Uncertainty Mamagement in Database and Knowledge-base Systems

- **Part I:** Uncertainty in Relational Databases.

Fereidoon (Fred) Sadri
Dept. of Mathematical Sciences
Univ. of North Carolina at Greensboro
Greensboro, NC
email: `sadri@uncg.edu`

- **Part II:** Uncertainty in Deductive Databases.

Laks V. S. Lakshmanan
Dept. of Computer Science
Concordia University
Montreal, Quebec
email: `laks@cs.concordia.ca`

Part I: Uncertainty in Relational Databases.

- Introduction

- Review
  - Qualitative Methods.
  - Quantitative Methods.
    - ∗ Probabilistic.
    - ∗ Evidential (based on Dempster-Shafer theory of evidence).
    - ∗ Fuzzy-set theoretic (possibilistic).

- Information Source Tracking Method.
  - Model.
  - Query processing.
  - Semantics and correctness.
  - Efficiency.
  - Integrity constraints.
  - Aggregate operations.
- ProbView

Early works –
missing information, null values

Codd; TODS 1979.

Biskup; TODS 1983.

Lipski; TODS 1979.

Imielinski, Lipski; JACM 1984.

Reiter; 1984 (in *On Conceptual Modeling*).

Late 80's, early 90's: Many works on uncertainty in database and knowledge-base systems.

Various paradigms:

qualitative – quantitative

probabilistic, evidential, fuzzy

Renewed interest

Data mining

Information integration

A qualitative approach:

Liu and Sunderraman

"Indefinite and Maybe Information in Relational Databases," ACM TODS Vol.15, No. 1, March 1990, pp 1-39.

"A Generalized Relational Model for Indefinite and Maybe Information," IEEE TKDE Vol. 3, No. 1, March 1991, pp 65-77.

IDEA: A relation is partitioned into three components: *Definite*, *indefinite* (or *disjunctive*), and *maybe*.

Relational algebra operations are extended to operate on these (extended) relations.

Example:

s1	p2
s2	p3
s3	p1
s3	p5
s4	p3
s5	p3
s6	p4

p1	red
p3	blue
p2	red
p2	blue
p4	blue
p5	red

Query: Which suppliers supply a blue part?

s2
s4
s5
s1
s6

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Liu and Sunderraman's M-Tables

Improvement: Can represent disjunctions of tuples from different relations.

M-table scheme: List of relation schemes.

M-table instance has two components: *sure* and *maybe*.

Sure component represents definite tuples, and indefinite (disjunctive) tuples. Possible to have disjunctions of tuples from different relations.

Maybe component represents maybe tuples.

Relational Algebra extended for M-tables. New operations: R-projection, merge.

A quantitative approach – probabilistic

Barbara, Garcia-Molina, and Porter,

"The Management of Probabilistic Data," IEEE TKDE, Vol. 4, No. 5, October 1992, pp 487-502.

Probabilistic Data Model (*PDM*)

Example:

Empl	Dept	Quality Bonus	Sales
John	toy	0.3 [great yes] 0.4 [good yes] 0.2 [fair *] 0.1 [* *]	0.3 [$30–34K] 0.5 [$35–39K] 0.2 [*]

Deterministic key;
Deterministic or stochastic attribute-groups;
Can represent missing probabilities.

Probability ranges due to missing probabilities: John's sales is [$30-34K] with probability 30% to 50%.

PDM uses (extended) relational algebra operators project, select, natural join.

Select:
Condition: (att = value; prob θ value)
where θ is a comparison operator.

Natural join:
Common attributes should be the key for the second relation.
Missing probabilities may be generated.

Special operators, ϵ-select, stochastic, discrete, and group.

Set-difference is not discussed.

A quantitative approach – evidential

Lim, Srivastava, and Shekhar,

"An Evidential Reasoning Approach to Attribute Value Conflict Resolution in Database Integration," IEEE TKDE, Vol. 8, No. 5, October 1996, pp 707–723.

Extended relational model, based on *Dempster-Shafer Theory of Evidence.*

- Key values are definite.
- A (Non-key) attribute can be uncertain; value represented by a mass function.
- Each tuple also has a pair (sn, sp), lower and upper bound probabilities that tuple belongs to the relation (*tuple membership attribute*).

Extended relational algebra operators – select, project, union, Cartesian product, join (set-difference is not discussed.)

Select:

Conditions of the forms

A **is** $\{c_1, \ldots, c_n\}$, or $A\theta B$,

A, B: attributes or mass functions,

$c_1, \ldots, c_n$: (atomic) values,

θ: arithmetic comparison.

Dempster-Shafer Theory of Evidence.

A *mass function* m on the domain D assigns probabilities to subsets of D, such that:

$$m(\phi) = 0$$
$$\sum_{A \subseteq D} m(A) = 1$$

Given a set of values $X \subseteq D$, its *belief* and *plausibility* are defined as

$$\begin{aligned} bel(X) &= \sum_{A \subseteq X} m(A) \\ pls(X) &= \sum_{A \cap X \neq \phi} m(A) = 1 - bel(\bar{A}) \end{aligned}$$

Intuition:
Belief and plausibility are the lower-bound and upper-bound probabilities for v $(v \in X)$.

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Dempster's rule of combination.

Mass functions m_1 and m_2 on domain D, *combined mass* $m = m_1 \oplus m_2$:

$$m(Z) = \frac{\sum_{X \cap Y = Z} m_1(X).m_2(Y)}{\eta}$$

$X, Y, Z \subseteq D$.

Normalization factor

$$\eta = 1 - \sum_{X \cap Y = \phi} m_1(X).m_2(Y)$$

A quantitative approach − fuzzy

Fagin,

"Combining Fuzzy Information from Multiple Systems," ACM PODS, 1996, pp 216-226.

IBM **Garlic** project: A multimedia information system.

A query is a Boolean combination of atomic queries.

The answer to an atomic query is a fuzzy set:

Each element is a pair (x, g),
  x is an object
  $g \in [0, 1]$ is the membership *grade*

How to combine answers to atomic queries?

```
select  cd
where   artist     = "Beatles"  and
        albumColor = "red"      and
        sound      = "soft"
```

Here the multimedia (sub)system(s) contain artist information, snapshots of album covers, and audio of the songs, and have "Query by content" capability.

In principle, *Fuzzy queries* posed to traditional databases are handled in a similar manner.

Standard combination rules of fuzzy logic

Notation: $\mu_Q(x)$ = grade of x with respect to atomic query Q

$$\mu_{Q_1 \wedge Q_2}(x) = min\{\mu_{Q_1}(x), \mu_{Q_2}(x)\}$$

$$\mu_{Q_1 \vee Q_2}(x) = max\{\mu_{Q_1}(x), \mu_{Q_2}(x)\}$$

$$\mu_{\neg Q}(x) = 1 - \mu_Q(x)$$

Theorem (Yager; Dubois and Prade)

min and max are the unique combination functions for $\wedge$ and $\vee$ that
(1) preserve logical equivalence, and
(2) are monotone

NOTE: Combination and certainty calculation functions of IST also have these properties.

Fagin's paper contribution:

An efficient algorithm to obtain the top k answers to a conjunctive query.

Information Source Tracking Method: A Hybrid Approach

Idea:

Associate sources with data

Model accuracy of data by the reliability of its sources

Answers to a query / derived data: Also compute sources

Given quantitative reliability figures for sources, can compute quantitative certainty values for data / derived data.

Formally:

Associate a propositional formula with each unit of data.

ship	cause	source
Titanic	iceburg	*true*
Britanic	mine	$(s_1 \wedge s_2) \vee s_3$
Britanic	torpedo	$s_4 \vee s_5$

s_1, s_2, ... represent information sources.

(Britanic, mine) is true

if $(s_1 \wedge s_2) \vee s_3 = \mathit{true}$

if (s_1 and s_2 are correct) or s_3 is correct.

Certainty of (Britanic, mine) is a function of the *reliabilities* of sources.

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

What is an information source?
An abstraction.

A person (in a context)

A device

A test

...

A *composite* source

Different sources are independent.

