A Statistical Approach to Incomplete Information in Database Systems

EUGENE WONG
University of California, Berkeley

There are numerous situations in which a database cannot provide a precise answer to some of the questions that are posed. Sources of imprecision vary and include examples such as recording errors, incompatible scaling, and obsolete data. In many such situations, considerable prior information concerning the imprecision exists and can be exploited to provide valuable information for queries to which no exact answer can be given. The objective of this paper is to provide a framework for doing so.

Categories and Subject Descriptors: H.2.1 [**Database Management**]: Logical Design—*data models; schema and subschema*

General Terms: Theory

Additional Key Words and Phrases: Incomplete information, missing values, null values

1. INTRODUCTION

There are numerous situations in which a database cannot provide a precise and unambiguous answer to some of the queries that we wish to pose. The potential sources for the difficulty vary. These include examples such as measurement and recording errors, missing data, incompatible scaling, obsolescence, and data aggregation of one kind or another. Different approaches to this problem have been tried. These range from a consistent way of handling a place-holder "value not known" to Lipski's recent work [3] on dealing with the truth value "possible" in an extended propositional calculus. Although the focus is different, the problem also arises in the artificial intelligence literature (see, e.g., [1]).

In many of the situations where a precise answer cannot be obtained from the database, much more prior information than "value unknown" or "predicate is possibly true" is available. The goal of this paper is to propose a framework wherein such prior information can be effectively exploited. Statistics being the science of handling data in the face of uncertainty, the natural framework for extracting information from an imprecise database is perforce statistical.

The organization of this paper is as follows: first, we enumerate a number of commonly occurring sources of imprecision, and propose a general model that encompasses all of these. Using this model, we restate queries on an imprecise

Author's address: Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA 94720.

database as problems of statistical inference. We then propose a definition for "answers" to a query, and consider the merits of these relative to processing ease and consistency under query transformations. Problems of acquisition and storage of a priori statistical information are of great practical importance, but their consideration is deferred until a follow-up study.

2. SOURCES OF IMPRECISION

We begin with an enumeration of some common sources of imprecision.

(a) *Scale differences.* Here, we are referring to scale differences that cause an ambiguity and not merely a change in units; for example, changing a temperature from degrees Fahrenheit to degrees Celsius is a change in units, but changing temperature from degrees Fahrenheit to one of four values {cold, cool, warm, hot} is a scale change that creates imprecision.
(b) *Missing attributes.* One or more attributes may be absent altogether in a database.
(c) *Combined attributes.* Two or more attributes may get combined in an irreversible way. For example, "cost of labor" and "cost of parts" may get combined into "total cost."
(d) *Missing data.* The value of a given attribute may be missing for some entities but not others. This can be considered a special case of missing attribute by partitioning the set of entities into one consisting of those for which the attribute value is available and one that is not.
(e) *Classification.* Entities may get grouped into classes, and individual attribute values are replaced by class characteristics. For example, instead of recording maximum cruising speed for individual ships, one might record the maximum speed for each of the classes: destroyers, aircraft carriers, and so forth.
(f) *Obsolescence.* The data that are available may be out of date, as, for example, last year's salary or yesterday's ship position.
(g) *Measurement error.* Random errors are often introduced in measurement and recording.
(h) *Data aggregation.* Sometimes the recorded class characteristics are data dependent, for example, the total salary for each department. We call such class charcteristics, *aggregated data.*

3. A MODEL FOR IMPRECISE DATABASES

Consider an *idealized world* represented by a mapping

$$E \xrightarrow{f} V$$

where E is a set of entities and V is a space of values. We assume that all queries are expressible in terms of the schema of the idealized world. The actual database, on the other hand, is an instantiation of a *real-world* schema represented by a mapping

$$E \xrightarrow{h} U$$

where U is the space of observed values. In other words, queries concern f but only h is known.

Consider the following possible relationships between the real and idealized worlds:

(a) There exists a known function g

$$V \xrightarrow{g} U$$

such that $h(e) = g(f(e))$ for all e in E. If g is invertible, then every query on f can be expressed as a query on h, and no problem of imprecision is involved. Hence, we assume that g is not invertible and call imprecision of this type, type-1 imprecision.

(b) For each e, $h(e)$ is a random variable whose distribution depends on the value $f(e)$. We assume that

$$\text{prob}(h(e) = u \mid f(e) = v) = p(u \mid v)$$

is known and independent of e. Furthermore, we assume that for different e's the $h(e)$ are statistically independent, that is,

$$\text{prob}(h(e_i) = u_i,\ i = 1, 2, \ldots, m \mid f(e_i) = v_i,\ i = 1, \ldots, m) = \prod_{i=1}^{m} p(u_i \mid v_i).$$

We call this type-2 imprecision.

The two types of imprecision do not exhaust all possible situations, but they do cover many common sources of imprecision. Of the examples that we have considered the following are of type 1.

Scale Difference. With an augmentation of V as necessary, a difference in scale can always be represented by a noninvertible mapping g.

Missing Attributes. Here g is simply a projection.

Combined Attribute. This case corresponds to U having a lower dimensionality than V.

Class Characteristics. A data-independent class characteristic corresponds to a noninvertible map g. A class constraint such as "maximum speed for each type of vehicle" is reflected in the definition of V.

