A Hypothesis Refinement Method for Summary Discovery in Databases

Do Heon Lee and Myoung Ho Kim

Department of Computer Science,
Korea Advanced Institute of Science and Technology,
373-1, Kusung-dong, Yusung-gu, Taejon, 305-701,
South Korea
Tel: +82-42-869-3563 Fax: +82-42-869-3510
E-mail: dhlee@adam.kaist.ac.kr

Abstract

As database systems are playing major roles in more and more applications, the amount of information in databases is rapidly growing. In order to comprehend those large volumes of information, computerized summary discovery methods are required. In this paper, we propose a hypothesis refinement method for constructing and evaluating fuzzy hypotheses. Based on them we propose an effective and robust algorithm to discover simple linguistic summaries. In addition, we present ideas for exploiting discovered summaries to various applications such as querying database knowledge, handling query failures and semantic query optimization.

KEYWORDS: knowledge discovery in databases, summary discovery, fuzzy set theory, query failure, semantic query optimization

1 Introduction

The size of databases in many data-intensive applications such as office automations, aerospace and other scientific databases is rapidly growing. For example, earth observation satellites, planned for the 1990s, are expected to generate one terabyte of data every day and the federally funded Human Genome project will store thousands of bytes for each of the several billion genetic bases[1]. The census databases are also typical examples of a large amount of information.

CIKM '93 - 11/93/D.C., USA

© 1993 ACM 0-89791-626-3/93/0011 ....$1.50

For those large volumes of information, manual analysis is no longer possible. A recent National Science Foundation workshop on the future of database research ranked data mining among the most promising research topics for the 1990s[2]. *Summary discovery* is one of the major parts of data mining techniques, which provide the user with comprehensible representations for a large amount of data.

There are several requirements for effective techniques of summary discovery from actual databases. First, since most databases contain nominal data as well as numerical data, discovery techniques must be able to deal with both types of data. Second, those techniques must be robust in a sense that some noisy data can not affect the major results. It is because actual databases usually contain noisy data due to either wrong inputs or genuine exceptions[1]. Third, the discovered summaries must have simple forms, because summary discovery is done to comprehend a large amount of data more easily.

The limitations of conventional statistical methods and inductive machine learning methods in applying to knowledge discovery in actual databases are well described in [1]. Although statistical methods have been useful tools for summarizing data, they are applicable only to numerical types of data. Artificial intelligence researchers have made many efforts to elicit some representative patterns from given fact sets, i.e. inductive learning[3]. Typically, those approaches assume that noise-free fact sets and relatively small amount of facts. Those assumptions are no longer true for the actual database in practice.

As a newly emerging field, there have been several database-oriented researches on summary discovery techniques[4][5][6][7][8]. Many of them exploit some kind of concept trees/hierarchies that represent relationships among various domain concepts. In fact,

the exploitation of such domain knowledge is a crucial component of knowledge discovery in databases[1]. Each node in a concept tree represents a concept in the problem domain. The ancestor-descendant relationships among nodes in the tree are corresponding to general-specific relationships among concepts. The techniques assumes that each concept has rigid boundary. In other words, when several elements constitute a concept, each element either belongs to the concept completely, or does not belong to it at all.

However, there are many exceptions in real world. For instance, even though the genetic engineering may be thought to have some aspects of the natural science, it is hard to say that the genetic engineering is a pure natural science subject. The argument that the genetic engineering is not a natural science subject may be also unacceptable. The following example illustrates how the rigid boundary problem makes difficulties in summary discovery. Suppose that Figure 1 is an enrollment relation in a biology department.

NAME	COURSE	DEPT	SEX
Lee	genetic eng.	biology	male
Kim	chemistry	biology	female
Yoon	genetic eng.	biology	male
Park	genetic eng.	biology	male
Son	physics	biology	female
Choi	generic eng.	biology	male
Yang	generic eng.	biology	male

Figure 1: An example data relation ENROLLMENT

When a college curriculum analyst asks a question such as "Which subject between the natural science and the engineering is preferred by most students in the biology department?", a summary discovery procedure may be triggered. Is the answer that most of them prefer the natural science right? Otherwise, do they prefer the engineering?

In this paper, we describe a hypothesis refinement method for constructing and evaluating fuzzy hypotheses. Using the hypothesis refinement method, we propose an effective and robust algorithm to discover simple linguistic summaries, which accomodate nominal data as well as numerical data. The algorithm utilizes hierarchies of domain concepts with fuzzy boundaries rather than rigid ones. Furthermore, the discovered summaries can be used for querying database knowledge, handling query failures and semantic query optimization.

The rest of the paper is organized as follows. Section 2 describes the hypothesis refinement method and Section 3 proposes a method to discover summaries. The possible applications of discovered summaries are discussed in Section 4. Section 5 concludes and mentions the future research problems.