Representation / Data Structures

information source vectors

Source information (propositional formula) is represented by vectors

ship	cause	source				
Titanic	iceburg	–	–	–	–	–
Britanic	mine	1	1	–	–	–
Britanic	mine	–	–	1	–	–
Britanic	torpedo	–	–	–	1	–
Britanic	torpedo	–	–	–	–	1

Elements of vectors

$\{\mathcal{O}, -1, +1, -\}$

$+1$	s_i	
-1	$\neg s_i$	
$-$		(*Independence*)
$\mathcal{O}$	$s_i \wedge (\neg s_i)$	(*Overspecification*)

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Efficiency / Space

Sparse matrix techniques.

Dual representation.

Query Processing in IST

(1) Determine sources for each answer.

(2) (optional) Compute certainties for answers.

Example:

ship	cargo	source				
Tina	flour	1	–	–	–	–
Manuel	rice	1	–	–	–	–
Manuel	flour	–	–	1	–	–
Oral	rice	–	–	–	1	–

ship	destination	source				
Tina	Lisboa	–	1	–	–	–
Manuel	Marseille	–	1	–	–	–
Manuel	Osaka	–	–	–	1	–
Oral	Marseille	–	–	–	–	1

Query: Which cities are receiving a shipment of rice?

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Query: Which cities are receiving a shipment of rice?

```
select  destination
from    shipCargo, shipDest
where   cargo = "rice" and
        shipCargo.ship = shipDest.ship
```

Answer:

destination	source				
Osaka	1	–	–	1	–
Marseille	1	1	–	–	–
Marseille	–	–	–	1	1

(Osaka) @ $s_1 \wedge s_4$
(Marseille) @ $(s_1 \wedge s_2) \vee (s_4 \wedge s_5)$

Certainties of answers

If reliabilities of sources are known, certainties can be calculated for the answers.

For example, assume all sources have a reliability of 80%, then:

Using probabilistic combination rules:

$cert$(Osaka): 64%
$cert$(Marseille): 87.04%

Using fuzzy sets combination rules:

$cert$(Osaka): 80%
$cert$(Marseille): 80%

Source Vector Operations

A set of source vectors represent a propositional formula:

$\{(1\ 1\ -\ -), (-\ -\ 1\ -)\}$ represents $(s_1 \wedge s_2) \vee s_3$.

sv-conjunction, *sv-disjunction*, *sv-negation*
$(\overset{s}{\wedge}, \overset{s}{\vee}, \overset{s}{\neg})$

Operands and result: source-vector sets,

Implement logical $\wedge$, $\vee$, $\neg$ of the corresponding propositional formulas.

Examples of source vector operations

$$\{(1\ 1\ -\ -),\ (-\ -\ 1\ -)\} \overset{s}{\wedge} \{(-\ -\ -\ 1)\} =$$
$$\{(1\ 1\ -\ 1),\ (-\ -\ 1\ 1)\}$$

$$((s_1 \wedge s_2) \vee s_3) \wedge (s_4) =$$
$$(s_1 \wedge s_2 \wedge s_4) \vee (s_3 \wedge s_4)$$

$$\{(1\ 1\ -\ -),\ (-\ -\ 1\ -)\} \overset{s}{\vee} \{(-\ -\ -\ 1)\} =$$
$$\{(1\ 1\ -\ -),\ (-\ -\ 1\ -),\ (-\ -\ -\ 1)\}$$

$$((s_1 \wedge s_2) \vee s_3) \vee (s_4) =$$
$$(s_1 \wedge s_2) \vee s_3 \vee s_4$$

$$\overset{s}{\neg} \{(1\ 1\ -\ -),\ (-\ -\ 1\ -)\} =$$
$$\{(\bar{1}\ -\ \bar{1}\ -),\ (-\ \bar{1}\ \bar{1}\ -)\}$$

$$\neg((s_1 \wedge s_2) \vee s_3)) =$$
$$(\neg s_1 \wedge \neg s_3) \vee (\neg s_2 \wedge \neg s_3)$$

Extended Relational Algebra

Select, project, union

$$\sigma_C(r) = \{t@u \mid t@u \in r, \text{ and } t \text{ satisfies condition } C\}$$

$$\Pi_X(r) = \{t[X]@u \mid t@u \in r\}$$

$$r \cup s = \{t@u \mid t@u \in r \text{ or } t@u \in s\}$$

where u is the information source vector associated with tuple t.

Note: Implicit sv-disjunction takes place for projection and union.

Intersection, Cartesian product, natural join

$$r \cap s \;=\; \{t@(u_1 \overset{s}{\wedge} u_2) \mid t@u_1 \in r, t@u_2 \in s\}$$

$$r \times s \;=\; \{(t_1.t_2@(u_1 \overset{s}{\wedge} u_2) \mid t_1@u_1 \in r, t_2@u_2 \in s\}$$

$$\begin{aligned} r \bowtie s \;=\; \{(t_1 \otimes t_2@(u_1 \overset{s}{\wedge} u_2) \mid\ & t_1@u_1 \in r, t_2@u_2 \in s, \\ & t_1 \text{ and } t_2 \text{ join } \} \end{aligned}$$

Set difference

$$\begin{aligned} r - s \;=\; \{t@x \mid\ & t@x \in r, t \notin s, \text{ or,} \\ & t@y \in r, t@z \in s, x = y \overset{s}{\wedge} (\overset{s}{\neg} z)\} \end{aligned}$$

How do we get -1 in source vectors?

List ships that have a cargo of rice but are not going to Osaka.

```
select ship
from   shipCargo
where  cargo = "rice" and
       ship not in (select ship
                    from   shipDest
                    where  destination = "Osaka")
```

Answer

ship	source				
manuel	1	–	–	$\bar{1}$	–
Oral	–	–	–	1	–

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Certainty Calcutaion: Phase 2 of IST

Given $t@x$, x is a set of source vectors

compute the certainty of t

as a function of the reliabilities of sources.

Equivalently: Given $t@f$, f a propositional formula in sources.

Can choose among different paradigms: probabilistic, possibilistic (fuzzy-set theoretic), ...

Probabilistic Approach

(Osaka) @ $s_1 \wedge s_4$

cert(Osaka) $= p_1 \times p_4$

Assumption:
Different information sources are independent.

How about t @ $s_i \vee s_j$?

cert(t) $= p_i + p_j - p_i \times p_j$

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Algorithm to calculate $cert(t)$ given $t@f$.

Convert f to disjunctive normal form (sum of products), $f = f_1 \vee \ldots \vee f_n$ (eliminate duplicates), then

$$cert(t) = \sum_{i=1}^{n} w_i$$

where w_i, the weight of f_i, is obtained by the product rule.

Example

(Marseille) @ $(s_1 \wedge s_2) \vee (s_4 \wedge s_5)$

$$
\begin{aligned}
(s_1 \wedge s_2) \vee (s_4 \wedge s_5) = \\
&(s_1 \wedge s_2 \wedge s_4 \wedge s_5) \vee \\
&(s_1 \wedge s_2 \wedge \neg s_4 \wedge s_5) \vee \\
&(s_1 \wedge s_2 \wedge s_4 \wedge \neg s_5) \vee \\
&(s_1 \wedge s_2 \wedge \neg s_4 \wedge \neg s_5) \vee \\
&(s_1 \wedge s_2 \wedge s_4 \wedge s_5) \vee \\
&(\neg s_1 \wedge s_2 \wedge s_4 \wedge s_5) \vee \\
&(s_1 \wedge \neg s_2 \wedge s_4 \wedge s_5) \vee \\
&(\neg s_1 \wedge \neg s_2 \wedge s_4 \wedge s_5)
\end{aligned}
$$

Note that $s_1 \wedge s_2 \wedge s_4 \wedge s_5$ is repeated.

$cert$(Marseille) = 87.04%

Possible-Worlds Semantics

Reference: F. Sadri, "Modeling Uncertainty in Databases," IEEE ICDE, 1991.

An IST relation *represents* a set of (regular) relations called its *possible world*.

k sources $\Rightarrow 2^k$ combinations of Boolean values for source variables.

