4.02	2.41	
2	12.22	0.83	0.48	4.02	2.41	...
2	13.65	0.76	−0.09	4.23	2.12	
⋮	⋮	⋮	⋮	⋮	⋮	⋮
6	9.27	0.17	0.10	3.95	2.37	
6	9.27	0.17	0.10	3.95	2.37	...
6	10.04	0.52	0.03	3.83	1.90	
⋮	⋮	⋮	⋮	⋮	⋮	⋮
11	12.98	1.62	1.63	3.83	2.54	
11	12.98	1.62	1.63	3.83	2.54	...
11	14.44	0.84	0.16	4.18	2.40	
...	...	...	...	...	...	...

Step 2: Since there is a nominal attribute called *clusters* that can be used for forming clusters, sets are formed by using the qualitative values [*effTemp*] defined in *Step* 1 for every cluster. For example, sets can be formed for the *luminosity* attribute of $cluster_1$ and $cluster_8$ as follows:

$\{luminosity\}_{1-c_1} = \{\}$
$\{luminosity\}_{2-c_1} = \{1.07,1.1,1.16,1.32,1.39,1.41, \ldots, 3.18,3.29,3.39,3.48,4.08,4.33\}$
$\{luminosity\}_{3-c_1} = \{1.04,1.06,1.11,1.13,1.14,1.15,\ldots, 1.4,1.41,1.5,1.52,1.58,1.66,1.73\}$
$\{luminosity\}_{4-c_1} = \{\}$
$\{luminosity\}_{1-c_8} = \{4.15,4.49,4.52,4.64,4.7\}$
$\{luminosity\}_{2-c_8} = \{2.85,3.11,3.18,3.22,3.29,3.3,\ldots, 3.94,4.04,4.05,4.1,4.18,4.3,4.32,4.6\}$
$\{luminosity\}_{3-c_8} = \{2.07,2.16,2.29,\ldots, 2.49,2.59,3.24,3.3\}$
$\{luminosity\}_{4-c_8} = \{\}$.

Step 3: Evaluate the uncertainty of contradictory values. That is, calculate their probabilities. For example, the probability of the contradictory values between $\{luminosity\}_{2-c_1}$ and $\{luminosity\}_{3-c_1}$ for $cluster_1$ is 0.097.

Step 4: If the probabilities of the contradictory values are smaller than the threshold value, then hypothesize that there are strong functional relationships between the attribute *effTemp* and some other attributes such as the *luminosity*, and the color indexes *B-V* and U-B for some clusters:

effTemp $\dot{\propto}_Q$ *luminosity*, and *effTemp*, *luminosity* $\in$ $cluster_1$, $cluster_3$, $cluster_4$, $cluster_5$, $cluster_7$, $cluster_8$, $cluster_{10}$, $cluster_{11}$ and $cluster_{12}$;
effTemp $\dot{\propto}_Q$ *B-V*, and *effTemp*, *B-V* $\in$ $cluster_1$, $cluster_2$, $cluster_3$, $cluster_4$, $cluster_5$, $cluster_7$, and $cluster_{12}$;
effTemp $\dot{\propto}_Q$ *U-B*, and *effTemp*, *U-B* $\in$ $cluster_1$, $cluster_2$, $cluster_3$, $cluster_4$, $cluster_5$, $cluster_7$, $cluster_8$, $cluster_{10}$, $cluster_{11}$ and $cluster_{12}$. □

These hypothetical functional relationships are first stored in a model-base by using qualitative equations[15], and then evaluated by the statistical methods to be given in Section 2.2.

2.2 Evaluation by Regression Analysis

In general, regression analysis is a statistical method for finding a *structural characteristic*

(also called a structural relation, i.e., a kind of approximate functional relationship represented by the regression model) hidden in the data, which allows the value of the objective variable to be predicted or inferred from the values of descriptive variables. Furthermore, a structural characteristic is a *strong* functional relationship if it was qualitatively inferred by means of the heuristics given in Section 2.1. Thus, *strong* functional relationships are finally represented as regression models, so that they can be easily managed and refined [17].