Of the examples we have considered, "errors," "obsolescence," and "aggregation" are well-modeled by type-2 imprecision. For example,

$$\text{if } f(e) = \text{current salary of } e$$

$$\text{and } h(e) = \text{last year's salary of } e, \text{ then}$$

$$f(e) = h(e) + \epsilon(e)$$

where $\epsilon(e)$ is salary increase of e during the last year. Suppose that

$$\epsilon(e) = \tau(e)h(e),$$

where $\tau(e)$ is independent of h with a distribution $\prod(\cdot)$. Then,

$$p(v \mid u) = \prod\left(\frac{v}{u} - 1\right).$$

Aggregation is in general a complex situation, but even here type-2 imprecision is usually an adequate model. For example, suppose that salaries of those employees in the same department are aggregated so that for a given employee e, the real database yields the dept of e, and the number of employees N and average salary $\bar{S}$ in that department. We can write

$$N(\text{dept}(e))\bar{S}(\text{dept}(e)) = \sum_{e' \in \text{dept}(e)} \text{salary}(e') = \text{salary}(e) + \sum_{\substack{e' \in \text{dept}(e) \\ e' \neq e}} \text{salary}(e'),$$

and it is not unreasonable to consider the sum

$$\sum_{\substack{e' \in \text{dept}(e) \\ e' \neq e}} \text{salary}(e')$$

a random variable, independent of salary (e), but with a distribution that depends on $N(\text{dept}(e))$. Therfore, we can write

$$\begin{aligned} h(e) &= [\text{dept}(e), N(\text{dept}(e)), N(\text{dept}(e))\bar{S}(\text{dept}(e))] \\ &= [\text{dept}(e), N(\text{dept}(e)), \text{salary}(e) + Z], \end{aligned}$$

where Z is a random variable with a distribution p_Z that depends on $N(\text{dept}(e))$. If we take

$$f(e) = [\text{dept}(e), N(\text{dept}(e)), \text{salary}(e)]$$

and denote $v = (v_1, v_2, v_3)$ and $u = (u_1, u_2, u_3)$, then

$$\text{prob}(h(e) = u \mid f(e) = v) = 1(u_1 = v_1)1(u_2 = v_2)p_Z(u_3 - v_3 \mid v_2),$$

where $1(a = b)$ is 1 if $a = b$ and 0 otherwise.

In the preceding example, we had to augment $f(e)$ by the component $N(\text{dept}(e))$ in order to satisfy the condition that, given $f(e)$, the distribution of $h(e)$ is completely determined. There are other instances where augmentation of f would be necessary. For example, suppose that an incompatible scale difference exists between two databases that we wish to amalgamate. In such a case the function f of the idealized world would have to include a scale fine enough to allow both of the actual databases to be defined as functional mappings of f.

To summarize, we have assumed that our prior knowledge concerning the relationship between the real and idealized worlds is either in the form of a distortion function g or in the form of a conditional distribution $p(u \mid v)$. In addition to the relationship between f and h, prior information may also exist for f. Again, we consider two cases.

(a) No known prior information exists for f.
(b) For each e, $f(e)$ is a random variable with a known distribution p_f that does not depend on e. Further, $f(e)$ for different e's are mutually independent.

Note that in case (b) we do not assume that the *components* of $f(e)$ are mutually independent. The distribution of one component may well depend on another. For example, suppose that $E = \{\text{ships}\}$ and $f(e) = (\textit{type}(e), \textit{speed}(e))$. The distribution of speed(e) may well depend on type(e).

Combining the two cases on f with the two cases on g, we have four possible situations regarding *prior knowledge*: We refer to the four cases numerically as indicated.

h \ f	nothing known	random
$h = g(f)$	1	3
random	2	4

4. RETRIEVAL AS A PROBLEM OF INFERENCE

Consider an elementary retrieval query of the form

$$\text{Find } f^{-1}(A) = \{e \in E : f(e) \in A\}$$

for a specified set A in V. Our database consists of $\{h(e), e \in E\}$. Now consider the four cases classified according to the prior information.

Case 1 (f *unknown*, g *known*). Let $g(A) = \{g(v) : v \in A\}$. Since $f(e) \in A$ implies $h(e) = g(f(e)) \in g(A)$, we have

$$h^{-1}(g(A)) \supset f^{-1}(A).$$

The set $h^{-1}(g(A))$ corresponds to the outer limit $\| f^{-1}(A) \|^*$ of Lipski [3].

To get Lipski's inner limit, define

$$U(A) = \{u \in U : g^{-1}(u) \subseteq A\}.$$

Then,

$$h(e) \in U(A) \quad \text{implies} \quad f(e) \in A,$$

so that

$$f^{-1}(A) \supset h^{-1}(U(A))$$

and the right-hand side is Lipski's inner limit $\| f^{-1}(A) \|_*$.

The true answer $f^{-1}(A)$ can be any set between the two limits. Without more information there is little more that one can say.

For the remaining cases the problem of finding $f^{-1}(A)$ can be restated as a problem of hypothesis testing. For each e in E we wish to decide beween

$$H = f(e) \in A$$

and

$$H_0 = f(e) \notin A,$$

and the decision is to be based on our prior knowledge and the observed database $\{h(e), e \in E\}$. Because of our independence assumption, for a given e only $h(e)$ is useful in making the decision, and the problem is one of deciding between H and H_0 using the observation $h(e)$.

Suppose that we restrict ourselves to nonrandomized decisions. That is, if $h(e_1) = h(e_2)$, the decision for e_1 and e_2 is always the same. Then, any decision rule

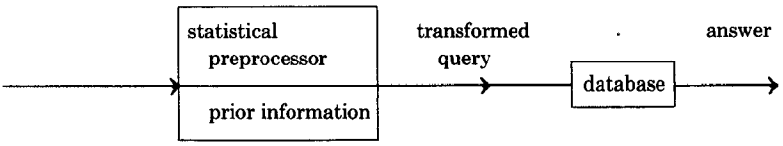
corresponds to a partition of the space U into sets $\tilde{A}$ and $U - \tilde{A}$, and we decide for H iff $h(e) \in \tilde{A}$. The set $\| f^{-1}(A) \| = \{e$: we decide H is true$\}$ is expressible as $h^{-1}(\tilde{A})$ and we have the following:

PROPOSITION 1 (SEPARATION PRINCIPLE). *For any nonrandomized decision rule, the approximate answer* $\| f^{-1}(A) \|$ *to a query* $f^{-1}(A)$ *is of the form*

$$\| f^{-1}(A) \| = h^{-1}(\tilde{A}),$$

where $\tilde{A}$ *depends only on* A *and the prior distributions.*

The separation principle, though little more than an observation on our model, has major significances in terms of processing. First, a query on data that we do not have has been transformed into one on data that we do have. Hence, the burden of coping with imprecision is confined to one of query transformation. Second, to perform the transformations requires only prior knowledge and not data. The following processing arrangement is suggested by the separation principle:



The next question is how do we find good decision rules? The answer depends on the specific prior information that we possess.