2 A Hypothesis Refinement Method

A hypothesis is an assertion supposed to be true. This section defines a fuzzy hypothesis and describes how fuzzy hypotheses are constructed and evaluated. After presenting the evaluation method of fuzzy hypotheses in Section 2.2, we describe how appropriate fuzzy hypotheses are constructed in Section 2.3 and 2.4.

2.1 Fuzzy Hypotheses

Since fuzzy concepts are effective to express complex phenomena in simplified forms [19], we adopt fuzzy hypotheses, i.e., hypotheses having fuzzy concepts, as vehicles to contain hypothetical summaries that are easily comprehensible. A fuzzy concept is represented as a fuzzy restriction, i.e. a fuzzy set[18]. The truth of a fuzzy hypothesis becomes a fuzzy value rather than either TRUE of FALSE. We use the term "evaluation" instead of "proving" to mean the activity of testing the truth of a fuzzy hypothesis. This is because the truth value is no longer dichotomous in fuzzy application domains.

We use the following notational conventions herein. R (A_1, ..., A_m) is a relation, where A_j denotes an attribute name. If a relation R is represented as $R = \{t_1, \ldots, t_n\}$, then t_i denotes a tuple. A fuzzy set $\tilde{F}_j$ is represented by its membership function $\mu_{\tilde{F}_j}(x)$, where x is an element on the domain, i.e. the universe of discourse.

Definition 1

A *Fuzzy Hypothesis* is defined recursively as follows.

- "A_j is $\tilde{F}_j$" is an *atomic fuzzy hypothesis*, where $\tilde{F}_j$ is a fuzzy restriction defined on the domain of A_j. An atomic fuzzy hypothesis is a fuzzy hypothesis.
- If H and I are fuzzy hypotheses, then the conjunction of H and I, i.e., $H \wedge I$, is also a fuzzy hypothesis.

A *Fuzzy Hypothesis Template* is a parameterized fuzzy hypothesis where all fuzzy restrictions are variables. A fuzzy hypothesis tem-

plate is said to be *instantiated* when its variables are all substituted by specific fuzzy restrictions.

(End of Definition 1)

Examples of fuzzy hypotheses are "AGE is $\widetilde{young}$" and "($SALARY$ is $\widetilde{high}$) $\wedge$ (AGE is $\widetilde{young}$)". We focus only on fuzzy hypotheses in this paper rather than hypotheses in general. Thus, we use "a hypothesis" to denote "a fuzzy hypothesis" thoughout this paper for simplicity.

2.2 Evaluation of Fuzzy Hypotheses

When a database relation R and fuzzy restrictions $\tilde{F}_j$'s are given, we can evaluate the truth of a hypothesis H with repspect to the relation R. The truth of a hypothesis depends on the degree to which the tuples in the relation R support the hypothesis H. We use $SD_R(H)$ to denote the support degree of the relation R for the hypothesis H. We will omit a relation R in $SD_R(H)$, i.e., $SD(H)$, if there is no ambiguity. In what follows, we describe how fuzzy hypotheses are evaluated.

- An atomic fuzzy hypothesis
  - $SD(A_j \ is \ \tilde{F}_j) = [\sum_{t_i} \mu_{\tilde{F}_j}(t_i.A_j)]/n$, where n is the number of tuples.
- Conjunctions of fuzzy hypotheses
  - $SD(H \wedge I) = SD(H) \bigotimes SD(I)$, where $\bigotimes$ denotes a t-norm opertor[19].

Consider an atomic hypothesis. The innermost term, $\mu_{\tilde{F}_j}(t_i.A_j)$ represents the degree to which individual tuple t_i supports the hypothesis H. Then, the sum of degrees over all the tuples represents the degree to which all relevant tuples support the hypothesis. The division by n is for normalization, i.e. making the result fall into the unit interval [0,1]. As a result, the support degree represents the ratio of the number of supporting tuples to the number of total relevant ones. Similar notions have been widely used in evaluating fuzzy quantifiers[5][18][20].

In evaluating conjunctive hypotheses, we use a t-norm operator to reflect the semantics of conjunction. A binary operator $\bigotimes$ belongs to the class of t-norm operators, which are widely used to represent conjunctions in fuzzy logical contexts, if it satisfies the following axiomatic properties[19].

When $a, a_i, b_i, c \in [0, 1]$,

1. $\bigotimes(0,0) = 0; \bigotimes(a,1) = \bigotimes(1,a) = a$,
2. $\bigotimes(a_1, b_1) \leq \bigotimes(a_2, b_2)$ *if* $a_1 \leq a_2$ *and* $b_1 \leq b_2$,
3. $\bigotimes(a,b) = \bigotimes(b,a)$,
4. $\bigotimes(\bigotimes(a,b),c) = \bigotimes(a,\bigotimes(b,c))$.

