Volume : 2 | Issue : 1 | Jan 2013 • ISSN No 2277 - 8179

Research Paper

Computer Science

KEYWORDS :

Uncertainty Lineage Databases with Confidence Value

Mr. Vishal K. Pandya — HOD, Shri V J Modha College Of IT

Dr. Dhaval R. Kathiriya — Director, Information Technology Anand Agricultural University, Anand

ABSTRACT

Uncertainty Lineage Databases model with confidence [probability] values. In ULDBC every tuple alternative in a base relation is coordinate with a numerical confidence value, defining the likelihood of it being present in a possible instance. Those are complete with respect to sets of possible instances with probability distributions over them. The computation of confidence values on query results, using the confidence values f base data. Instead of computing confidence during query processing compute they afterwards based on lineage. This approach enables a wider space of query plans to be selected from and it permits selective computations when not all confidence values are needed. It provides confidence computations algorithms for single data items, as well as efficient batch algorithm to compute confidence for an entire relation or database.

FUNDAMENTAL

Uncertainty Lineage Databases models with confidence values we get ULDBC data model. Every alternative in ULDBC database is associated with a confidence value, that are supplied by user or any application on base data and confidence values for tuple alternatives in query results are inferred from base data's confidence values. Here query processing in a ULDBC consists of two components. First is Computing result tuple alternatives and lineage and another is Computing confidence values for the result.

The Computation of tuple alternative and lineage of ULDBC are identical to query processing on ULDB database. In computing the confidence value of a single tuple in the result of a query over ULDBC relation with confidence values can have exponential data complexity, the flow is that dependencies among the base data and confidence values contributing to the result data and confidences are lost during query execution. Once way to overcome this situation, bound the acceptable query plans to safe ones, which ensure correct spread of confidence values, when better plan for query is safe, then this way yields a very effectively process on query strategy. Some times the most efficient plan for a query is not safe in fact in extreme cases query's safe plan may be randomly inferior to the best unsafe one. The most efficient query plan can be selected when lineage tracking does not restrict, after that data computation confidence values can be computed at any time based on lineage.

If some values portions of a result data are not needed, they will never be computed since confidence value computation can be an unavoidable expensive operation in some cases, this can be very useful in work practice. At the most general level one can envision a query optimizer that explores a space of execution strategies encompassing efficient safe plans when they exist, lineage schemes and hybrids of the two ways for that. In this paper we are shown the ULDBC Data Model, and Query Processing.

ULDBC Data Model

Uncertainty Lineage Database (ULDB), Lineage enables simple and consistent representation of uncertain data, it correlates uncertainty in query results with uncertainty in the input data and query processing with lineage and uncertainty together presents computational benefits over treating them individual. Lineage identifies a data item's derivation, in terms of other data in the database or outside data sources; Function C gives confidence values of each symbol in the database. If the confidence value of a symbol is unknown or not specified, we say that it is NULL. We restrict our attention to the special case of E = ∅, i.e., there are no external symbols in the databases. We introduce an extremely simple and tiny conference database as a running example. Let base relation Attends (person,day) contain days on which people may attend a conference, and let Events (day,event) contain scheduled conference activities. Suppose our relations contain the following data, with confidence values shown to the right of each tuple.

Attends

ID	Person	Day	
11	A1-B1	Monday	0.8
12	A1-B1	Wednesday	0.7
13	C1	Wednesday	0.6

Events

ID	Day	Event	
21	Monday	Meeting	0.8
22	Tuesday	Outdoor	1.0
23	Wednesday	Seminar	0.9

Since Attends and Events are base relations, the lineage of their tuples is the tuple itself (e.g., λ(11) = 11). The above database has 25 = 16 possible instances corresponding to the choices of including or omitting each of the five tuples with confidence < 1. For example, the probability of the possible instance that includes all tuples from Attends and only tuple 22 from Events is $(0.8 \cdot 0.7 \cdot 0.6 \cdot (1 - 0.8) \cdot 1 \cdot (1 - 0.9))$. Intuitively, the probability of a possible instance is determined by the choices made for the base tuples, which are all independent of one another. The probability of picking a particular choice of alternatives is the product of the confidence values of all picked base alternatives times the product of "residual confidence" (1 − Ci) for all base tuples ti from which no alternative is picked. It gives a probability distribution over possible instances, i.e., the sum of probabilities of all possible instances is 1.

Query Processing

First of all define the semantics of query, where queries may be composed of any relational operators. We then show an example to illustrate that operator-by-operator confidence computation may give incorrect results. Finally, we present a basic lineage-based confidence computation algorithm that is improved upon in the subsequent sections.