Case 2 (*h random, f unknown*). For each e the distribution of $h(e)$ belongs to one of two families:

$$\{p(u \mid v), v \in A\} \quad (H)$$

$$\{p(u \mid v), v \notin A\} \quad (H_0).$$

We have to decide for each value u which is the case. The situation here is one of testing one composite hypothesis against a composite alternative. A decision rule often used in such a situation is the generalized likelihood ratio test [2].

Define the likelihood ratio by

$$L(u, A) = \frac{\max_{v \in A} p(u \mid v)}{\max_{v \notin A} p(u \mid v)}.$$

Intuitively, if $L(u, A) \gg 1$, then H is more likely, and vice versa. The generalized likelihood ratio test is a one-parameter family of decision rules of the form

$$\text{decide } H \text{ is true} \quad \text{iff} \quad L(h(e), A) \geq \alpha.$$

The parameter α is adjusted according to how one feels about the two types of errors:

miss (decide H_0 when H is true)

false alarm (decide H when H_0 is true).

Increasing α will reduce false alarm at the expense of having more misses.

We denote the approximate answer by

$$\| f^{-1}(A) \|_\alpha = \{e : L(h(e), A) \geq \alpha\}.$$

Cases 3 and 4 (f random). Case 3 can be considered a special case of Case 4 with

$$\begin{aligned} \text{prob}(h(e) = u \mid f(e) = v) &= 1, \quad \text{if} \quad u = g(v) \\ &= 0 \qquad\qquad\quad \text{otherwise} \end{aligned}$$

and we do not need to consider it separately.

Upon observing $h(e)$, we can summarize our knowledge concerning $f(e)$ by the a posteriori distribution

$$p(v \mid u) = \text{prob}(f(e) = v \mid h(e) = u).$$

The "minimum cost Bayes decision rule" is given as follows.

PROPOSITION 2. *Suppose that the cost for false alarm is α, and for miss $1 - \alpha$. Then the average cost is minimized by the decision rule*

$$\textit{decide} \quad H \quad \textit{iff} \quad p(A \mid h(e)) \geq \alpha,$$

where

$$p(A \mid u) = \sum_{v \in A} p(v \mid u).$$

PROOF. For a given $h(e)$ one can decide in one of two ways. If we decide for H, the cost is that of false alarm α and the probability of having a false alarm is $1 - p(A \mid h(e))$. Similarly, the weighted cost if we decide for H_0 is $(1 - \alpha)p(A \mid h(e))$. Hence, the decision rule that minimizes average cost is to choose the smaller of the two

$$\alpha[1 - p(A \mid h(e))], \qquad (1 - \alpha)p(A \mid h(e)),$$

or

$$\begin{aligned} &\text{decide} \quad H \quad \text{if} \quad p(A \mid h(e)) \geq \alpha \\ &\text{decide} \quad H_0 \quad \text{otherwise.} \end{aligned}$$

Q.E.D.

The family of sets

$$\| f^{-1}(A) \|_\alpha = \{e : p(A \mid h(e)) \geq \alpha\}$$

decreases with increasing α, and represents a family of approximations to $f^{-1}(A)$. The parameter α is adjusted according to the relative cost that is assigned to "false alarm." In practice, there is no need to distinguish between a "logically impossible event" and an event with 0 probability. With this assumption, we can then identify Lipski's bounds as follows

$$\begin{aligned} \| \quad \|_* &= \| \quad \|_1 \\ \| \quad \|^* &= \| \quad \|_{0^+}. \end{aligned}$$

In most cases where probabilities are available, these limiting bounds are not useful approximations.

Since the decision rule given by Proposition 2 is nonrandomized, the separation principle applies. If we write

$$\tilde{A}_\alpha = \{u \in U : p(A \mid u \geq \alpha)\}$$

then

$$\| f^{-1}(A) \|_\alpha = h^{-1}(\tilde{A}_\alpha)$$

and the original query has been modified into a query on the actual database.

Summarizing, for a query of the form

$$\text{Find}\{e : f(e) \in A\}$$

we propose the following as approximate answers:

$$\text{Case (2):} \quad \| f^{-1}(A) \|_\alpha = \{e : L(h(e), A) \geq \alpha\}$$

$$\text{Cases (3) and (4):} \quad \| f^{-1}(A) \|_\alpha = \{e : p(A \mid h(e)) \geq \alpha\}.$$

In each of these cases the separation principle applies and we can write

$$\| f^{-1}(A) \|_\alpha = h^{-1}(\tilde{A}_\alpha)$$

with

$$\tilde{A}_\alpha = \begin{cases} \{u = L(u, A) \geq \alpha\} \\ \{u = p(A \mid u) \geq \alpha\}. \end{cases}$$

5. AN EXAMPLE

Let E be a collection of ships, each identified by name, and let the idealized database consist of

$$f(e) = (\text{type}(e), \text{speed}(e), \text{current location}(e)).$$

The space V is defined by

$$V_{\text{type,speed}} = \{(\text{carrier}, [20, 30]), (\text{sub}, [25, 40])\}$$

$$V_{\text{location}} = \{\text{Atlantic, Pacific, Indian, Med}\}$$

The actual database consists of

$$h(e) = (\text{type}(e), \text{last week's location}(e)).$$

Suppose that our prior knowledge can be summarized by the probabilities

$$p(\text{current location} \mid \text{LWL}), \qquad (\text{LWL} = \text{last week's location})$$

and

$$p(\text{speed} \mid \text{type})$$

as indicated in Figures 1 and 2.