In addition to those common properties, each of them has its own specific properities, which determine the suitability of the operator for a specific appilcation domain. Examples of the $\bigotimes$ operator include MIN and product operators[19].

2.3 Construction of Fuzzy Hypotheses

We construct fuzzy hypotheses in a stepwise refinement fashion. Once we generate a collection of fuzzy hypotheses with the most general terms, we select hypotheses supported strongly by actual database. The selected hypotheses are refined to include more specific terms.

To facilitate the refinement procedure, we first define a fuzzy restriction hierarchy composed of concept nodes, each of which represents a concept with fuzzy boundaries.

Definition 2

A fuzzy restriction hierarchy is a partially ordered set, (Γ, $\subseteq$) where Γ is a set of fuzzy restrictions defined on the domain D. For $\tilde{F}_i$ and $\tilde{F}_j$ in Γ, $\tilde{F}_i \subseteq \tilde{F}_j$ *iff* $\forall x \in D, \mu_{\tilde{F}_i}(x) \leq \mu_{\tilde{F}_j}(x)$

$\tilde{F}_i$ is called a *specialization* of $\tilde{F}_j$ if $\tilde{F}_i \subseteq \tilde{F}_j$ and $\tilde{F}_i \neq \tilde{F}_j$. $\tilde{F}_i$ is called a *maximal fuzzy restriction* if there is no other $\tilde{F}_j$ such that $\tilde{F}_i \subseteq \tilde{F}_j$.

(End of Definition 2)

For example, if we define membership functions of "small" and "very small" as in Figure 2, "very small" is a specialization of "small". Note that fuzzy restriction hierarchies can be defined on nominal domains as well as numerical ones. Figure 10 in the next section gives an example of a hierarchy defined on a nominal domain. Based on a fuzzy restriction hierarchy, we define a fuzzy hypothesis hierarchy.

Definition 3

A fuzzy hypothesis hierarchy is a partially ordered set ($\aleph$, $\preceq$) where $\aleph$ is a set of fuzzy hypotheses. For H_i and H_j in $\aleph$, $H_i \preceq H_j$ *iff*

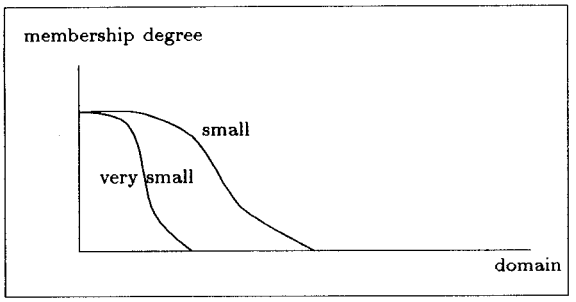

Figure 2: Membeship functions for "small" and "very small"

fuzzy hypothesis templates of H_i and H_j are the same as each other and each fuzzy restriction in H_i is either the same as or a specialization of its corresponding fuzzy restriction in H_j.

H_i is called a *specialization* of H_j if $H_i \preceq H_j$ and $H_i \neq H_j$. And, H_i is called a *direct specialization* of H_j if $H_i \preceq H_j$, $H_i \neq H_j$ and there is no other H_k such that $H_i \preceq H_k \preceq H_j$.

(End of Definition 3)

A fuzzy hypothesis hierarchy is constructed from the given fuzzy restriction hierarchy and a fuzzy hypothesis template. For example, suppose that we are given a fuzzy restriction hierarchy in Figure 3 and a fuzzy hypothesis template "(AGE is X) $\wedge$ (SCORE is Y)". Figure 4 shows a part of the fuzzy hypothesis hierarchy, where (α, β) denotes the hypothesis "(AGE is α) $\wedge$ (SCORE is β)".

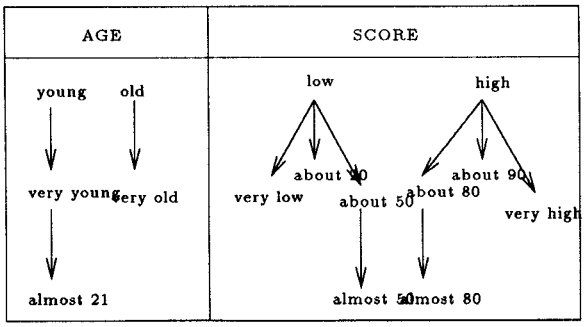

Figure 3: Fuzzy restriction hierarchies

As shown in Figure 4, if we have a small set of fuzzy restrictions at first, a large number of hypotheses can be derived due to all possible combinations. However, the following theorem gives an insight by which unnecessary derivations can be avoided.