Each regular relation corresponds to one Boolean combination.

Hence, up to 2^k regular relations.

Example

ship	cargo	source	
Tina	flour	1	–
Manuel	rice	1	–
Manuel	flour	–	1

Possible world contains the empty relation, plus:

$s_1 =$ *true*, $s_2 =$ *false*

ship	cargo
Tina	flour
Manuel	rice

$s_1 =$ *false*, $s_2 =$ *true*

ship	cargo
Manuel	flour

$s_1 =$ *true*, $s_2 =$ *true*

ship	cargo
Tina	flour
Manuel	rice
Manuel	flour

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Soundness and Completeness

The result of applying an extended relational algebra operation to IST relations and taking the possible world of result **is the same as** first taking the possible worlds of IST relations and then applying the (regular) relational algebra operation.

$$\begin{tikzcd}
\text{IST relations} & r \arrow[d] & s \arrow[d] \arrow[r] & r \text{ extendedOp } s \arrow[d] \\
\text{Regular relations} & w(r) & w(s) \arrow[r] & w(r)\ \text{Op}\ w(s) = w(r \text{ extendedOp } s)
\end{tikzcd}$$

$w(r)$ represents the possible world of r.

Correctness of Certainty Calculation

Let r be an IST relation.

$t@f$ is a tuple in r.

Each (regular) relation in the possible world of r, $r_i \in w(r)$, has a probability P_i of being the *real* world. P_i is a function of source reliabilities.

Certainty of t

$$cert(t) = \sum_{t \in r_i} P_i$$

This is exactly what our algorithm computes.

Efficiency of IST

Reference: F. Sadri, "Information Source Tracking Method: Efficiency Issues." *IEEE TKDE*, Vol. 7, No. 6, December 1995.

I/O complexity of IST is similar to regular relations. That is, assuming no indexing,

Linear for select, project.

Almost linear (by hashing) for join, union, intersection, set-difference.

BUT IST relation size (compared to regular relations):

Tuple width – extra "source attribute"

Number of tuples – exponential in number of sources in the worst case.

Alternate scheme – Dual IST

For *base* relations, same number of tuples as regular relations.

IST uses "disjuncts of conjuncts" (sum of products) representation;

Dual IST uses "conjuncts of disjuncts".

Integrity Constraints

Reference: F. Sadri, "Integrity Constraints in the Information Source Tracking Method." *IEEE TKDE*, Vol. 7, No. 1, February 1995.

Key constraints / single-valued attributes

It is illegal for a source to provide contradictory information.

Definition: Let r be an IST relation, K be the key, t_1@u and t_2@v be tuples in r. If $t_1(K) = t_2(K)$, then u and v can not both have a $+1$ for the same source s_i.

Different sources can provide contradictory information.

Example

ship	position	source		
Tina	Atlantic Ocean	1	–	–
Tina	Pacific Ocean	–	1	–
Oral	Mediterranean	1	–	–

"ship" is the key.

We can not assume sources s_1 and s_2 are independent.

In fact, they are *mutually exclusive*.

e_i : "event" that i-th tuple is valid.

$$p(e_1|e_2) = p(e_2|e_1) = 0$$

where $p(e_i|e_j)$ is the (conditional) probability of the event e_i given e_j.

But, this causes a problem.

Assume s_1 and s_2 have high reliabilities, say 90% and 80%.

$P(e_1 \vee e_2) = P(e_1) + P(e_2) = 170\%$!!!

But, in practice, reliable sources sometimes produce contradictory information.

Our solution

Modify reliabilities of conflicting sources *only as far as contradictory data is concerned*.

Introduce new (virtual) sources only in the conflicting tuples.

ship	position	source				
Tina	Atlantic Ocean	–	–	–	1	–
Tina	Pacific Ocean	–	–	–	–	1
Oral	Mediterranean	1	–	–	–	–

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

How to calculate reliabilities of virtual sources?

1. Preservation of ratio:

$$\frac{p_4}{p_5} = \frac{p_1}{p_2}$$

2. Preservation of all-fail probability:

   $p(\neg e_1 \wedge \neg e_2) = 1 - p_4 - p_5 = (1 - p_1) \times (1 - p_2)$

   At least one of the sources is correct, with the same probability that it would be if there were no conflict.

For $p_1 = 90\%$ and $p_2 = 80\%$

$p_4 = 51.88\%$ and $p_5 = 46.12\%$

$$\begin{aligned} cert(Tina, AtlanticOcean) &= 51.88\% \\ cert(Tina, PacificOcean) &= 46.12\% \\ cert(Oral, Mediterranean) &= 90\% \end{aligned}$$

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

General Case

Clusters of *coinciding* sources.

Conflicting clusters.

Each cluster is modeled by a single virtual source (only in contradictory data).

ship	position	source		
Tina	Atlantic Ocean	1	–	–
Tina	Atlantic Ocean	–	–	1
Tina	Pacific Ocean	–	1	–
Oral	Mediterranean	1	–	–

After adjustment (virtual source s_4 represents the cluster $\{s_1, s_3\}$, and s_5 represents $\{s_2\}$).

ship	position	source				
Tina	Atlantic Ocean	–	–	–	1	–
Tina	Pacific Ocean	–	–	–	–	1
Oral	Mediterranean	1	–	–	–	–

Certainty Calculation Algorithm Revisited

Legal disjunctive normal form:

Has at most one positive variable corresponding to conflicting virtual sources in each conjunct.

Example Let s_1, s_2, s_3 be (ordinary) sources, and s_4 and s_5 be conflicting virtual sources. Take (1 1 1 − −) (represents $s_1 \wedge s_2 \wedge s_3$.) Its legal disjunctive normal form is:

$\{(1\ 1\ 1\ 1\ -1), (1\ 1\ 1\ -1\ 1), (1\ 1\ 1\ -1\ -1)\}$, or,

$(s_1 \wedge s_2 \wedge s_3 \wedge s_4 \wedge \neg s_5) \vee (s_1 \wedge s_2 \wedge s_3 \wedge \neg s_4 \wedge s_5) \vee (s_1 \wedge s_2 \wedge s_3 \wedge \neg s_4 \wedge \neg s_5)$

Algorithm to calculate $cert(t)$ given $t@f$.

Convert f to legal disjunctive normal form $f = f_1 \vee \ldots \vee f_n$, eliminate duplicates,

$$cert(t) = \sum_{i=1}^{n} w_i$$

w_i, the weight of f_i, is calculated as

$$w_i = \prod_{a_j=1} p_j \prod_{a_j=-1} (1 - p_j) \prod_{k=1,\ldots,m} g_k$$

where the formula f_i, in its source vector representation, has the form $(a_1, \ldots, a_n, x_1, \ldots, x_m)$, $a_1, \ldots, a_n$ are values corresponding to (ordinary) sources, while each x_k represents a *vector* corresponding to a conflicting virtual source group G_k, and

$$g_k = \begin{cases} p_j & \text{if } f_i \text{ has a } +1 \text{ for } s_j \text{ in group } G_k \\ p' & \text{otherwise} \end{cases}$$

p' is the "all-fail probability" of group G_k.

Possible Worlds Revisited

ship	position	source				
Tina	Atlantic Ocean	–	–	–	1	–
Tina	Pacific Ocean	–	–	–	–	1
Oral	Mediterranean	1	–	–	–	–

Legal possible world: Empty relation and

ship	position
Oral	Mediterranean

ship	position
Tina	Atlantic Ocean

ship	position
Tina	Pacific Ocean

ship	position
Tina	Atlantic Ocean
Oral	Mediterranean

ship	position
Tina	Pacific Ocean
Oral	Mediterranean

Correctness

In the presence of key constraints, query processing and certainty calculation algorithms for IST are correct with respect to the legal possible worlds semantics.

Aggregate Operations

Reference: F. Sadri "Aggregate Operations in Information Source Tracking Method." *Theoretical Computer Science*, Vol. 133, 1994.

Aggregate operations: min, max, sum, count, average.