Now consider the following query:

q: Find all ships in Mediterranean with speed ≥ 30 knots.

Fig. 1. p(current | LWL): M, Mediterranean Sea; I, Indian Ocean; P, Pacific Ocean; A, Atlantic Ocean.

current	LWL: A	LWL: P	LWL: I	LWL: M
M	0.1	0	0.2	0.8
I	0	0.15	0.8	0.1
P	0.1	0.8	0	0
A	0.8	0.05	0	0.1

a in knots	carrier	sub
20	0	0
25	0.4	0
30	0.1	0.2
35	1	0.8
40	1	1

Fig. 2. p(speed $< a$ | type).

The probability

$$p(\text{speed} \geq 30, \text{loc} = \text{Med} \mid \text{type}, \text{LWL})$$

can be easily computed, and we find

type = carrier

type = sub

$p = 0$ for all LWL

LWL	A	P	I	M
p	0.08	0	0.16	0.64

Therefore, $\tilde{A}_\alpha$ is given by

$$\begin{aligned}
\tilde{A}_\alpha &= \varnothing && \text{for} \quad \alpha > 0.64 \\
&= (\text{type} = \text{sub}, \text{LWL} = \text{Med}) && 0.16 < \alpha \leq 0.64 \\
&= (\text{type} = \text{sub}, \text{LWL} \in \{I, \text{Med}\}) && 0.08 < \alpha \leq 0.16 \\
&= (\text{type} = \text{sub}, \text{LWL} \neq \text{Pacific}) && 0 < \alpha \leq 0.08
\end{aligned}$$

All this has been determined without reference to the actual database.

6. EXTRACTING VALUES AND PREPROCESSING

Thus far, we have dealt only with queries that select sets, but not those that extract values. Now, consider a query of the form

$$q\text{: Find} \quad F[f(e)] \quad \text{for all} \quad e \in f^{-1}(A)$$

where A is a specified set in V, and F is a specified function: $V \rightarrow V'$. A typical example for F is a projection operator. Now, there are two sources of uncertainty: $f^{-1}(A)$ and values of f.

Example. Consider the example in Section 5 and a query of the form

q: Find the speed of each ship in the Mediterranean

$A = \{(\text{type}, \text{speed}, \text{loc}): \text{loc} = \text{Med}\}$

$F((\text{type}, \text{speed}, \text{loc})) = \text{speed}.$

We propose to deal with queries of this type by separating them into (a) finding $f^{-1}(A)$, and (b) finding $F[f(e)]$ for each qualifying e. The first point having been dealt with at length, we now address the second. The problem is to find an approximation to $F[f(e)]$ for each e in a specified set S.

We confine ourselves to the three cases for which some prior distribution is known as classified in Section 3. For these cases define [2]

(Case 2): $\hat{f}(e)$ = maximum likelihood estimate of $f(e)$ given $h(e)$

(Cases 3 and 4): $\hat{f}(e)$ = a posteriori most likely estimate of $f(e)$ given $h(e)$.

We then take as an approximation to $F[f(e)]$

$$\widehat{F[f(e)]} = F[\hat{f}(e)].$$

THEOREM 1. *Let $\hat{f}$ and $\widehat{F[f(e)]}$ be as defined. Then, (a) $\{\hat{f}(e), e \in E\}$ is a query independent modification of the database $\{h(e), e \in E\}$, and (b) there exists a data-independent function $\hat{F}$ so that*

$$\widehat{F[f(e)]} = \hat{F}[h(e)].$$

PROOF. Define $\hat{v}(u)$ as follows:

$$\text{(Case 2):} \quad \max p_g(u \mid v) = p_g(u \mid \hat{v}(u)), \qquad v \in V$$

$$\text{(Cases 3 and 4):} \quad \max p(v \mid u) = p(\hat{v}(u) \mid u), \qquad v \in V$$

where

$$p_g(u \mid v) = \text{prob}(g(v) = u)$$

$$p(v \mid u) = \text{prob}(f(e) = v \mid h(e) = g(f(e)) = u)$$

are defined as in Sections 3 and 4. The function $\hat{v}(u)$, $u \in U$, depends only on the prior distributions, but on neither the query nor the data. By the definition of the maximum likelihood and a posteriori most probable estimators, we have

$$\hat{f}(e) = \hat{v}(h(e))$$

and part (a) follows. Now take

$$\hat{F}(u) = F[\hat{v}(u)], \qquad u \in U;$$

then part (b) of the theorem follows. Q.E.D.

Remarks. Part (a) asserts that one can *preprocess* the data, changing $\{h(e), e \in E\}$ into $\{\hat{f}(e), e \in E\}$. Part (b) is a "separation theorem." To evaluate $F[f(e)]$, we can transform F into $\hat{F}$ in a data-independent way and evaluate $\hat{F}[h(e)]$.

The idea of *preprocessing* is intuitively appealing. Indeed, this is probably what is currently done. When there is imprecision or when data do not quite fit, one applies a small correction. The question is, "Is it a good idea?" The answer is, in general, "No." There are at least two serious objections to preprocessing. First, preprocessing violates the separation principle: If the idealized world and the real world do not agree, introduce something to mediate between them, but do not tamper with the data. Second, preprocessing works for extraction of values,

but not always for retrieval. That is, the approximation $\|f^{-1}(A)\|_\alpha$, as defined in the last section, is, in general, not expressible in terms of the preprocessed data $\{\hat{f}(e), e \in E\}$. An exception is when the data quality is "good" and retrieval is restricted to "equality on value." The following theorem makes this precise.