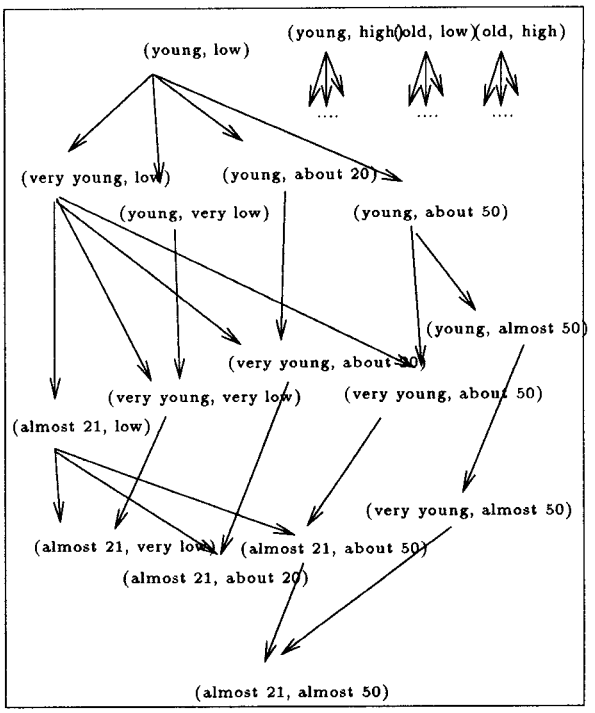

Figure 4: A fuzzy hypothesis hierarchy

Theorem 1 *If H is a specialization of I, the support degree of H can not be greater than that of I.*

Proof:

The theorem says that if $H \subseteq I$, $SD(H) \leq SD(I)$. We prove the theorem by induction.

The basis covers the case that H and I are atomic hypotheses. Suppose that $H = (A_i \text{ is } \tilde{F}_i)$ and $I = (A_i \text{ is } \tilde{G}_i)$. By definition of $SD()$, $SD(H) = \sum_i(\mu_{\tilde{R}}(t_i) \bigotimes \mu_{\tilde{F}_j}(t_i.A_j))$ and $SD(I) = \sum_i(\mu_{\tilde{R}}(t_i) \bigotimes \mu_{\tilde{G}_j}(t_i.A_j))$, And $H \subseteq I$ implies $\forall x, \mu_{\tilde{F}_j}(x) \leq \mu_{\tilde{G}_j}(x)$. So, $SD(H) \leq SD(I)$.

For the induction, we consider the case of conjunctive hypotheses, i.e. $H = H_i \wedge H_j$ and $I = I_i \wedge I_j$. Since H is a specialization of I, $H_i \subseteq I_i$ and $H_j \subseteq I_j$. So the induction hypothesis says that $SD(H_i) \leq SD(I_i)$ and $SD(H_j) \leq SD(I_j)$. By the monotonicity axiom of t-norm operators in Section 3.2, $SD(H_i \wedge H_j) = SD(H_i) \bigotimes SD(H_j) \leq SD(I_i) \bigotimes SD(I_j) = SD(I_i \wedge I_j)$, thus $SD(H) \leq SD(I)$.

Q.E.D

From Theorem 1, we can conclude that the more specific a hypthesis is, the less strongly it is supported.

The argument is, in fact, intuitively clear. For instance, consider two hypotheses "(AGE is $\widetilde{very\ young}$) $\wedge$ ($SCORE$ is $\widetilde{low}$)" and "(AGE is $\widetilde{very\ young}$) $\wedge$ ($SCORE$ is $\widetilde{very\ low}$)" in the hypothesis hierarchy in Figure 4. The latter is more specific than the former. In other words, the latter has more constraints to be satisfied in order to be supported by tuples in databases. Thus the latter can not have higher support degree than the former.

In practice, an assertion supported by only small number of tuples needs not be sustained. These tuples may even be noisy data that must be excluded to avoid disturbance. Recall that the support degree represents the ratio of the number of supporting tuples to the number of total relevant ones. If we empirically choose a small real number as a *threshold value* to identify an ignorable ratio, an hypothesis with a support degree lower than the threshold is not of interest. Thus, if the support degree of a hypothesis is lower than the given threshold value, the further specialization of it is not necessary by Theorem 1.

2.4 An Algorithm for Constructing and Supporting Fuzzy Hypotheses

We present an algorithm for constructing and evaluating fuzzy hypotheses. The following algorithm adopts the breadth first search technique with prunnings.