Example

Employee	Salary	source
a	40K	1 0
b	60K	0 1

Emp

```
select  sum(Salary)
from    Emp
```

Four possible answers: 0, 40K, 60K, 100K (one per regular relation in w(Emp))

Given source reliabilities 90% and 80% for s_1 and s_2, the certainty of the four answers are: 2%, 18%, 8%, and 72%, respectively.

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Algorithm (Based on the conversion of source vectors to disjunctive normal form)

Employee	Salary	source	
b	60K	-1	1
a	40K	1	-1
a	40K	1	1
b	60K	1	1

Expanded-Emp

Carry out the aggregation for each source group:

```
select     sum(Salary), source
from       Expanded-Emp
group by   source
```

Sum	source	
60K	-1	1
40K	1	-1
100K	1	1

Complexity (worst-case): $O(2^k n)$
k is the number of sources

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Other Formulations

- P1: What is the probability that the result of the aggregate operation is equal to c.

- P2: Find the largest (smallest) possible value of the result of the aggregate operation.

- P3: Is it possible for the result to be greater than or equal to (or less than or equal to) a given value?

- P4: Find the expected value of the result.

- P5: Find the normalized expected value of the result. (Do not take the empty (regular) relation in $w(r)$ into account).

Algorithms

We have algorithms that avoid the materialization of possible worlds relations for the following:

P1-min, P1-max,
P2-min, P2-max,
P3-min, P3-max,
P4-sum, P4-count,
P5-sum, P5-count.

Algorithm P4-sum

P4: Find the expected value of $agg_A(r)$.

$$\textit{expecValue}(sum_A(r)) = \sum_{i=1}^{n} (t_i(A) \times cert(t_i))$$

where $cert(t_i)$ is the certainty of tuple t_i.

Complexity Results

The following problems are shown to be NP-Complete:

P1-sum, P1-count, P1-average
P2-sum, P2-count, P2-average
P3-sum, P3-count, P3-average

Lakshmanan, Leone, Ross, Subrahmanian

"ProbView: A Flexible Probabilistic Database System," ACM TODS, Vol. 22, No. 5, September 1997, pp 419–469.

Quantitative approach – probabilistic.

Attribute-level uncertainties.

Uncertainties are intervals.

Example (Probabilistic Relation)

Tid	File	Person
t_1	im1.gif	John Smith [0.3,0.4], Mark Bloom [0.5,0.65]
t_2	im1.gif	Mark Bloom [0.3,0.35], Ted Lewis [0.6,0.8]
t_3	im2.gif	Mark Bloom [1,1]
t_4	im2.gif	Ted Lewis [0.1,0.2]
t_5	im3.gif	Lynn Bloom [0.6,0.7], Elsa Bloom [0.2,0.25]

Flattening probabilistic relations

Tuple t_1 Gives rise to two possible worlds:

w_1, in which t_1.person = John Smith
w_2, in which t_1.person = Mark Bloom

Probabilities of w_1 and w_2, p_1 and p_2, should satisfy

$$\begin{gathered} 0.30 \leq p_1 \leq 0.40 \\ 0.50 \leq p_2 \leq 0.65 \\ p_1 + p_2 \leq 1 \\ p_1, p_2 \geq 0 \end{gathered}$$

Lower (and upper) bounds for p_is are obtained by minimizing (and maximizing) p_i with respect to the above constraints (linear programming).

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

Annotated relation (obtained by flattening probabilistic relation)

File	Person	LB	UB	path
im1.gif	John Smith	0.3	0.4	w_1
im1.gif	Mark Bloom	0.5	0.65	w_2
im1.gif	Mark Bloom	0.3	0.35	w_3
im1.gif	Ted Lewis	0.6	0.7	w_4
im2.gif	Mark Bloom	1	1	w_5
im2.gif	Ted Lewis	0.1	0.2	w_6
im3.gif	Lynn Bloom	0.6	0.7	w_7
im3.gif	Elsa Bloom	0.2	0.25	w_8

An annotated tuple t is *consistent* if

t.LB $\leq$ t.UB, and

t.path is a consistent Boolean expression

Extended SQL

User specifies combination mode (function)

Example

```
UNDER   conj=ignorance
SELECT  F.File, E.Salary
FROM    face F, emp E
WHERE   F.Person = E.name AND
        F.LB     > 0.7    AND
        E.Salary > 100,000
```

Ignorance is the combination mode where nothing is known about the dependency between events:

$$[\alpha_1, \beta_1] \wedge_{ig} [\alpha_2, \beta_2] = \\ [max(0, \alpha_1 + \alpha_2 - 1), min(\beta_1, \beta_2)]$$

Some Combination Functions

Ignorance:

$$[\alpha_1, \beta_1] \wedge_{ig} [\alpha_2, \beta_2] = \\ [max(0, \alpha_1 + \alpha_2 - 1), min(\beta_1, \beta_2)]$$

$$[\alpha_1, \beta_1] \vee_{ig} [\alpha_2, \beta_2] = \\ [max(\alpha_1, \alpha_2), min(1, \beta_1 + \beta_2)]$$

Positive Correlation:

$$[\alpha_1, \beta_1] \wedge_{pc} [\alpha_2, \beta_2] = [min(\alpha_1, \alpha_2), min(\beta_1, \beta_2)]$$

$$[\alpha_1, \beta_1] \vee_{pc} [\alpha_2, \beta_2] = [max(\alpha_1, \alpha_2), max(\beta_1, \beta_2)]$$

Others functions are possible – should satisfy certain properties (postulates)

Extended Relational Algebra

Relational algebra is extended to handle special attributes LB, UB, and path.

Compaction is a new operator to combine duplicate tuples in an annotated relation.

Part II: Uncertainty in Deductive Databases

Overview

- Motivation
- Lay of the land
- Fuzzy/Possibilistic Approaches
- Probabilistic Approaches
- Hybrid Approaches
  - Deductive IST
  - situation theory
- Parametric Framework and Query Optimization
- Summary and Future Directions

Scope of Part II

Uncertainty in Knowledge Representation:

- vast field with long history

- numerous uncertainty formalisms: e.g., *fuzzy set theory, probability theory, Dempster-Shafer theory, multi-valued logic, ad hoc "formalisms" (e.g., MYCIN)*, etc.

- comprehensive overview – beyond scope of this tutorial

- our focus: *modeling and manipulation of uncertainty in deductive databases and logic programming systems*

1. Motivation

- What are DDB/LP + uncertainty systems and what are they good for?

- Basic idea:
  - uncertain relational DBs: tuples annotated with some measure of uncertainty.
  - give rules similar status.

- Example 1: Medical domain.

$$highRisk(X, D) \stackrel{[0.6,0.7]}{\longleftarrow} midaged(X) \wedge familyHistory(X, D).$$

$$prognosis(X, D) \stackrel{[0.7,0.8]}{\longleftarrow} highRisk(X, D).$$

$midaged(john)$: [0.3, 0.3].
$midaged(peter)$: [0.7, 0.7].
$midaged(mary)$: [0.2, 0.3].
$familyHistory(john, diabetes)$: [0.4, 0.5].
$familyHistory(mary, myopia)$: [0.6, 0.6].
...
Query: $\leftarrow prognosis(X, myopia)$.

- many other applications: *image databases* (e.g., the FIST system of Univ. of Maryland), *Information integration involving media objects* (e.g., the Garlic project at IBM Almaden), *defense applications*, *data mining* (see Heckerman 97), ...

2. Lay of the Land

Two major criteria for classification.

1. **Underlying uncertainty formalism**:

- Fuzzy set approaches:

- Possibility theoretic:

- Probability theoretic:

- Evidence theory, Dempster-Shafer theory based:

- Multi-valued logics:

- Hybrid (numeric + non-numeric):

2. Lay of the Land (contd.)

2. Uncertainty attachment mechanism:

- *Implication Based*: uncertainty captures degree of truth of implication; e.g., the rule: "midagedness and family history for a disease *implies* you are a high risk candidate for that disease" is true with probability between 0.6 and 0.7.