In our application, the main goals of evaluation by regression analysis are to find the optimal regression model (i.e., the best structural characteristic) among variables that have a hypothetical linear (or quadratic) functional relationship and simultaneously detect to the error of that model. If it is inferred that there is a linear (or quadratic) functional relationship between X_I and X_J by using the heuristics as stated in Section 2.1, then let $Y = X_I$ be an objective variable, X_J be a descriptive variable, and ε be the error (or the residual) that is normally distributed with a mean of zero and a variance of σ^2. The number of descriptive variables can be more than one. For evaluation of the hypothetical functional relationships, three methods of regression analyses, multiple regression (MR), polynomial regression (PR), and auto-regression (AR), are provided in our system.

However, it usually does not mean that the optimal regression model can be obtained by increasing the number of descriptive variables. Although it may cause a decrease in the variance, it may also decrease the stability of the regression model. Hence, it is very important to select the most effective set of descriptive variables in order to obtain the optimal model. For this purpose, we need a criterion. A criterion called the Akaike Information Criterion (AIC), introduced by Akaike for evaluating the accuracy of prediction, can be used to select the best model [1]. It is defined as follows:

$$AIC = -2 \times (Maximum\ Likelihood\ of\ Model) + 2 \times (Numbers\ of\ Parameters\ of\ Model).$$

Among all combinations of potential descriptive variables, the model that produces the smallest AIC value is the best one. That is, we select the model with both better stability and smaller variance as the *structural characteristic* discovered by AIC.

Table 2 The structural characteristics discovered in the star database.

Clusters	Polynomial regression models
1	$Y = 4.83 + 0.08X_{lum} + \varepsilon$
3	$Y = 5.072 - 0.183X_{lum} + 0.044X_{lum}^2 + \varepsilon$
...	.....
1	$Y = 4.987 + 0.337X_{b-v} - 0.2X_{b-v}^2 + \varepsilon$
2	$Y = 4.673 + 0.677X_{b-v} - 0.23X_{b-v}^2 + \varepsilon$
...	.....

Table 3 The σ^2 and AIC values of the discovered structural characteristics shown in Table 2.

Clusters	σ^2	AIC
1	0.0183	−140.303
3	0.0125	−170.801
...	...	...
1	0.0139	−173.207
2	0.0285	−105.978
...	...	...

Example 2. We can evaluate the hypothesized *strong* functional relationships found by using the heuristic search described in Example 1, and select the best ones as the *structural characteristics* by using regression analysis and AIC. **Tables 2** and **3** show a part of the results. In Table 2, Y is *effective temperature*, X_{lum} is *luminosity* and X_{b-v} is *B-V*. □

The process described in this section leads from qualitative to quantitative discovery through the cooperative use of heuristic search and statistical methods. Although regression analysis can both generate and evaluate functional relationships, it has several limitations [15]. On the other hand, if we use only AI techniques such as heuristic search, the problem of handling more uncertain data cannot be solved satisfactorily. Hence, we try to combine both types of technique, so that we can

- Use heuristics, domain knowledge, and so on for search control. Thus, the search for generation of functional relationships are not blind but heuristic;
- Conduct qualitative analysis to determine whether there are strong functional relationships.

Sometimes, if an approximate functional relationship has a larger error that is only generated and evaluated by regression analysis, this does not mean that the relationship should be rejected, but that there may be an interesting approximate regularity for some users.

3. Management and Refinement

Since the discovered structural characteristics are finally denoted by regression models, as stated in Section 2.2, the management and refinement of the structural characteristics are essentially those of regression models that are represented as deductive rules and sets of data for showing the errors of those rules. This section will describe in detail the management and refinement of the structural characteristics.

3.1 Inheritance Inference on Regression Models

Inheritance inference on regression models is a central task in managing and refining structural characteristics discovered in databases. Inheritance inference is used to find matches with models for situations similar to those under study, in order to obtain a starting model for analysis. A good starting model can save a user much time, and effective inference can also save storage space by eliminating the need to save similar models. Here, we would like to describe two kinds of inheritance inferences, *downward* and *upward*[14),17)], which can be mainly used for three purposes:

- Inferring the model for representing the structure of a sample data set when only a sub-set of this data set is known or used.
- Inferring the varying degree of a model when the sample data set is partly updated (added/deleted).
- Refining and managing a family of regression models.