THEOREM 2. *For each u choose unique* $\hat{v}(u)$ *such that*

$$p(\hat{v}(u) \mid u) = \max p(v \mid u), \qquad v \in V$$

and denote

$$\prod(u) = p(\hat{v}(u) \mid u).$$

Then, for $\alpha > \frac{1}{2}$

$$\|f^{-1}(\{v_0\})\|_\alpha = \{e \in E : \hat{v}(h(e)) = v_0 \quad \textit{and} \quad \prod(h(e)) \geq \alpha\}.$$

PROOF. Suppose that $\hat{v}(h(e)) = v_0$ and $\prod(h(e)) \geq \alpha$. Then,

$$\begin{aligned} \text{prob}(f(e) = v_0 \mid h(e)) &= p(v_0 \mid h(e)) \\ &= p(\hat{v}(h(e)) \mid h(e)) \\ &= \prod(h(e)) \geq \alpha. \end{aligned}$$

Hence, for any α

$$\|f^{-1}(\{v_0\})\|_\alpha \supset \{e : \hat{v}(h(e)) = v_0 \text{ and } \prod(h(e)) \geq \alpha\}.$$

Conversely, suppose that $e \in \|f^{-1}(\{v_0\})\|_\alpha$. Then,

$$p(v_0 \mid h(e)) \geq \alpha$$

and

$$\max_v p(v \mid h(e)) \geq \alpha.$$

If $\alpha > \frac{1}{2}$, there can be at most one v such that $p(v \mid h(e)) \geq \alpha$. Hence

$$\hat{v}(h(e)) = v_0 \quad \text{and} \quad \prod(h(e)) \geq \alpha$$

and the proof is complete. Q.E.D.

Remarks

(a) If we replace $\{h(e), e \in E\}$ by $\{\hat{f}(e), p(e)\}$, where $\hat{f}(e) = \hat{v}(h(e))$ and $p(e) = \prod(h(e))$, then for $\alpha > \frac{1}{2}$

$$\|f^{-1}(\{v_0\})\|_\alpha = \{e : \hat{f}(e) = v_0 \quad \text{and} \quad p(e) \geq \alpha\}$$

so that we would no longer need the original data $h(\,)$, or the prior distributions.

(b) The restriction $\alpha > \frac{1}{2}$ is intuitively reasonable, since we would not expect preprocessing to work unless the data were reasonably "clean."

7. COMBINED QUERIES

Thus far, we have only dealt with atomic queries. The questions are, "What happens when queries are combined, for example, under Boolean operations?"

"Can the approximate answer to the combined query be expressed in terms of the approximate answers of its components?" These are the same questions that were posed in Lipski [3] for the upper and lower bounds that he introduced. Our treatment of this topic is not yet complete. Here, we present some results on the two most frequently occurring operations: *conjunction* and *existential quantification*. These results are limited to Cases (3) and (4) where the a posteriori distribution of $f(e)$ given $h(e)$ is known.

Consider a query of the form: Find $\{e : f(e) \in A \cap B\}$. Since

$$f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B)$$

we can normally process the two components in the conjunction one at a time. This possibility is extensively exploited in query-processing algorithms, especially where data are dispersed [5]. The question that we shall consider here is whether a conjunctive query remains conjunctive when imprecision is involved.

The specific question is

$$\| f^{-1}(A \cap B) \|_\alpha \stackrel{?}{=} \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\alpha$$

or, equivalently,

$$(A \cap B)_\alpha \stackrel{?}{=} \tilde{A}_\alpha \cap \tilde{B}_\alpha.$$

The answer is an immediate "no!" This may appear to severely limit our ability to decompose conjunctive queries when imprecision is involved. However, the following theorem shows that this need not be the case.

THEOREM 3. *The approximations* $\| f^{-1}(A) \|_\alpha$ *defined in Section* 4 *for Cases* (3) *and* (4) *satisfy the following relationships under intersection*:

$$\| f^{-1}(A \cap B) \|_{\alpha+\beta-1} \supset \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\beta \supset \| f^{-1}(A \cap B) \|_{\max(\alpha,\beta)} \tag{7.1}$$

$$\begin{aligned} \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\alpha &\supset \| f^{-1}(A \cap B) \|_\alpha \\ &\supset \| f^{-1}(A) \|_{(1+\alpha)/2} \cap \| f^{-1}(B) \|_{(1+\alpha)/2}. \end{aligned} \tag{7.2}$$

If for every e, *prob* $(f(e) \in B) = 0$ *or* 1, *then*

$$\| f^{-1}(A \cap B) \|_\alpha = \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_1. \tag{7.3}$$

If A *and* B *are conditionally independent given the observation, that is,* $p(A \cap B \mid u) = p(A \mid u)p(B \mid u)$ *for all* $u \in U$, *then*

$$\| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\beta \subset \| f^{-1}(A \cap B) \|_{\alpha\beta}. \tag{7.4}$$

PROOF. We begin with the elementary equality

$$\begin{aligned} \text{prob}(f(e) \in A \cup B) = \text{prob}(f(e) \in A &+ \text{prob}(f(e) \in B) \\ &- \text{prob}(f(e) \in A \cap B), \end{aligned}$$

which reflects the fact that probability is additive for disjoint events. Since any probability is bounded from above by 1, we have

$$1 \geq \text{prob}(f(e) \in A) + \text{prob}(f(e) \in B) - \text{prob}(f(e) \in A \cap B)$$

or

$$\text{prob}(f(e) \in A \cap B) \geq \text{prob}(f(e) \in A) + \text{prob}(f(e) \in B) - 1.$$

It follows that

$$\begin{aligned} & e \in \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\beta \\ & \Leftrightarrow \text{prob}(f(e) \in A) \geq \alpha \quad \text{and} \quad \text{prob}(f(e) \in B) \geq \beta \\ & \Rightarrow \text{prob}(f(e) \in A \cap B) \geq \alpha + \beta - 1 \\ & \Leftrightarrow e \in \| f^{-1}(A \cap B) \|_{\alpha+\beta-1}. \end{aligned}$$

We have proved the left half of (7.1). Taking $\alpha = \beta$ and making a change of parameter, we have also proved the right half of (7.2).