- *Annotation Based*: implication is classical; rules say whenever probabilities of subgoals lie in certain ranges, then probability of rule head lies in a range that is a function of the subgoal ranges;

  Example 2:

  $highRisk(X, D) : [0.6 * min(P_1, P_3), 0.7 * min(P_2, P_4)] \longleftarrow$
  $midaged(X) : [P_1, P_2] \ \wedge$
  $familyHistory(X, D) : [P_3, P_4].$

  P_i – uncertainty variables.

  could use other functions in place of '*' and *min*.

2. Lay of the Land (contd.)

1. **Uncertainty formalisms**:

- Fuzzy set based:
  - Logic programs with fuzzy (un)certainties: Shapiro 83.
  - Quantitative deduction – first formal framework: van Emden 86
    efficient query processing: Steger et al. 89.

- Possibility theoretic: Dubois et al. 91.

- Multi-valued Logic based:
  - generalizing fuzzy set based approaches.
  - Subrahmanian 87, Kifer and Li 88, Blair and Subrahmanian 89, Kifer and Subrahmanian 91.
  - Belief and Doubt – bilattice based approaches: Fitting 88, 91.

- Evidence theoretic:
  - Baldwin 87
  - Dempster-Shafer theory: Ng and Subrahmanian 91, 93.

2. Lay of the Land (contd.)

- Probabilistic:
  - Nilsson 86: Probabilistic logic.
  - Fagin et al. 89, Halpern 91: probabilistic logics and reasoning.
  - Ng and Subrahmanian 89-present: probabilistic LP.
  - Kiessling et al. 89-97, Guntzer et al. 91: DUCK system for uncertainty manipulation, based on probabilistic calculus, support for problematic knowledge, fusion of knowledge from multiple sources.
  - Lakshmanan and Sadri 94, 97: Deductive databases with IST model of uncertainty; query optimization.
  - Lakshmanan and Sadri 94: probabilistic deductive databases with user controlled modes of event interaction; PTIME data complexity for fixpoint evaluation.
  - Lakshmanan 94: Epistemic foundation for probabilistic uncertainty in terms of situations.
  - Lakshmanan and Shiri 97, Shiri 97: Parametric framework for unifying all known implication based frameworks for DDB + uncertainty; all formalisms can be embedded; query optimization studied in a very generic context.

2. Lay of the Land (contd.)

2. **Uncertainty attachment mechanisms**:

- Annotation Based:

  Annotated logic programming:
  Blair and Subrahmanian 89, Kifer and Subrahmanian 91.
  annotation – orthogonal to uncertainty.

  LP + uncertainty frameworks inspired by ALP:
  Subrahmanian 87, Kifer and Li 88, Ng and Subrahmanian 89, 90, 90+.

- Implication Based:

  Fuzzy/Possibilistic: van Emden 86, Fitting 88, 91, Dubois et al. 91.

  Probabilistic/Hybrid: Lakshmanan 94, Lakshamnan & Sadri 94.

3. Fuzzy/Possibilistic Approaches

van Emden 86:

- quantitative logic programming.

- fact = relational tuple.

- rule = a Horn clause, i.e. a conjunctive query, possibly recursive:
  $H \stackrel{c}{\longleftarrow} G_1, \ldots, G_n \quad \langle max, \times, min \rangle.$

- c – a number in the uncertainty lattice $[0, 1]$: the so-called attenuation factor.

- *min* playing the role of AND for combining certainties of G_i.

- $\times$ used for "propagating" certainty from rule body to head.

- *max* playing the role of OR for combining certainties of multiple derivations of a tuple.

3. Fuzzy/Possibilistic Approaches (contd.)

Example 3:

$$
\begin{array}{lll}
r_1: & close(a,b) \xleftarrow{[0.9]} ; & \langle max, \times, \_ \rangle \\
r_2: & close(a,c) \xleftarrow{0.8} ; & \langle max, \times, \_ \rangle \\
r_3: & close(b,c) \xleftarrow{0.7} ; & \langle max, \times, \_ \rangle \\
r_4: & close(Y,X) \xleftarrow{1} close(X,Y); & \langle max, \times, \_ \rangle \\
r_5: & outbreak(d,a) \xleftarrow{1} ; & \langle max, \times, \_ \rangle \\
r_6: & connected(X,Y) \xleftarrow{1} close(X,Y); & \langle max, \times, \_ \rangle \\
r_7: & connected(X,Y) \xleftarrow{0.8} close(X,Z), connected(Z,Y); & \\
 & & \langle max, \times, min \rangle \\
r_8: & affected(D,A) \xleftarrow{0.7} outbreak(D,A); & \langle max, \times, \_ \rangle \\
r_9: & affected(D,A) \xleftarrow{0.8} close(A,A_1), affected(D,A_1); & \\
 & & \langle max, \times, min \rangle
\end{array}
$$

■

Query: $\longleftarrow \; affected(d,A).$

Some answers: $affected(d,a) : 0.7$

$affected(d,b) : 0.8 \times min(0.9, 0.7) = 0.56$

$affected(d,c) : p$

where $p = max(0.8 \times min(0.8, 0.7), 0.8 \times min(0.7, 0.7)) = 0.56.$

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

3. Fuzzy/Possibilistic Approaches (contd.)

Highlights:

- Simple and intuitive for user.
- consistently extends classical theory of DDB and LP.
- recursive queries can be evaluated in PTIME in database size, just as for datalog. (Lakshmanan & Sadri 94).

Limitations:

- restricted modes for combining probabilities of tuples.
- does not capture selective interdependence.
- intervals of certainties more relaistic.
- no negation (classical or otherwise).

3. Fuzzy/Possibilistic Approaches (contd.)

Dubois et al. 90:

Example 4:

$$\begin{array}{ll} r_1: & A \xleftarrow{0.8} B. \\ r_2: & A \xleftarrow{0.7} C. \\ r_3: & B \xleftarrow{0.7} . \\ r_4: & C \xleftarrow{0.8} . \end{array}$$

All conjunction and propagation functions are *min*;
all disjunction functions are *max*.

MYCIN (Buchanan and Shortliffe 75): revist Example 4.
All conjunction and propagation functions are $*$;
all disjunction functions are f_d, defined as: $f_d(\alpha, \beta) = \alpha + \beta - \alpha * \beta$.
Rationale for f_d: independence assumed.

Fairly similar to van Emden's framework.
Similar remarks apply.

4. Probabilistic Approaches

Ng and Subrahmanian 89-present:

Probabilistic Logic Programming:

- rules of the form $H : [\alpha_0, \beta_0] \longleftarrow G_1 : [\alpha_1, \beta_1] \wedge \cdots \wedge G_n : [\alpha_n, \beta_n]$.
- α_i, β_i – constants and $0 \leq \alpha_i \leq \beta_i \leq 1$.
- G_i can be annotated conjunctions or disjunctions.
- Conjunction in rule bodies and implication associated with rules – classical.
- assignment of annotation for head atom – constant and chosen by user.
- Intervals $[p_1, q_1]$ and $[p_2, q_2]$ associated with same tuple A are combined into $[max(p_1, p_2), min(q_1, q_2)]$.

4. Probabilistic Approaches (contd.)

Probabilistic Logic Programming (contd.):

Example 5: Reliability of connections provided by a long distance carrier.

$connection(X, Y)$: [0.85, 0.95] $\longleftarrow$ $a(X, Y)$: [1, 1].
$connection(X, Y)$: [0.9.1] $\longleftarrow$ $b(X, Y)$: [1, 1].
$connection(X, Y)$: [0.8, 0.95] $\longleftarrow$ $a(X, Y)$: [1, 1]$\wedge$
    $connection(Y, Z)$: [0.85, 1].
$connection(X, Y)$: [0.85, 1] $\longleftarrow$ $b(X, Y)$: [1, 1]$\wedge$
    $connection(Y, Z)$: [0.75, 1].

Suppose we have the following facts:

$a(p, q)$: [1, 1], $a(q, r)$: [1, 1], $b(q, r)$: [1, 1]. Then what is the reliability of the connection between p and r?