A Hypothesis Refinement Method

/* This algorithm constructs and evaluates fuzzy hypotheses */

Given Input: a data relation R
fuzzy restriction hierarchies
a fuzzy hypothesis template
a support degree threshold value τ for only meaningful assertions

Initial:
CS = hypotheses made by instantiating the given template with all possible combinations of maximal fuzzy restrictions.
/* candidate hypothesis set */
SS = ϕ /* selected hypothesis set */

SelectHypo(CS, SS) {
(1) while (CS $\neq \phi$) {
(2) for each H in CS {
(3) Compute $SD(H)$ on R ;
(4) if $SD(H) \geq \tau$ then {
(5) $SS = SS \cup \{ H \}$;
(6) $NextCS = NextCS \cup$ direct specializations of H;
(7) } /* end of if */
(8) } /* end of for */
(9) CS = NextCS ;
(10) } /* end of while */
}

(End of Algorithm)

In line (3), the support degree of each hypothesis is computed. Among them, only hypotheses supported higher than the given threshold value τ, are refined in line (6). In virtue of this threshold value, some noisy data due to either wrong inputs or genuine exceptions can not affect the major results. Each iteration in the while-loop corresponds to each refinement step.

Let us consider the efficiency of the algorithm in terms of the number of disk accesses. Since the number of (derived) hypotheses in one refinement step is relatively small, we can assume that the system buffer space can hold all candidate hypotheses in the same step. Then, the number of disk accesses is *the number of refinement steps* multiplied by *the number of disk accesses to read all relevant tuples.* As the number of refinement steps is equal to the depth of the fuzzy hypothesis hierarchy, the number of disk accesses becomes $d \times p$, where d and p denote the depth of the fuzzy hypothesis hierarchy and the number of pages containing relevant data, respectively. In fact, even though the cost is acceptable when the depth of the fuzzy hypothesis hierarchy is shallow, there remain some possibilities to reduce it. Currently, we are investigating more efficient method to reduce the number of disk accesses.

3 Discovery of Summaries

The hypothesis refinement method is a powerful mechanism for summary discovery. In order to utilize this hypothesis refinement method for summary discovery, we need several informations including a relevant data specification, a hypothesis template description and names of preferred fuzzy restriction hierarchies.

Though a database has a large amount of data, only some portions of it are relevant to discover summaries. We can specify relevant data by using normal data retrieval queries[21]. A hypothesis template specifies which attributes of the data are to be summarized. Fuzzy restriction hierarchies reflect different subjective cognitions of users. We design a query language satisfying all the above requirements. The formal grammar of it in BNF-like form is given in Figure 5.

In Figure 5, the symbol <table expression> denotes the same grammatical symbol in the standard specification of SQL language[21]. It includes commonly used

<knowledge query> ::=
DISCOVER SUMMARY
IN TERMS OF <attribute list>
USING <hierarchy bindings>
<table expression>
WITH <threshold>

<hierarchy bindings> ::=
<attribute name>:<hierarchy name>
[,<attribute name>:<hierarchy name>...]

<attribute list> ::=
<attribute name> [,<attribute name>...]

Figure 5: The grammar of a query for summaries

FROM-WHERE clauses, which functions as a specification of relevant data. The procedure to discover the summary consists of three steps as follows.

Procedure for summary discovery

1. Retrieve relevant data by executing the part of <table expression>.
2. Construct the hypothesis template in terms of attribute names in <attribute list>.
3. Run **SelectHypo()**.

Now, we present the each discovery step with an example. Suppose that a company database has a relation about employees as in Figure 6. Suppose also that we have a series of fuzzy restrictions on each attribute domain, whose semantics are represented in the form of *semantic relations* as in Figure 8 and Figure 9. The detailed descriptions on defining and manipulating semantic relations can be found in [10]. And a fuzzy restriction hierarchy is depicted in Figure 10 and Figure 11. A query for summaries is given as Figure 7.

NO	NAME	DEPT	MAJOR	AGE	MARRIAGE
1	Lee	plan	commerce	24	unmarried
2	Kim	plan	history	25	unmarried
3	Park	sales	account	30	married
4	Jung	plan	physics	40	unmarried
5	Noh	plan	account	32	unmarried
6	Wang	plan	commerce	26	unmarried
7	Hong	sales	account	55	married
8	Yoon	plan	commerce	22	unmarried
9	Choi	plan	comp sci	26	married
10	Soh	develop	comp sci.	29	unmarrried
11	Moon	plan	history	42	unmarried
12	Kong	plan	account	21	unmarried

Figure 6: An example data relation EMP

DISCOVER SUMMARY
IN TERMS OF MAJOR, AGE
USING MAJOR : H_MAJOR , AGE : H_AGE
FROM EMP
WHERE DEPT = *plan* **OR**
MARRIAGE = *unmarried*
WITH THRESHOLD 0.5