The right half of (7.1) is proved by observing that since $A \cap B$ is contained in both A and B,

$$\text{prob}(f(e) \in A \cap B) \leq \min(\text{prob}(f(e) \in A), \text{prob}(f(e) \in B)).$$

Hence,

$$\begin{aligned} \text{prob}(f(e) \in A \cap B) \geq \max(\alpha, \beta) \Rightarrow & \text{prob}(f(e) \in A) \geq \max(\alpha, \beta) \geq \alpha, \\ & \text{and} \\ & \text{prob}(f(e) \in A) \geq \max(\alpha, \beta) \geq \beta \end{aligned}$$

and the right half of (7.1) is proved. The left half of (7.2) follows by setting $\alpha = \beta$.

If $\text{prob}(f(e) \in B) = 0$ or 1 for every e, then for each e

$$\begin{aligned} \text{prob}(f(e) \in A \cap B \mid h(e)) &= 0, && \text{if} \quad \text{prob}(f(e) \in B) = 0 \\ &= \text{prob}(f(e) \in A \mid h(e) && \text{if} \quad \text{prob}(f(e) \in B) = 1. \end{aligned}$$

Hence, (7.3) follows.

Finally, if $p(A \cap B \mid u) = p(A \mid u)p(B \mid u)$ for every $u \in U$, then

$$\text{prob}(f(e) \in A \cap B \mid h(e)) = \text{prob}(f(e) \in A \mid h(e))\text{prob}(f(e) \in B \mid h(e)).$$

Hence,

$$\begin{aligned} e \in \| f^{-1}(A) \|_\alpha \cap \| f^{-1}(B) \|_\beta \Leftrightarrow & \text{prob}(f(e) \in A \mid h(e)) \geq \alpha \\ & \text{and} \\ & \text{prob}(f(e) \in B \mid h(e)) \geq \beta \\ \Rightarrow & \text{prob}(f(e) \in A \cap B \mid h(e)) \geq \alpha\beta \\ \Rightarrow & e \in \| f^{-1}(A \cap B) \|_{\alpha\beta}. \end{aligned}$$

Q.E.D

Remarks

(a) Suppose that for some α and β

$$(A \cap B)_{\alpha+\beta-1} = (A \cap B)_{\max(\alpha,\beta)}.$$

Then equality obtains in (7.1) and exact decomposition of the intersection obtains. Observe that this condition is verifiable in terms of the prior information alone and does not involve data.

(b) Equation (7.3) allows one to decouple the portion of a conjunctive query that references exact data from that which references imprecise data, thereby limiting the effect of imprecise data on processing.

Example. Consider the example of the last section, and let A = "speed ≤ 30" and B = "loc = Med." We found that $\tilde{A}_\alpha$ remained constant for $0.16 < \alpha \leq 0.64$. Taking $\alpha = \beta = 0.64$ in (7.1), we get

$$\| f^{-1}(A \cap B) \|_{0.28} \supset \| f^{-1}(A) \|_{0.64} \cap \| f^{-1}(B) \|_{0.64} \supset \| f^{-1}(A \cap B) \|_{0.64}.$$

Since the outer limits are equal, we have equality in this case.

Even when perfect decomposition is not possible, (7.1) and (7.2) allow us to use the answers from decomposed pieces with some measure of confidence. This is especially true when imprecision is not severe and one demands a high degree of confidence in the answer. For such cases, α and β would be taken to be near 1 and $(\alpha + \beta - 1)$ does not differ much from $\max(\alpha, \beta)$.

Suppose that E is a product space $E_1 \times E_2$ and consider a query of the form

$$\text{q: Find } \{e_1 \in E_1 : \exists e_2 \in E_2 f(e_1, e_2) \in A\},$$

where $\exists$ is the quantifier "there exists." This set can be rewritten in two ways as follows:

$$\{e_1 \in E_1 : \exists e_2 \in E_2 f(e_1, e_2) \in A\} = \bigcup_{e_2 \in E_2} \{e_1 \in E_1 : f(e_1, e_2) \in A\}$$

$$= \text{project}_1 \{(e_1, e_2) : f(e_1, e_2) \in A\}.$$

Each of the latter two forms suggests a natural answer to the query that is consistent with our proposed answer for the basic queries. The question is, "Are they equal and are they reasonable?"

THEOREM 4. *Let f_{e_2} denote the function $E_1 \to V$ defined by*

$$f_{e_2}(e_1) = f(e_1, e_2).$$

Let $\| f^{-1}(A) \|_\alpha$ be defined by

$$\| f^{-1}(A) \|_\alpha = \{e : prob(f(e) \in A) \geq \alpha\}.$$

Then, (a)

$$\bigcup_{e_2 \in E_2} \| f_{e_2}^{-1}(A) \|_\alpha = project_1 \| f^{-1}(A) \|_\alpha$$

$$= \{e_1 \in E_1 : \exists e_2 \, prob(f(e_1, e_2) \in A \mid h(e_1, e_2)) \geq \alpha\}.$$

(b) There exists $\tilde{A}_\alpha$ such that the common quantity in (a) is expressible as

$$\{e_1 \in E_1 : \exists e_2 h(e_1, e_2) \in \tilde{A}_\alpha\}.$$

PROOF

$$e_1 \in \bigcup_{e_2 \in E_2} \| f_{e_2}^{-1}(A) \|_\alpha \Leftrightarrow \exists e_2 : e_1 \in \| f_{e_2}^{-1}(A) \|_\alpha$$

$$\Leftrightarrow \exists e_2 : \text{prob}(f(e_1, e_2) \in A \mid h(e_1, e_2)) \geq \alpha.$$

Similarly,

$$e_1 \in \text{project}_1 \| f^{-1}(A) \|_\alpha \Leftrightarrow \exists e_2 : (e_1, e_2) \in \| f^{-1}(A) \|_\alpha$$

$$\Leftrightarrow \exists e_2 : \text{prob}(f(e_1, e_2) \in A \mid h(e_1, e_2)) \geq \alpha,$$

and part (a) is proved.