It is important that inheritances are quantified for their utility. That is, the strength of the inheritance inference is quantified, or the inherited numeric parameters are themselves quantified. These quantizations assume that the data sets in a database are like simple random samples with respect to one another. Although exact values and bounds derived from inheritance cannot be argued, estimates, which are the most common type of statistical inheritance, can be roughly quantified as to our certainty about the value derived. From a statistical point of view, this means finding standard errors of estimates in using statistics of related sets. If we can approximate the relationship between a target and a related set as a sampling process in either directions, we can use sampling theory for this[2),14)].

Theorem 1. If the mean of the set is approximately a mean of the superset, then the level of approximation has a standard error of

$$\tau = \sigma\sqrt{\frac{1}{n} - \frac{1}{N}}, \tag{1}$$

where n is the size of the set, N is the size of the superset, and σ is the standard deviation of the set or the superset.

Theorem 1 shows that the strength of inheritance is stronger the closer in size the two sets are. Sample theory also says that the standard deviation of the set will be approximately the standard deviation of the superset when n is not too small. Eq. (1) can be used as a criterion for selecting the best inheritance when there are more regression models that can be selected. That is, among all potential inheritance models, the one that produces the smallest τ value is the best model. However, Eq. (1) is not suitable for evaluating the strength of the inheritance between a set and its superset. This is because it is difficult to give the threshold value for distinguishing meaningful inheritance in advance. Fortunately, if we know the standard deviations of both a set and its superset, F distribution can be used for quantifying evaluation.

Definition 7. We say there is a stronger inheritance relationship between a set and its superset if

$$F = \frac{\sigma_1^2}{\sigma_2^2} \leq F_\alpha(n_1 - 1, n_2 - 1), \tag{2}$$

where σ_1^2 relates to the set with larger variance between a set and its superset, n_1 and n_2 are the sizes of the sets with σ_1^2 and σ_2^2 respectively, and α is either 0.05 or 0.01. □

Eq. (2) is called an F distribution with the degrees of freedom $d.f. = (n_1\text{-}1, n_2\text{-}1)$. This question is that of whether σ_1^2/σ_2^2 is too far from 1 to be explained by chance. Note that, unlike a comparison between two means, which is phrased in terms of the difference $\mu_1 - \mu_2$, a comparison between variances is formulated by using the ratio σ_1^2/σ_2^2. This is because a sampling distribution that is instrumental to the present inference situation involves the variances σ_1^2 and σ_2^2 only through the ratio.

Example 3. Inheritance inference can be used for managing/refining the structural characteristic when the sample data in a database are partly updated (deleted/added). To describe *downward inheritance*, every cluster of the star

database is divided into *group-1* as a set for fundamental data, as shown in Table 1, and *group-1-sub* as a subset of *group-1* obtained by deleting some data from *group-1* for its variation. Let the size of *group-1* for $cluster_1$ be 126 and its variance be 0.0183, and the size of *group-1-sub* for $cluster_1$ be 116 and its variance be 0.0173. Thus, using Eq. (2), we obtain $F = 1.058 < F_{0.05}(125, 115) \simeq 1.25$.

Example 4. We again use the star database as an example for describing *upward inheritance*. Every cluster of the database is divided into *group-1* as a set for fundamental data, and *group-1-sup* as a superset of *group-1* obtained by adding some data to *group-1* for its variation. Let the size of *group-1* for $cluster_1$ be 126 and its variance be 0.0183, and the size of *group-1-sup* be 146 and its variance be 0.0313. Thus, using Eq. (2), we obtain $F = 1.71 > F_{0.05}(145, 125) \simeq 1.25$.