Figure 7: An example summary query

NAME	μ_{LI}	μ_{EC}	μ_{MA}	μ_{EN}
Korean lt	1 0	0 0	0 0	0.0
English lt	1.0	0 0	0.0	0.0
history	0.7	0.4	0 0	0.0
account	0 0	1 0	0 0	0 0
commerce	0 0	1.0	0 8	0.0
management sci.	0.0	0 2	1.0	0 2
computer sci.	0.0	0.0	0 9	1 0
mechanics	0.0	0 0	0.0	1.0
genetic eng.	0 0	0.0	0.0	1 0
physics	0.0	0.0	0.0	0.0
chemistry	0.0	0.0	0 0	0 0

NAME	μ_{NS}	μ_{SO}	μ_{SE}
Korean lt	0.0	1.0	0 0
English lt	0 0	1.0	0.0
history	0.0	1.0	0.0
account	0.0	1 0	0.0
commerce	0 0	1 0	0.8
management sci.	0.0	1.0	0.2
computer sci.	0.0	0.9	1.0
mechanics	0.0	0.0	1 0
genetic eng.	0.9	0.0	1.0
physics	1.0	0.0	1.0
chemistry	1.0	0 0	1.0

Figure 8: An example semantic relation for *MAJOR*

1. Retrieve relevant data

We execute the part of <table expression> to retrieve relevant data. It is to select every tuple whose DEPT is *plan* and MARRIAGE is *unmarried* from EMP.

2. Construct the hypothesis template

The hypothesis template has the form as $(A_1$ *is* $X_1) \wedge \ldots \wedge (A_m$ *is* $X_m)$. The corresponding hypothesis template to the above example is (MAJOR is X) $\wedge$ (AGE is Y).

3. Run SelectHypo() .

After execution of **SelectHypo()**, we have the resulting summaries, depicted in Figure 12. From the result, we conclude that most of unmarried employees in the *plan* department have majored in *sociology*, especially *economics*, and they are *young*.

LOWER	UPPER	$\mu_{very\ young}$	μ_{young}	μ_{medium}
0	20	1.0	1 0	0 0
20	25	0 8	0 9	0 0
25	30	0 2	0 4	0 0
30	35	0 0	0 1	0 2
35	40	0 0	0 0	1.0
40	45	0 0	0 0	0.9
45	50	0 0	0.0	0.2
50	55	0 0	0 0	0.0
55	60	0 0	0 0	0 0
60	∞	0 0	0 0	0 0

LOWER	UPPER	μ_{old}	$\mu_{very\ old}$
0	20	0 0	0.0
20	25	0 0	0.0
25	30	0 0	0 0
30	35	0 0	0 0
35	40	0 0	0 0
40	45	0 0	0 0
45	50	0 1	0 0
50	55	0 4	0 2
55	60	0 9	0 8
60	∞	1 0	1.0

Figure 9: An example semantic relation for *AGE*

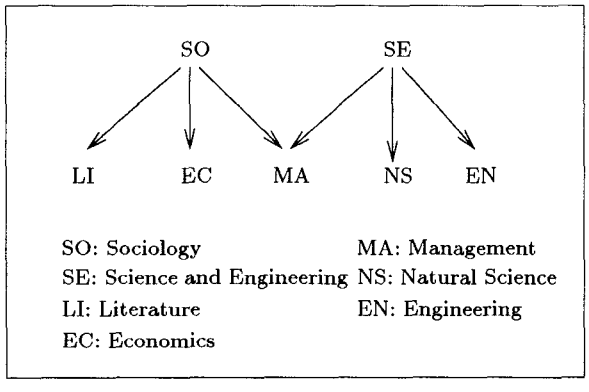

Figure 10: A fuzzy restriction hierarchy *H_MAJOR* on MAJOR

4 Applications of the Discovered Summaries

Summary discovery not only is useful for users who need certain knowledge on databases, but also has many applications if discovered summaries are appropriately maintained. Since the summaries themselves form a useful data set, an intelligent database system should be able to store, renew and access summary data set. In what follows, we present potential applications where the discovered summaries are effectively utilized.