- We get both $connection(q, r)$: [0.85, 0.95] and $connection(q, r)$: [0.9, 1].
- interval intersection gives a combined reliability of $connection(q, r)$: [0.9, 0.95].
- Then required reliability is $connection(p, r)$: [0.85, 1].
- Key: $connection(q, r)$: [0.9, 0.95] logically implies (and hence satisfies the condition) $connection(q, r)$: [0.75, 1].

4. Probabilistic Approaches (contd.)

Probabilistic Logic Programming (contd.):

Highlights:

- first formulation of probabilistic logic programming.

- neatly integrates probabilistic reasoning within LP framework.

- elegantly deals with the problem of combining probabilities of simple events into compund boolean combinations, using Fenstad's results: principle of "total ignorance".

  – Given $A : [\alpha, \beta]$ and $B : [\gamma, \delta]$, knowing/assuming nothing about the kind of interdependence between A and B.

  – tightest interval we can derived for $A \wedge B$ is $[max(0, \alpha + \gamma - 1), min(\beta, \delta)]$.

  – tightest interval we can derived for $A \vee B$ is $[max(\alpha, \gamma), min(1, \beta + \delta)]$.

- simple intuitive semantics and efficient query processing procedures.

- fixpoint (and hence queries) can be computed in PTIME in database size (follows from results of Lakshmanan and Sadri 94).

4. Probabilistic Approaches (contd.)

Probabilistic Logic Programming (contd.):

Limitations:

- Limited expressive power; source: constant annotations.
- ignorance based approach is elegant and "safe" (i.e. makes no unrealistic assumptions); but overly conservative: what if you *knew* some events were independent, e.g.?
- interval intersection is just one possible means of combining probabilities of multiple derivations of a tuple; there are other reasonable means as well: e.g., independence, positive/negative correlation. ...; sometimes intersection can be unintuitive: e.g., if there are several (unreliable) paths from a to b, depending on the overlap among the paths, can choose some form of correlation for reliability combination.
- some of these limitations overcome in the hybrid probabilistic logic programming framework of Dekhtyar and Subrahmanian 97; faster query evaluation strategies developed as well.

4. Probabilistic Approaches (contd.)

Probabilistic Logic Programming (contd.):

Extensions (Ng and Subrahmanian 90s):

- pf-programs: annotations can involve variables and functions.
  - pf-rule: $H : \mu_0 \longleftarrow G_1 : \mu_1, \ldots, G_n : \mu_n$.
    μ_i – annotations;
    e.g. $[V_1, V_2], [V_1 * 0.5, V_2 * 0.5], [0.5, 0.6], [(V_1 + V_2)/2, (V_3 + V_4)/2]$;
    Gs and H may be conjunctions or disjunctions.
  - incorporating negation and stable model semantics.
  - Hybrid pf-programs (Dekhtyar and Subrahmanian 97): extending pf-programs by allowing multiple modes for combining probabilities.

Example 6:

$(F_1 \wedge F_2) : [V_1 * V_3, V_2 * V_4] \longleftarrow F_1 : [V_1, V_2], F_2 : [V_3, V_4]$.
– independence.

$A \wedge B : [p * v_1, p * V_2] \longleftarrow B : [V_1, V_2]$.
– says $\mathbf{Pr}(A|B) = p$.

$\bar{F} : [1 - V_2, 1 - V_1] \longleftarrow F : [V_1, V_2]$.
– (epistemic) negation.

$A : [min(V_1, \ldots, V_n), min(V_1, \ldots, V_n)] \longleftarrow B_1 : [V_1, V_1], \ldots, B_n : [V_n, V_n]$.
– simulates van Emden's framework.

■

4. Probabilistic Approaches (contd.)

pf-programs (contd.):

Highlights:

- clean model-theoretic semantics and fixpoint theory.
- sound and weakly complete query processing procedure.
- method for handling probabilistic inconsistency: e.g., $A : [0.2, 0.3]$ vs. $A : [0.4, 0.5]$.

Limitations:

- limitatons of p-programs framework, except expressive power; specifically, fixed means of combining probabilities of multiple derivations of tuples.
- continuity is lost: $\Rightarrow$ for certain recursive query programs, answers cannot be determined to arbitrary precision in finite time, even though input DB is finite.
- as a consequence, not clear which classes of queries can be efficiently evaluated (say, in PTIME).

4. Probabilistic Approaches (contd.)

Probabilistic Deductive Databases (Lakshmanan and Sadri 94):

- need for simultaneous belief and doubt:
  - form of expert's knowledge about application domain: phenomenon F occurs in soe situations and does not occur in some other situaions $\Rightarrow$ need for both belief and doubt
  - insufficient knowledge about exact nature of event interaction $\Rightarrow$ need for probability intervals
  - incomplete expert knowledge about "situations" in which phenomena modeled occur/do not occur $\Rightarrow$ situations where expert is unsure of F's truth status $\Rightarrow$ belief and doubt intervals are *not* complementary
- implication based framework; p-rule of the form: $A \stackrel{\langle[\alpha,\beta],[\gamma,\delta]\rangle}{\longleftarrow} B_1, \ldots, B_n$.
- facts – special cases of rules: $A : \langle[\alpha, \beta], [\gamma, \delta]\rangle$.

4. Probabilistic Approaches (contd.)

PDDB (contd.):

Example 7: (Pooling together Gallop polls before 1993 Canadian Federal elections.)

Results for age group 19-30.

1. Between 50% and 53% favor liberals.
2. Between 30% and 33% favor the reformists.
3. Between 5% and 8% favor the tories.

One way to assimilate the poll results:

1. $vote(X, liberals) \xleftarrow{\langle[0.5,0.53],[0.35,0.41]\rangle} ageGroup30(X).$
2. $vote(X, reform) \xleftarrow{\langle[0.3,0.33],[0.55,0.61]\rangle} ageGroup30(X).$
3. $vote(X, tories) \xleftarrow{\langle[0.05,0.08],[0.8,0.86]\rangle} ageGroup30(X).$

■

4. Probabilistic Approaches (contd.)

PDDB (contd.):

Basic Idea:

- each fact and rule associated with a *confidence level* of the form $\langle[\alpha, \beta], [\gamma, \delta]\rangle$.
- $0 \leq \alpha \leq \beta \leq \gamma \leq \delta \leq 1$.
- $[\alpha, \beta]$ not complement of $[\gamma, \delta]$.
- interesting algebraic structure of confidence levels – called *trilattice*.
- choice of modes for combining probability intervals:
  - multi-valued logic:
    $\wedge_t$: pointwise *min* of the two belief intervals and pointwise *max* of the two doubt intervals;
    $\wedge_k$: pointwise *min* for both belief and doubt;
    $\wedge_p$: (completed) union of belief (doubt) intervals;
    similar options for $\vee$
  - probabilistic: *independence*, *ignorance*, *positive correlation*, *negative corrlation*, *mutual exclusion*, etc.;
    symmetric treatment of $\wedge$ and $\vee$
  - many choices for semantics – each useful under different cirsumstances

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

4. Probabilistic Approaches (contd.)

PDDB (contd.):

Example 8:

1. $highRisk(X, D) \xleftarrow{\langle[0.65,0.65],[0.1,0.1]\rangle} midaged(X),$
   $familyHistory(X, D).$

2. $takes(X, T) \xleftarrow{\langle[0.4,0.4],[0,0]\rangle} highRisk(X, D), treatment(D, T).$

3. $prognosis(X, D) \xleftarrow{\langle[0.7,0.7],[0.12,0.12]\rangle} highRisk(X, D).$

4. $prognosis(X, D) \xleftarrow{\langle[0.05,0.05],[0.9,0.9]\rangle} takes(X, T),$
   $sideEffects(T, D).$

Combination Modes:

- Conjunction: Rules 1 & 4 – *independence*; Rules 2 & 3 – *ignorance*.

- Disjunction: *positive correlation* for all predicates.