From the results of Section 4, we know that there exists $\tilde{A}_\alpha$ such that

$$\| f^{-1}(A) \|_\alpha = h^{-1}(\tilde{A}_\alpha).$$

Hence,

$$\{e_1 : \exists e_2 (e_1, e_2) \in \| f^{-1}(A) \|_\alpha\} = \{e_1 : \exists e_2 (e_1, e_2) \in h^{-1}(\tilde{A}_\alpha)\}$$
$$= \{e_1 : \exists e_2 h(e_1, e_2) \in (\tilde{A}_\alpha)\}$$

and the proof is complete. Q.E.D.

Part (a) of Theorem 4 supports the claim of $\text{project}_1 \| f^{-1}(A) \|_\alpha$ as an answer to the existential query, and part (b) adds further strength to that claim. Part (a) shows consistency under alternative forms of processing, since

$$\bigcup_{e_2 \in E_2} \| f_{e_2}^{-1}(A) \|_\alpha$$

would result from tuple-substitution for e_2 and $\text{project}_1 | f^{-1}(A) |_\alpha$ would result from processing a restriction on $E_1 \times E_2$ and then projecting. Part (b) is a strong separation theorem. It says that not only can we transform the query in a data-independent way, but that the resulting query is of the same existential form as the original.

Part (a) of Theorem 4 also yields an interpretation for the proposed answer for an existential query q, namely,

$$\| q \|_\alpha = \{e_1 : \exists e_2 \, \text{prob}(f(e_1, e_2) \in A \mid h(e_1, e_2)) \geq \alpha\}.$$

How reasonable is this as an answer? In some situations, it can be argued that this answer is too restrictive. An alternative answer might be

$$\| q \|^\alpha = \{e_1 : \text{prob}(\exists e_2 : f(e_1, e_2) \in A \mid \text{observed data}) \geq \alpha\}.$$

For the same α $\| q \|^\alpha$ is in general much larger than $\| q \|_\alpha$. How would one process the actual database to get $\| q \|^\alpha$? The answer is given by the following theorem:

THEOREM 5. *Assume that given the observation* $\{h(e), e \in E\}$, $f(e)$ *for different e's are statistically independent, and define*

$$F(u, A) = 1 - prob(f(e) \in A \mid h(e) = u)$$
$$= 1 - \sum_{v \in A} p(v \mid u).$$

Then

$$\| q \|^\alpha = \left\{ e_1 : 1 - \prod_{e_2 \in E_2} F(h(e_1, e_2), A) \geq \alpha \right\}$$

$$= \left\{ e_1 : - \sum_{e_2 \in E_2} ln \, F(h(e_1, e_2), A) \geq - ln(1 - \alpha) \right\}$$

$$= \{e_1 : 1 - prob(\forall e_2 f(e_1, e_2) \notin A \mid h(\cdot)) \geq \alpha\}.$$

Table I

s	j	prob(s.city = j.city | observed data)
A	1	0.2
A	2	0.2
A	3	0.2
A	4	0.2
B	3	0.5

PROOF. Under the independence assumption

$$\text{prob}(\forall e_2 f(e_1, e_2) \notin A \mid h(\cdot)) = \prod_{e_2} \text{prob}(f(e_1, e_2) \notin A \mid h(e_1, e_2))$$
$$= \prod_{e_2} F(h(e_1, e_2), A)$$

so that the first equality is proved. The second follows by elementary rearrangements. Q.E.D.

Remark. Theorem 5 is in the form of a *weak* separation theorem. The original query q is transformed into a query on the actual database in a data-independent way, but form is not preserved. The second expression for $\| q \|^\alpha$ in the theorem shows that while q involved only restriction and projection, the transformed query involves aggregation and thus is considerably more difficult to process.

The two proposed solutions to q have rather different interpretations, and can represent very different sets. For a comparison use the approximation $-\ln(1 - x) = x$ in the second expression for $\| q \|^\alpha$ in Theorem 4 and get

$$\| q \|^\alpha = \left\{ e_1 : \sum_{e_2 \in E_2} \text{prob}(f(e) \in A \mid h(e)) \geq \alpha \right\};$$

on the other hand, we can write

$$\| q \|_\alpha = \{ e_1 : \exists e_2 \ \text{prob}(f(e) \in A \mid h(e)) \geq \alpha \}$$
$$= \left\{ e_1 : \max_{e_2} \text{prob}(f(e) \in A \mid h(e)) \geq \alpha \right\}.$$

It is evident that how much these answers differ depends on the number of e_2's for which prob($f(e_1, e_2) \in A \mid h(e_1, e_2)$) is nonzero for each e_1.

As an example, let E_1 = {suppliers}, E_2 = {projects}, and $f(e_1, e_2)$ = (city(e_1), city(e_2)). Consider the query

$$q\text{: Find} \quad \{ s \text{ in } E_1 : \exists j \text{ in } E_2 (j.\text{city} = s.\text{city}) \}.$$

Now suppose that after processing the actual database, we find the values shown in Table I. The largest value of α for which $\| q \|_\alpha$ is nonempty is 0.5 and

$$\| q \|_\alpha = \{\text{B}\} \qquad \text{for} \quad 0.2 < \alpha \leq 0.5.$$

On the other hand, the largest value of α for which $\| q \|^\alpha$ is nonempty is $1 - (1 - 0.2)^4$ or 0.5904 and

$$\| q \|^\alpha = \{\text{A}\} \qquad \text{for} \quad 0.5 < \alpha \leq 0.5904.$$

Table II

	$0 < \alpha < 0.2$	$0.2 < \alpha < 0.5$	$0.5 < \alpha < 0.5904$	$0.5904 < \alpha < 1$
$\|\| q \|\|^{\alpha}$	A,B	A,B	A	∅
$\|\| q \|\|_{\alpha}$	A,B	B	∅	∅

The complete results are summarized in Table II. For supplier A no project to which one can point is likely to be in the same city as A, but the number of projects that can be in the same city is sufficiently large to make the existence of at least one quite probable.