3.2 Knowledge Representation

This section describes a method of knowledge representation based on the expansion capability of multi-layer logic [10),11)]. That is, when the domain set of a variable is finite, a formula of multi-layer logic can be expanded according to the following equivalent expressions:

$$[\forall X/\mathbf{x}]p(X) \cap \mathbf{x} = \{x_1, x_2, \ldots, x_n\} \longleftrightarrow p(x_1) \cap p(x_2) \cap \ldots \cap p(x_n),$$

$$[\exists X/\mathbf{x}]p(X) \cap \mathbf{x} = \{x_1, x_2, \ldots, x_n\} \longleftrightarrow p(x_1) \cup p(x_2) \cup \ldots \cup p(x_n).$$

This capability is used for extracting from a set the elements that possess specified properties. It is syntactically defined by appending "#" after the variable to be expanded in the prefix of the multi-layer logic formula. The following *Rule-1* is an example of using the expansion capability:

```
Rule-1: /* the rule for inferring the effective temperature of stars from the luminosity of stars */
!ins_e clusters 1, 3; /* use clusters 1, 3 */
!ins_e variance 0.0183, 0.0125; /* the variances of the regression models belonging to a family */
!ins_e ai-pr-1-0 4.83, 5.072; /* the coefficient A0 */
!ins_e ai-pr-1-1 0.08, -0.183; /* the coefficient A1 */
!ins_e ai-pr-1-2 0, 0.044; /* the coefficient A2 */
[∀ X-luminosity,Y-effTemp/float] [∀ Mode,Check-N/int]
[∀ A0#/ai-pr-1-0] [∀ A1#/ai-pr-1-1] [∀ A2#/ai-pr-1-2]
/* declare the domains of variables */
( | (p-stars Mode Check-N Y-effTemp X-luminosity)
/* infer the effTemp from the luminosity */
  ~($pr 2 Y-effTemp A0 X-luminosity A1 A2)
/* infer the effective temperature by the PR model */
  ~($scope_kb rule-set3)
/* transfer to the world: rule-set3 */
  ~(storeInfor Mode Check-N pr Y-effTemp X-luminosity)
/* store the inferred result and the variable */
).
```

Rule-1 is represented as a multi-layer logic formula in the If-Then form and a data set called *variance* to show the errors. Ordinary If-Then rules are represented in the form $A_1 \& A_2 \ldots A_n \rightarrow B$, but are written as (| (B $\sim A_1$ $\sim A_2 \ldots$ $\sim A_n$)) in our system by using the knowledge-based system KAUS [11)]. *Rule-1* reads "The effective temperature of stars can be inferred from their luminosity by the polynomial regression model. Also, the effective temperature inferred will be saved for future use together with the luminosity." On the other hand, "!*ins_e* **x** $x_1 \ldots x_n$;" means that $x_1 \ldots x_n$ are elements of **x**. This is the syntax used to represent the "set-element" relation in KAUS.

In particular, we see that in *Rule-1*, the symbol "#" denoting the expansion capability is used in the prefix of the multi-layer logic formula (i.e., [∀ A0#/ai-pr-1-0] ...) and the regression coefficients are recorded in the sets *ai-pr-1-0* $\sim$ *ai-pr-1-2*. This is a kind of model representation. That is, by means of the expansion capability, only the elements of sets are modified, but the multi-layer logic formula (rule) is not generally changed when the refinement (or generalization) is done. For example, if the regression coefficients are changed (added/updated; e.g., if the regression coefficients of the model obtained from $cluster_8$ are added), then only the values in the sets *ai-pr-1-0* $\sim$ *ai-pr-1-2* and *variance* are changed (added/updated).

3.3 Management

In our system, a rule chain and an inheritance graph corresponding to a family of regression models are used for management. By means of them, the following jobs can be done:

- Regression models discovered from databases are first stored in the rule chain, and then are refined by using the method to be described in Section 3.4.
- The time and history of regression models are represented and managed; that is, the rule chains for storing regression models are dynamically generated as time goes on, in order to record the evolution process of regression models.
- A suitable regression model is selected from a family of regression models by using the