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

4. Probabilistic Approaches (contd.)

PDDB (contd.):

Example 8 (contd.):

Input DB:
$midaged(john)$: $\langle[0.5, 0.6], [0.2, 0.3]\rangle$.
$familyHistory(john, cancer)$: $\langle[0.3, 0.5], [0.2, 0.4]\rangle$.
$familyHistory(john, hairloss)$: $\langle[0.2, 0.3], [0.6, 0.7]\rangle$.
$treatment(cancer, t1)$: $\langle[0.6, 0.7], [0.1, 0.2]\rangle$.
$sideEffects(t1, hairloss)$: $\langle[0.1, 0.2], [0.7, 0.8]\rangle$.

Query: $\longleftarrow$ $prognosis(hohn, D)$.

One answer: $prognosis(john, hairloss)$: $\langle[p, q], [r, s]\rangle$, where

$$\begin{array}{l}
\langle[p, q], [r, s]\rangle = \\
((\langle[0.65, 0.65], [0.1, 0.1]\rangle \wedge_{ind} \\
\langle[0.5, 0.6], [0.2, 0.3]\rangle \wedge_{ind} \\
\langle[0.2, 0.3], [0.6, 0.7]\rangle) \wedge_{ign} \\
\langle[0.7, 0.7], [0.12, 0.12]\rangle) \\
\qquad \vee_{pc} \\
(((\langle[0.65, 0.65], [0.1, 0.1]\rangle \wedge_{ind} \\
\langle[0.5, 0.6], [0.2, 0.3]\rangle \wedge_{ind} \\
\langle[0.3, 0.5], [0.2, 0.4]\rangle) \wedge_{ign} \\
\langle[0.6, 0.7], [0.1, 0.2]\rangle) \wedge_{ign} \\
\langle[0.4, 0.4], [0, 0]\rangle) \wedge_{ind} \\
\langle[0.1, 0.2], [0.7, 0.8]\rangle \wedge_{ind} \\
\langle[0.05, 0.05], [0.9, 0.9]\rangle.
\end{array}$$

4. Probabilistic Approaches (contd.)

PDDB (contd.):

Highlights:

- simple intuitive syntax and clean declarative semantics.
- sound and complete query processing procedure.
- (fixpoint) bottom-up and proof-theory (top-down) models of computation for query answers.
- simultaneous support for multiple, independent modes for combining probabilities.
- PTIME data complexity for recursive queries (just as for classical datalog) under disciplined use of modes: e.g., *pc* for disjunction associated with all recursive predicates; arbitrary modes on acylic data.
- proof of PTIME complexity result implies similar result for many previous frameworks, e.g., van Emden, p-programs of Ng and Subrahmanian, possibilistic framework of Dubois et al. ... .
- extension for stratified negation – straightforward.

Limitations:

- Head limited to atoms (no conjunctions/disjunctions).
- tradeoff between expressive power (in terms of kinds of probabilistic statements expressible) and complexity/termination properties.

5. Hybrid Approaches

5.1 Deductive IST (Lakshmanan and Sadri 94-97):

- extension of IST to deductive data model.
- rule: $A \xleftarrow{S} B_1, \ldots, B_n$;
  S a set of vectors over information sources/providers, with positive (1), negative (-1), noncommittal ($\perp$) and overcommittal ($\top$) contributions.
- facts – special case of rules, as usual.
- Normally, for base facts, contributions are limited to $1, \perp$.
- $-1, \top$ come about boolean combinations on source contributions, induced by rules.
- similar assumptions about individual sources – all their contributions are totally dependent – and about different sources – their contributions are independent.
- e.g.: $\mathit{target}(X, \mathit{submarine}) \xleftarrow{\{(1,1)\}} \mathit{sensor1}(X, \mathit{high}),$
  $\mathit{sensor2}(X, \mathit{high})$.
  "the target is a submarine if its readings on sensors 1 and 2 are both high".

Lakshmanan and Sadri, Uncertainty in DB & KB, ICDE 1998

5. Hybrid Approaches

5.1 Deductive IST (contd.):

- model theory, fixpoint theory, and proof theory.
- soundness and completeness results.
- easy extension for incorporating negation.
- *Query Optimization*: Conjunctive query containment
  - Q_1, Q_2 two conjunctive queries (i.e. rules) in DIST framework.
  - *qualitative*: $Q_1 \subseteq Q_2$ iff $\forall$ database D: $\forall$ tuple t, $t \in Q_2(D)$ holds in every "*situation*" in which $t \in Q_1(D)$ holds.
  - *quantitative*: $Q_1 \leq Q_2$ iff $\forall$ database D: the certainty of every tuple computed by Q_1 on input D is no more than the certainty of t computed by Q_2 on input D.
  - qualitative $\Rightarrow$ quantitative
  - how to characterize and test containment?
  - (when) do the two notions of containment coincide? what is the complexity of testing containment?
- *Query Optimization*: (Recursive) query program containment
  - do classical techniques for datalog extend?
  - uniform equivalence?

5. Hybrid Approaches

5.1 Deductive IST (contd.):

Major Results:

- Q_1 is qualitatively contained in Q_2 iff there is a containment mapping from Q_2 to Q_1 and the annotation of Q_1 implies that of Q_2.

- program containment/equivalence is undecidable: follows from result for classical datalog.

- uniform containment: $P_1 \subseteq^u P_2$ iff for every input DB D containing relations for all predicates – base and derived – $P_1(D) \subseteq P_2(D)$ in the qualitative sense.

- $P_1 \subseteq^u P_2$ iff $r \subseteq^u P_2$, for every rule r in P_1.

- $r \in P$ iff the chase procedure applied to the database $db(r)$ using program P succeeds: non-trivial extension of result for classical datalog.

- qualitative containment always coincides with quantitative containment:
  $Q_1 \subseteq Q_2$ iff $Q_1 \leq Q_2$;
  $P_1 \subseteq^u P_2$ iff $P_1 \leq^u P_2$.

5. Hybrid Approaches

5.1 Deductive IST (contd.):

Major Results (contd.):

- conjunctive query containment (CQC) is NP-complete when no. of information sources is a constant (frequent case).

- CQC is co-NP-complete in the number of information sources.

- CQC can be solved in PTIME if annotations are "-1-free".

- CQ optimization by eliminating subgoals is *no harder* than for classical datalog, even when no. of information sources is not a constant.

- similar remarks for uniform program containment.

5. Hybrid Approaches

5.2 Situation Theory (Lakshmanan 94):

- annotations include both belief and doubt.
- belief/doubt = a set of vectors; captures complex combination of basic scenarios under which phenomenon is believed/doubted.
- no specific assumptions between basic scenarios.
- knowledge about interdependence (independence, correlation, etc.) between scenarios captured by a mathematical program.
- interesting connections to modal logic; captures modalities more general than necessity and possibility.
- computation model: fixpoint or proof theoretic evaluation of answer tuples, followed by constraint solver solving a mathematical program, to compute certainties.
- many of the query optimization techniques for DIST can be extended.

5. Hybrid Approaches

5.2 Situation Theory (contd.):

Example 9: Some (relative) modalities expressible using situational framework:

- *necessity*: $\langle\{(\perp, \ldots, \perp)\}, \{\}\rangle$ – true under all known scenarios and false under no known scenarios.

- various degrees of *possibility*: $\langle S, T\rangle$, S, T any two sets of vectors, not both equivalent to "true" – true sometimes and false sometimes.

- *impossibility*: $\langle\{\}, \{(\perp, \ldots, \perp)\}\rangle$.

- *non-necessity*: $\langle S, T\rangle$, where S is any set of vectors and T is a set of vectors not equivalent to $\{\}$, i.e. to "false".

6. Generic/Parametric Framework (Lakshmanan and Shiri 97, Shiri 97)

Highlights:

- Goal: pave the way for QO for (D)DB + uncertainty (very little prior work).

- Problem: numerous frameworks with great diversity in formalisms, syntax, and semantics; different frameworks useful under diferent circumstances.

- Our approach: unify known frameworks into a generic framework, serving as their umbrella; develop QO techniques for the generic framework.

- generic framework – characterized by a set of *parameters*; diff. instantiations $\rightarrow$ different frameworks.

- prefer implication based framework over annotation based one for greater ease of implementation and for better complexity and continuity properties of IB frameworks.

- compatible syntax and semantics; sound and complete proof procedure and fixpoint query evaluation methods.