Querying database knowledge

Database knowledge represents the semantic information associated with databases, which includes

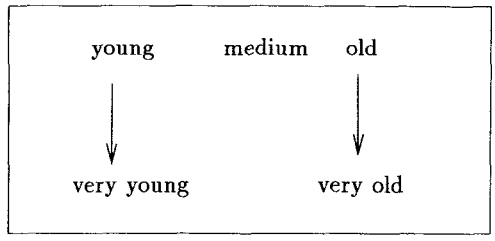

Figure 11: A fuzzy restriction hierarchy *H_AGE* on AGE

	MAJOR	*AGE*	*SD()*
(1)	*SO*	*young*	0.725
(2)	*EC*	*young*	0.650
(3)	*SO*	*very young*	0.613
(4)	*EC*	*very young*	0.562

Figure 12: The result of the summary discovery procedure

deduction rules, integrity constraints, concept hierarchies about data, and especally, general data characteristics[4]. General data characteristics, i.e., data summaries, provide means of answering particular questions about the data and ways of formatting the information to enable an analyst to comprehend the content of the data easily[5]. The facility to summarize databases has much to do with communicating observations about the problem domain in a useful and understandable manner. It also provide a starting point for the ability to make useful inferences from large collections of data. The statement that "Most students in biology department prefer natural science subjects" allow a curriculum planner to make inferences about the viability of opening either "chemistry" or "physics" course in the next term.

Handling query failures

Discovered summaries can also be used for handling query failures. A conventional querying system does not provide any helpful information when a query fails, i.e., a null answer. This may not be serious in some applications, but for applications such as Decision Support Systems and Advice Giving Systems, that need to provide cooperative fashion of interfaces, responses with null answers may not be satisfactory.

Suppose that a user asks a list of employees satisfying the conditions, (*i*) unmarried, (*ii*) works for the *planning* department, (*iii*) young, (*iv*) ***possibly*** majored in *management science*. Then he/she may formulate

the following query in the SQL syntax[21].

SELECT NO, NAME, DEPT
FROM EMP
WHERE MARRIAGE = *unmarried*
AND DEPT = *plan*
AND AGE < 30
AND MAJOR = *management science*

If the query is evaluated on Figure 6, it ends up with a null answer. Generally, this kind of query failure is a frustrating event in the course of interaction with a database management system[13]. If a querying system can provide a measure to help reformulating the query effectively, it is regarded as a cooperative system. Actually, constraints in a data request can be classified into two categories, one is mandatory constraints and the other is optional ones. In the above data request, the first two constraints can be regarded as mandatory and the other two can be regarded as optional. If the distinctions are explicitly expressed in the query, it may be as follows.

SELECT NO, NAME, DEPT
FROM EMP
WHERE MARRIAGE = *unmarried*
AND DEPT = *plan*
AND AGE < 30 : **OPT**
AND MAJOR = *management science* : **OPT**

If the query ends up with the null answer, we can either search the stored summary database or trigger the summary discovery procedure to help the user reformulate the query effectively. The corresponding summary query is already demonstrated in Figure 7. The result of the summary query, presented in Figure 12, says that most unmarried employees in the *plan* department are young and majoring in subjects related to *economics* rather than *management*. Upon the helping result, the user may modify the last constraint in the failed query as "MAJOR = *account* or MAJOR = *commerce*".

Semantic query optimization

Semantic query optimization applies database semantics, integrity constraints and knowledge rules to optimize queries for efficient processing[15]. Since join is the most costly operation among several relational operations, there have been many efforts to decide efficient join ordering. If there are not any other additional aids such as index, relative comparison of relation cardinalities can be used to decide inner and outer relation for nested-loop join[16]. The discovered summaries provide useful guides for estimating cardinalities of join relations.

Suppose that the following binary join query is subject to optimization.

SELECT NO, NAME, DEPT
FROM SUPPLYER, PART
WHERE SUPPLYER.CITY = *Taejon*
AND SUPPLYER.STATUS > 10,000
AND PART.MADEOF = *wood*
AND PART.COLOR = *black*
AND SUPPLYER.PNAME = PART.NAME

Suppose also there are discovered summaries such as "SUPPLYERS.CITY = *Taejon* : SUPPLYERS.STATUS is $\widetilde{about-9000}$ with $SD() = 0.7$" and "PART.MADEOF = *wood* : PART.COLOR is $\widetilde{bright}$ with $SD() = 0.8$". If the cardinarities of two relations SUPPLYER and PART are similar to each other, we can expect that the relation PART is to be reduced significantly after the selection process that precedes the join. In the contrast, the relation SUPPLYER is not to be reduced much. Upon the estimation, we may decide the PART as the inner relation. In distributed query processing environments, the similar information can also be utilized for ordering semi-join sequences.

5 Concluding Remarks

In this paper, we have described a hypothesis refinement method for constructing and evaluating fuzzy hypotheses. The hypothesis refinement method once constructs general hypotheses in terms of the broadest concepts. It evaluates the hypotheses and then refines only strongly supported hypotheses. The refined hypotheses are evaluated again for further refinements. After some stepwise refinements, it comes up with useful specific hypotheses as the results. Based on the hypothesis refinement method, we proposed an effective and robust algorithm to discover linguistic summaries from databases.