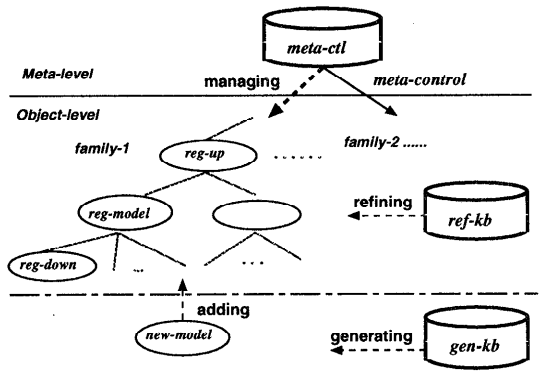

Fig. 2 An inheritance graph for a family of regression models and operations on it.

method described in Zhong and Ohsuga[17].

- The inheritance graph of regression models is dynamically generated, in order to describe the relationship among regression models.

The rule chain of regression models is defined by set-element relationships and the multi-layer logic formulae. The rule chain and the inheritance graph are managed by using a meta-knowledge level.

Figure 2 shows the structure of the inheritance graphs and some operations on them. That is, *reg-up*, *reg-model*, *reg-down*, and so on shown in Fig. 2 denote several regression models belonging to a family with inheritance relationship. Three knowledge-bases, *meta-ctl*, *ref-kb* and *gen-kb*, which are divided into two knowledge levels, *meta-level* and *object-level*, are respectively used for controlling and performing different operations. A further example of the operations for refinement can be found in Example 5 in Section 3.4.

3.4 Refinement

On the basis of the preparation stated above, the method of refining the structural characteristics can be roughly described as follows:

- If structural characteristics (regression models) were discovered, then store them first as rules such as *Rule-1* in Section 3.2, in the dynamic worlds of a knowledge-base in the order in which they were discovered.
- If there are two or more regression models, then apply meta-reasoning to the worlds in which the discovered regression models are stored, in order to find their inheritance relationship.
- If there are two or more *true* results of meta-reasoning, and their answers are nearly the same, then evaluate the strength of inheritance among them.
  - If there are more than two regression models for several sets and their superset, then first use Eq. (1) to select the best model from the ones corresponding to the sets, and use Eq. (2) to evaluate the strength of inheritance between the selected best model and the model corresponding to its superset.
  - If there are only regression models corresponding to sibling sets or two models for a set and its superset, then use Eq. (2) to evaluate the strength of inheritance between them.
  - If Eq. (2) holds and the regression model discovered later has a smaller variance, store the coefficients and the variance value of the regression model discovered later, and delete the coefficients and the variance value of the older regression model. Then delete the rule discovered later (i.e., delete the multi-layer logic formula and the sets in which the coefficients and the variance value are stored), and create or revise the inheritance graph of the regression model's family and the corresponding rule chain.
- If there are two or more *true* results of meta-reasoning, but their answers are different, then generalize the rules stored in the dynamic worlds (i.e., merge the multi-layer logic formulae and the sets in which the coefficients and the variance value are recorded), and create or revise the inheritance graph of the regression model's family and the corresponding rule chain.
- If there are not two or more *true* results of meta-reasoning, then the rules stored in the dynamic worlds are not modified.

Example 5. After the above refinement has been carried out for the cases described in Examples 3 and 4, *Rule-1* is changed into the following *Rule-2*:

Rule-2: /* the rule for inferring the effective temperature of stars from their luminosity */
!ins_e clusters **1-sub, 1-sup,** 3; /* use a subset and superset of cluster-1, and cluster-3 */
!ins_e variance **0.0173, 0.0313,** 0.0125; /* the variances of the regression models belonging to a family */
!ins_e ai-pr-1-0 **4.856, 4.673,** 5.072;
/* the coefficient A0 */
!ins_e ai-pr-1-1 **0.075, -0.0235,** -0.183;
/* the coefficient A1 */

```
!ins_e ai-pr-1-2 0, 0.2116, 0.044;
  /* the coefficient A2 */
[∀ X-luminosity,Y-effTemp/float] [∀ Mode,Check-N/int]
[∀ A0#/ai-pr-1-0][∀ A1#/ai-pr-1-1] [∀ A2#/ai-pr-1-2]
( | (p-stars Mode Check-N Y-effTemp X-luminosity)
  ~($pr 2 Y-effTemp A0 X-luminosity A1 A2)
  ~($scope_kb rule-set3)
  ~(storeInfor Mode Check-N pr Y-effTemp X-
luminosity)
).
```