(1) Semantics Of Query: Just as with ULDBs, the semantics of queries over ULDBcs are defined through possible instances. The result of performing a query Q on ULDBc D with possible instances D1, . . . , ,Dn and probabilities Pr(D1), . . . ,Pr(Dn) respectively, is a ULDBc D′ with possible instances D1 +Q(D1), . . . ,Dn +Q(Dn) with probabilities Pr(D1), . . . ,Pr(Dn) respectively. The data and lineage of query results over ULDBcs are computed identically as that for ULDBs.

(2) Safe/Unsafe Confidence transmission: Now suppose Attends and Events contain only the tuples involving Wednesday, i.e., two tuples in Attends and one in Events. At our imagined conference, events are also canceled if they have no attendees (in addition to weather cancelations), and we want to determine the likelihood of Wednesday's Seminar taking place. We use the query: EventLikelihood = δ(_event (Attends ◁▷ Events))

Volume : 2 | Issue : 1 | Jan 2013 • ISSN No 2277 - 8179

The correct answer including lineage and confidence is: EventLikelihood

ID	Event
51	Seminar

We consider two relationally equivalent plans for evaluating this query:

Plan 1: δ(_event(δ(_day(Attends)) ⊲⊳ Events))
Plan 2: δ(_event(Attends ⊲⊳ Events))

Plans are specified as an algebraic expression whose operators are evaluated one by one, with lineage propagated through the operators along with the data. A special "lineage structure" for operator-by-operator confidence computation according to the plans.

(3) Basic Confidence Computation- Algorithm:

```
1:  Conf(t) {
2:  if c(t) 6= NULL then return c(t);
3:  else return Compute-Conf(λ(t)); }
4:  Compute-Conf( f(t1, t2, . . . , tn) ) {
5:  all-base = true;
6:  for (i = 1; i ≤ n; i + +) {
7:  gi = λ(ti);
8:  if (gi 6= ti) then all-base = false; }
9:  if all-base then {
10: for (i = 1; i ≤ n; i + +): ci = c(ti);
11: return Eval(f, c1, c2, . . . , cn); }
12: else return Compute-Conf(f(g1, g2, ..., gn)); }
```

It provides pseudo code for our basic approach to computing the confidence of a tuple t using its lineage. This algorithm is the basis for improvements in the sections. Note that even this basic algorithm is an improvement over previous approaches in the sense that it does not limit the query plans we can use for data computation. Algorithm uses two primitive functions for ULDBc: λ(t) returns the lineage formula for t, and c(t) returns the current confidence value of t stored in the database. Recall from ULDBc data model we assume c(t) is NULL whenever the confidence of t has not been computed, and that confidences are always present for base data. The main function Conf(t) in Algorithm returns t's confidence if already computed. Otherwise it calls the recursive function Compute- Conf(f) on t's lineage formula f. This function computes a final boolean formula B by recursively expanding the current formula until all variables refer to base tuples. Then, the probability of B is computed and returned using the confidence values of the base tuples. We assume a function Eval, which takes a boolean formula over independent variables and the probabilities of each variable as input and computes the probability of the Boolean formula. For Example Consider the two query plans for EventLikelihood from safe Plan 1 and unsafe Plan 2. Suppose we want to compute the confidence of the single projected result tuple (Seminar). Since lineage of tuples in query results always refers to tuples in the query input, in the query results for both plans we have λ(Seminar) = (12∨13)∧23, where 12, 13, and 23 are base tuples. Thus Pr(Seminar) = Pr((12∨13)∧23). Using Pr(12) = 0.7, Pr(13) = 0.6, and Pr(23) = 0.9, we get Pr(Seminar) = 0.792, which is the correct answer. In general, computing the probability of an arbitrary boolean formula of independent events (our function Eval) is known to have exponential worst-case complexity, unless P=NP. If approximate answers suffice then we could employ Monte-Carlo simulations for the final computation, as proposed.

This is introduced the ULDBc data model, which extends the ULDB data model with confidence values. The query processing over ULDBc database is that of confidence computation for the result. I observed that lineage significantly eases the process of confidence computation.

REFERENCE

(1) R. M. Karp and M. Luby. "Monte-Carlo Algorithms for Enumeration and Reliability Problems". In: Proc. of Symp. on Foundations of Computer Science (FOCS), 1983. | (2) M. R. Garey and D. S. Johnson. "Computers and Intractability". W. H. Freeman and Company, 1979. | (3) C. Re, N. Dalvi, and D. Suciu. "Efficient Top-k Query Evaluation on Probabilistic Data". In: Proc. of Intl. Conf. on Data Engineering (ICDE), 2007. | (4) N. Dalvi and D. Suciu. "Efficient Query Evaluation on Probabilistic Databases". In: Proc. of Conf. on Very Large Data Bases (VLDB), 2004. |