- complete characterization of conjunctive query containment under numerous conditions.

6. Parametric Framework (contd.)

The parameters:

- underlying lattice of certainties.
- family of conjunction functions: how to compute certainty of subgoals in rule body, given individual certainties?
- family of propagation functions: how to propagate certainty of rule body to rule head?
- family of disjunction functions: how to combine certainties of multiple derivations of a tuple?

Example 10:

$$\begin{array}{ll} r_1: & A \xleftarrow{\alpha_1} B, C. \\ r_2: & A \xleftarrow{\alpha_2} D. \\ r_3: & B \xleftarrow{\alpha_3} . \\ r_4: & C \xleftarrow{\alpha_4} . \\ r_5: & D \xleftarrow{\alpha_5} . \end{array}$$

6. Parametric Framework (contd.)

- certainty lattice – $\{0, 1\}$; $\alpha_i = 1$; propagation and conjunction functions associated with all rules – *min*; disjunction function associated with all predicates – *max*: classical DDB framework.

- certainty lattice – $[0, 1]$; $\alpha_i \in [0, 1]$; rest as above: Dubois et al.'s possibilistic LP framework.

- same as above, except propagation function associated with all rules is $\times$: van Emden's quantitative DDB framework.

- same as above, except $\times$ is the conjunction and propagation function everywhere, and f_d is the disjunction function everywhere, where $f_d(\alpha, \beta) = \alpha + \beta - \alpha \times \beta$; certainties viewed as probabilities: MYCIN.

6. Parametric Framework (contd.)

- certainty lattice – $\mathcal{C}[0,1] \times \mathcal{C}[0,1]$, i.e. all pairs of closed intervals over $[0,1]$; $\alpha_i \in \mathcal{T}$; conjunction function of r_1 – *ind*; propagation function of r_1 – *ign*; propagation function of r_2 – *pc*; disjunction function of A – *ign*: probabilistic DDB framework of Lakshmanan and Sadri.

- certainty lattice – 2^{L^k}, where $L = \langle\{1, -1, \top, \perp\}, \leq_t, \leq_k\rangle$ is a bilattice such that $-1 \leq_t \perp, \top \leq_t 1$, and $\top \leq_d -1, 1 \leq_d \perp$; each α_i is a set of k-vectors over L – so-called information source vectors; conjunction and propagation function for all rules – source conjunction (componentwise "AND" for all pairs of vectors); disjunction function for all predicates – source disjunction (union of vector sets): Deductive IST of Lakshmanan and Sadri.

- Situation theoretic framework of Lakshmanan: similar (exercise to reader).

6. Parametric Framework (contd.)

Example 11: ("WHO" example revisited.)

$$r_1: \ close(a,b) \xleftarrow{[0.9]} \ ; \qquad \langle \vee_{ign}, \wedge_{pc}, \_ \rangle$$

$$r_2: \ close(a,c) \xleftarrow{0.8} \ ; \qquad \langle \vee_{ign}, \wedge_{pc}, \_ \rangle$$

$$r_3: \ close(b,c) \xleftarrow{0.7} \ ; \qquad \langle \vee_{ign}, \wedge_{pc}, \_ \rangle$$

$$r_4: \ close(Y,X) \xleftarrow{1} close(X,Y); \qquad \langle \vee_{ign}, \wedge_{pc}, \_ \rangle$$

$$r_5: \ outbreak(d,a) \xleftarrow{1} \ ; \qquad \langle \vee_{pc}, \wedge_{pc}, \_ \rangle$$

$$r_6: \ connected(X,Y) \xleftarrow{1} close(X,Y); \qquad \langle \vee_{pc}, \wedge_{ind}, \_ \rangle$$

$$r_7: \ connected(X,Y) \xleftarrow{0.8} close(X,Z), connected(Z,Y); \qquad \langle \vee_{pc}, \wedge_{pc}, \wedge_{pc} \rangle$$

$$r_8: \ affected(D,A) \xleftarrow{0.7} outbreak(D,A); \qquad \langle \vee_{pc}, \wedge_{pc}, \_ \rangle$$

$$r_9: \ affected(D,A) \xleftarrow{0.8} close(A,A_1), affected(D,A_1); \qquad \langle \vee_{pc}, \wedge_{pc}, \wedge_{pc} \rangle.$$

- Considerably more flexibility compared to known implication based frameworks.
- simultaneous use of multiple modes.
- (ProbLog) implemented on top of CORAL deductive DBMS.

■

6. Parametric Framework (contd.)

Query Optimization:

- Specific problem studied – conjunctive query containment.

- annotated database: assigns a certainty from certainty lattice to each tuple in each base predicate.

- $Q_1 \leq Q_2$ iff on all annotated databases D, $Q_1(D) \leq Q_2(D)$, i.e. for each tuple t, the certainty in $Q_1(D)$ is no more than that in $Q_2(D)$.

- Let the queries be:
  $Q_1 : \ A \xleftarrow{\alpha_1} B_1, \ldots, B_n \ \ \langle f_d^1, f_p^1, f_c^1 \rangle.$
  $Q_2 : \ C \xleftarrow{\alpha_1} D_1, \ldots, D_m \ \ \langle f_d^2, f_p^2, f_c^2 \rangle.$

- Characterization: need to distinguish several cases.

6. Parametric Framework (contd.)

Query Optimization (contd.): **Summary of results**:

necessary condition: $Q_1 \leq Q_2 \Rightarrow \exists$ a containment mapping (c.m.) form Q_2 to Q_1; not always sufficient.

Class	Proviso	Type of c.m.	Comments
$f_c^2 = \otimes,\ f_d^1 = \oplus$	none	any	necessary & sufficient
$f_c^2 = \otimes,\ f_d^1 \succ \oplus$	$\mathbf{Q}_1$ has no repeated subgoals	variable onto	necessary & sufficient
$f_c^2 \prec \otimes,\ f_d^1 = \oplus$	none	subgoal 1-1	sufficient
	$\mathbf{Q}_2$ has distinct subgoals	any	necessary & sufficient
$f_c^2 \prec \otimes,\ f_d^1 \succ \oplus$	none	subgoal 1-1 & variable onto	sufficient
	$\mathbf{Q}_1$,$\mathbf{Q}_2$ have distinct subgoals	variable onto	necessary & sufficient
Identical parameters $f_c \prec \otimes,\ f_d = \oplus$	$\mathbf{Q}_1$ has distinct subgoals	subgoal 1-1	necessary & sufficient
Identical parameters $f_c \prec \otimes,\ f_d \succ \oplus$	$\mathbf{Q}_1$ has distinct subgoals	subgoal 1-1 & variable onto	necessary & sufficient

Summary of our main results on containment.

7. Summary and Future Directions

- solid body of theoretical and systems work done on (deductive/relational) databases with uncertainty.

- great variety of uncertainty formalisms, attachment mechanisms, syntax and semantics, with different ones appropriate for different applications.

- some important prototypes: pf-programs (Univ. of Maryland), DUCK system (Univ. of Munich), and ProbLog (Concordia Univ.); many more in the form of KB/expert systems, or non-monotonic rule systems (with uncertainty).

- basic principles, syntax, semantics, query processing issues, query optimization techniques – highlighted.

7. Summary/Future Directions (contd.)

- Need for considerable work in future: leveraging DB + uncertainty technology for strategic applications like data mining, image and other media databases, information retrieval and integration, etc.
  - knowledge discovery from DBs using Baysean nets (Heckerman 97); probabilistic DDBs (either implication or annotation based) – simple and robust framework in which to investigate this problem; more appropriate to a database setting; sclability issues.
  - use of probabilistic DB techniques for IR (Fuhr 94).
  - investigation of computational complexity – many classes of query programs comptable in PTIME in DB size (just as for datalog) – identified by Lakshmanan and Sadri and Lakshmanan and Shiri; need for more investigation for a complete understanding of landscape.
  - integration of statistical and subjective probabilities in a uniform framework (initial work Ng 93; formal foundation: Halpern 91); potential significance for data mining.
  - deploying DB + uncertainty technology in real SQL systems.