8. STATISTICAL PROCESSING THROUGH VIEW SUPPORT

In database management systems (particularly relational ones) with facility for supporting views, such facility can be used to support statistical processing. Basically, the idea is to treat the prior information on the distributions as an additional database. A query on the idealized world is then transformed by view mapping into a query that spans both the real database $\{h(e), e \in E\}$ and the statistical database that contains the prior information. It is important to note the difference between such a procedure and the query transformation procedure suggested by the separation principle. The query transformation involved in view mapping is much simpler, but the resulting query is more complex. In effect, one is using the view-support and query-processing facilities that normally exist to achieve the computation needed to transform an ideal-world query into a real-world query.

We shall restrict our attention to the relational system INGRES, but the results are easily adapted to other relational systems of comparable power. Define a statistical subdatabase consisting of one or more relations of the form

distribution (ideal attribute, real attribute, probability)

Each tuple in this relation represents one instance of $p(v \mid u)$ in the form $(v, u, p(v \mid u))$. For example, the probabilities of Figure 1 would appear as in Figure 3. Now, suppose that the ideal-world schema consists of one or more relations of the form

rel-ideal (eid, v)

where v stands for one or more attributes and eid is the identifier for e. A tuple from such a relation (if one were available) would be an instance of $(e, f(e))$. For the example of Section 5, we would have an ideal-world relation:

ship-ideal (shipid, type, speed, current location)

Similarly, a real-world schema would consist of one or more relations of the form

rel-actual (eid, u)

a tuple being an instance of $(e, h(e))$. Continuing with the example of Section 5, we would have

shipdata (shipid, type, location-last-week)

distribution-location

current	last week	prob
M	M	0.8
M	I	0.1
M	A	0.1
I	I	0.8
I	P	0.15
I	M	0.1
P	P	0.8
P	A	0.1
A	A	0.8
A	M	0.1
A	P	0.05

Fig. 3. Distribution of location: M, Mediterranean Sea; I, Indian Ocean; P, Pacific Ocean; A, Atlantic Ocean.

Let A be a set in V of the form

$$A = \{v \in V : v * a\}$$

where $*$ is a comparison operator and a is a constant. A query $f^{-1}(A)$ would be expressed in QUEL [4] as

range of x is rel-ideal
retrieve into result (x.eid)
where $x.v * a$

Now rel-ideal is not a real relation. But for each α, $\| f^{-1}(A) \|_\alpha$ is obtained by running the following QUEL query

range of x is rel-actual
range of y is distribution
retrieve into approximation (x.eid)
where ($x.u = y.u$)
and sum (y.prob by $y.u$ where $y.v * a$) $\geq \alpha$

If the comparison operator $*$ is equality, the query for $\| f^{-1}(A) \|_\alpha$ can be expressed as

range of x is view-α
retrieve into approximation (x.eid)
where $x.v = a$

The view relation view-α is defined by

range of x is rel-actual
range of y is distribution
define view-α(x.eid, $y.v$)
where ($x.u = y.u$)
and (y.prob $\geq \alpha$)

Consider the example of Section 5 once again. Define a view relation

location-α(shipid, current location)

by

range of x is shipdata
range of y is distribution-location
define view location-α(x.shipid, y.current)

where (x.location-last-week = y.last-week)
and(y.prob $\geq \alpha$)

The approximation $\| \text{ships currently in "Med"} \|_\alpha$ would then be represented by executing the view query

range of s is location-α
retrieve into approximation (s.shipid)
where s.current = "Med"

9. CONCLUSION

A large number of theoretical issues and problems are suggested by the preliminary analysis that we have undertaken. For example, the existence of replicated data, especially in distributed systems, suggests the use of redundance to reduce uncertainty. However, the cost of using more than one copy is large and must be kept to a minimum by strategies provided by sequential analysis. Another issue concerns how the a priori distribution information is to be acquired. In some cases it must be done empirically by sampling. However, in many interesting cases (e.g., obsolescence) the distributions can be obtained by modeling the process by which the data ambiguity is introduced. However, we believe that the purely theoretical speculation should await some attempts at implementation for a major application where problems of incomplete information actually arise.

ACKNOWLEDGMENT

Much of the work reported here was done at the Computer Corporation of America. I am particularly grateful to Dr. John M. Smith for suggesting the problem and for many useful discussions.

REFERENCES

1. BARROW, H.G., AND TENENBAUM, J.M. MSYS: A system for reasoning about scenes. AI Center Tech. Note 121, SRI International, Menlo Park, Calif., 1976.
2. CHERNOFF, H., AND MOSES, L. *Elementary Decision Theory.* Wiley, New York, 1959.
3. LIPSKI, JR., W. On semantic issues connected with incomplete information databases. *ACM Trans. Database Syst. 4*, 3 (Sept. 1979), 262–296.
4. STONEBRAKER, M., WONG, E., KREPS, P., AND HELD, G. The design and implementation of INGRES. *ACM Trans. Database Syst. 1*, 3 (Sept. 1976), 189–222.
5. WONG, E. Retrieving dispersed data from SDD-1: A system for distributed databases. In Proc. 1977 Berkeley Workshop on Distributed Data Management and Computer Networks, Lawrence Berkeley Laboratory, Univ. of California, Berkeley, Calif., May 1977.

Received November 1980; revised July 1981; accepted July 1981