In the above rule, the parts in bold type are the ones that have been revised. That is, we see that the regression model for *group-1* is replaced by the one for *group-1-sub*, because there is a stronger inheritance relationship between them and the variance of the regression model for *group-1-sub* is smaller than the one for *group-1*. The regression model for *group-1-sup* is added because there is not a stronger inheritance relationship between the regression models for *group-1-sup* and *group-1-sub*. In this case, the corresponding inheritance graph for $cluster_1$ is as follows:

```
!ins_e *reg-inheritance-graph-cluster1 r1, r2; /* r1, r2
are reg-models in the inheritance-graph for cluster1 */
!ins_e r1 reg-model, reg-down; /* r1 is the downward
inheritance model of the reg-model */
!ins_e r2 reg-model, reg-up; /* r2 is the upward inheri-
tance model of the reg-model */.
```

4. Concluding Remarks

We have presented a *multi-phase* process for discovering, managing, and refining *strong* functional relationships hidden in databases. That is, we have described a more complete KDD process based on incipient hypothesis generation, evaluation, management, and refinement. We support qualitative/quantitative discovery, handle more uncertain data and changes in data, and control the discovery process by combining AI techniques with statistical methods in multiple learning phases. The KDD process described in this paper is the basic one that is executed in our GLS discovery system [16),18)].

Here, we would like to stress that although this paper has shown how to combine several AI techniques with statistical methods in a KDD process with multiple learning phases, its aim is not to create a complete description of all elemental techniques that should be used in the process, since this would require much more space than is available here. In fact, we have developed a systematic method for discovering functional relationships in databases by combining attribute calculation, in which a model base and heuristic search are used cooperatively, with regression analysis [15)]. In this method, non-linear functions can be also generated if multiple regression is used, although in this paper we only described the case in which polynomial regression is used, because non-linear functional relationships can be transformed into linear ones by attribute calculation before using multiple regression. Since the issue lies beyond the scope of this paper, we would like to describe it in detail in another paper.

Several issues remain to be investigated. Currently, the capabilities for management and refinement in our system involve mainly one of two main aspects, namely, how to manage and refine strong functional relationships discovered after the data in a database have been updated (added/deleted). Another important aspect of management and refinement is how to acquire higher-level knowledge from several strong functional relationships discovered in databases. The development of this aspect is a further extension of the KDD process described in this paper, for which need to use more domain knowledge.

Acknowledgements

The authors would like to thank Dr. Maria Zemankova and Prof. Jan Zytkow for their valuable comments in the ISMIS conferences and their help. Further, we would like to thank Professor Hori and Professor Takasu for useful discussions in OHlab seminar and SIG-KDD meeting. The authors also thank Mr.Yamauchi for his help in use of KAUS. Finally we would like to thank the referees of the paper for their valuable comments.

References

1) Akaike, H.: A New Look at the Statistical Model Identification. *IEEE Trans. Autom. Contr.*, Vol.AC-19, pp.716–723 (1974).
2) Cochran, W.G.: *Sample Techniques*, Wiley (1977).
3) Falkenhainer, B.C. and Michalski, R.S.: Integrating Quantitative and Qualitative Discovery in the Abacus System, *Machine Learning – An Artificial Intelligence Approach*, Michalski, R.S., et al. (Eds.), Morgan Kaufmann, pp.153–190 (1990).
4) Fayyad, U.M., et al. (Eds.): *Advances in Knowledge Discovery and Data Mining*, AAAI Press (1996).
5) Forbus, K.D.: Qualitative Process Theory, *Artificial Intelligence*, Vol.24, pp.95–168 (1984).