The proposed discovery method accomodates both nominal and numerical data through a unified way. It is robust in a sense that some noisy data due to either wrong inputs or genuine exceptions can not affect the major results. As it utilize concept hierarchies with fuzzy boundaries, more closer approximations to real domain knowledge can be exploited. Furthermore, because it results in simple linguistic descriptions of summaries, it is more effective to comprehend a large amount of data. The discovered summaries have many potential applications including querying database knowledge, handling query failures and semantic query optimization.

There remains several open issues to be addressed. Discovered summaries themselves can be regarded as

useful information to be stored and managed. However, since they are dependent on original raw databases, some summaries may be obsolete after considerable changes in the raw databases. Thus, the method to maintain consistencies between the raw database and the derived summary data set should be addressed. Considering a performance aspect, the same tuple is to be visited several times, actully the number of refinement steps. This redundancy should be avoided.

References

[1] W. Frawley, G. Piatetsky-Shapiro and C. Matheus, "Knowledge Discovery in Databases : An Overview", *Knowledge Discovery in Databases*, AAAI Press, pp.1-27, 1991

[2] A. Silberschatz, M. Stonebraker and J. Ullman, "Database Systems: Achievement and Oppertunities", *The Lagunita Report of the NSF Invitational Workshop*, TR-90-22, Dept. of CS, Univ. of Texas at Austin, 1990

[3] P. Cohen and E. Feigenbaum, *The Handbook of Artificial Intelligence*, Vol. 3, William Kaufmann Inc., 1982

[4] J. Han, Y. Cai and N. Cercone, " Knowledge Discovery in Databases: An Attribute-Oriented Approach ", *Proc. the 18th VLDB Conference*, 1992, pp. 547-559

[5] R. Yager, " On Linguistic Summaries of Data", *Knowledge Discovery in Databases*, AAAI Press, 1991, pp.347-363

[6] M. Chen and L. McNamee, " Summary Data Estimation Using Decision Trees", *Knowledge Discovery in Databases*, AAAI Press, 1991, pp.309-324

[7] P. Hoschka and W. Kloesgen, " A Support System for Interpreting Statistical Data", *Knowledge Discovery in Databases*, AAAI Press, 1991, pp.325-366

[8] K. Kaufmann, R. Mitchalski and L. Kerschberg, "Mining for Knowledge in Databases: Goals and General Description of the INLEN System", *Knowledge Discovery in Databases*, AAAI Press, 1991, pp.449-462

[9] T. Anwar, H. Beck and S. Navathe, " Knowledge Mining by Imprecise Querying: A Classification-Based Approach " *Proc. the 8th Conference on Data Engineering*, 1992, pp. 622-630

[10] D.H. Lee, H. Lee-Kwang and M.H. Kim, " A Study on the Fuzzy Selective Relational Algebra ", *Proc. the 2nd International Conference on Fuzzy Logic and Neural Networks*, 1992, pp. 353-356

[11] F. Cuppens and R. Demolombe, " Cooperative Answering: a methodology to provide intelligent access to Databases ", *Proc. of the 2nd Int'l Conf. on Expert Database Systems*, 1988, pp. 333-353

[12] A. Motro, " Extending the Relational Database Model to Support Goal Queries ", *Proc. Int'l Conf. Expert Database Systems*, 1986, pp.129-150

[13] A. Motro, " Query Generalization: A Technique for Handling Query Failure", *Proc. Int'l Workshop Expert Database Systems*, 1984, pp.314-325

[14] F. Corella, S. J. Kaplan, G. Wiederhold and L. Yesil, " Cooperative Responses to Boolean Queries ", *Proc. Int'l Conf. Data Engineering*, 1984, pp. 77-85

[15] U. Chakravarthy, J. Grant and J. Minker, "Logic-Based Approaches to Semantic Query Optimization", *ACM Trans. Database Systems*, Vol.15, No.2, 1990, pp. 162-207

[16] P. Selinger et al, "Access Path Selection in a Relational Database Management System", Proc. Int'l Conf. ACM SIGMOD, 1979

[17] L. Zadeh, " Fuzzy Sets ", *Information and Control*, Vol. 8, 1965, pp. 338-353

[18] L. Zadeh, " A Computational Approach to Fuzzy Quantifiers in Natural Language ", *Comp. and Maths with Appls*, 1983, pp. 149-184

[19] H. Zimmermann, *Fuzzy Set Theory and Its Applications*, Kluwer-Nijhoff Pub., 1985

[20] J. Kacprzyk and A. Ziółkowski, " Database Queries with Fuzzy Linguistic Quantifiers ", *IEEE Trans. on Systems, Man and Cybernetics*, Vol. 16, No. 3, 1986, pp. 474-478

[21] ISO 9075: Information Processing Systems – Database Language SQL, 1989