6) Hoschka, P. and Klosgen, W.: A Support System for Interpreting Statistical Data, *Knowledge Discovery in Databases*, Piatetsky-Shapiro and Frawley (Eds.), pp.325–345 (1991).
7) Langley, P., Simon, H.A., Bradshaw, G.L. and Zytkow, J.M.: *Scientific Discovery – Computational Explorations of the Creative Processes*, MIT Press (1987).
8) Langley, P. and Zytkow, J.M.: Data-driven Approaches to Empirical Discovery, *Artificial Intelligence*, Vol.40, Nos.1-3, pp.283–312 (1989).
9) Mavrovouniotis, M. and Stephanopoulos, G.: Reasoning with Orders of Magnitude and Approximate Relations, *Proc. AAAI-87*, pp.626–630 (1987).
10) Ohsuga, S. and Yamauchi, H.: Multi-layer Logic – A Predicate Logic Including Data Structure as Knowledge Representation Language, *New Generation Computing*, Vol.3, No.4, pp.403–439 (1985).
11) Ohsuga, S.: Framework of Knowledge Based Systems, *Knowl. Based Sys.*, Vol.3, No.4, pp.204–214 (1990).
12) Piatetsky-Shapiro, G.: Discovery, Analysis, and Presentation of Strong Rules, *Knowledge Discovery in Databases*, Piatetsky-Shapiro and Frawley (Eds.), pp.229–248, AAAI Press (1991).
13) Piskunov, A.E.: Bull. Inf. CDS, Vol.19, p.67 (1980).
14) Rowe, N.C.: Management of Regression-model Data, *Data & Knowl. Eng.*, Vol.6, No.4, pp.349–363 (1991).
15) Zhong, N. and Ohsuga, S.: An Integrated Calculation Model for Discovering Functional Relations from Databases, *Proc. 4th Inter. Conf. on Database and Expert Systems Applications (DEXA '93)*, LNCS, Vol.720, pp.213–220, Springer-Verlag (1993).
16) Zhong, N. and Ohsuga, S.: The GLS Discovery System: Its Goal, Architecture and Current Results, *Proc. 8th Inter. Symp. on Methodologies for Intell. Sys. (ISMIS'94)*, LNAI, Vol.869, pp.233–244, Springer-Verlag (1994).
17) Zhong, N. and Ohsuga, S.: Managing/Refining Structural Characteristics Discovered from Databases, *Proc. 28th Hawaii Int. Conf. on Sys. Sciences (HICSS-28)*, Vol.3, pp.283–292 (1995).
18) Zhong, N. and Ohsuga, S.: Toward A Multi-strategy and Cooperative Discovery System, *Proc. First Int. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, pp.337–342 (1995).
19) Zytkow, J.M. and Zembowicz, R.: Database Exploration in Search of Regularities, *J. of Intell. Infor. Sys.*, Vol.2, No.1, pp.39–81, KAP (1993).

(Received May 31, 1996)
(Accepted January 10, 1997)

Ning Zhong is currently an associate professor in the Department of Computer Science and Systems Engineering, Yamaguchi University, Japan. He is also a cooperative research fellow in Research Center for Advanced Science Technology (RCAST) at the University of Tokyo. He graduated at the Beijing Polytechnic University in 1982 and has been a lecturer in the Dept. of Computer Science, Beijing Polytechnic University. He received the Ph.D. degree in the Interdisciplinary Course on Advanced Science and Technology from the University of Tokyo. His research interests include knowledge discovery in databases, machine learning, and intelligent information systems. He is a member of IPSJ, JSAI, IEEE, ACM, AAAI, AFOS, and IRSS.

Setsuo Ohsuga is currently a professor in the Department of Information Science and Computer Science, Waseda University, Japan. He has been professor and director of Research Center for Advanced Science and Technology (RCAST) at the University of Tokyo. He has also been president of the Japanese Society for Artificial Intelligence (JSAI). He graduated at the University of Tokyo in 1957. From 1957 to 1961 he worked in Fuji Precision Machinery (the present Nissan Motors). In 1961 he moved to the University of Tokyo and received his Ph.D. in 1966. He became associate professor in 1967 and professor in 1981. His research interests include artificial intelligence, knowledge information processing, databases, and CAD. He is a member of the editorial boards of 9 scientific journals.