Global Journal of Computer Science and Technology
Software & Data Engineering
Volume 12 Issue 12 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Query Join Processing Over Uncertain Data for Decision Tree Classifiers

By V. Yaswanth Kumar & G. Kalyani
JNTU, Kakinada

Abstract - Traditional decision tree classifiers work with the data whose values are known and precise. We can also extend those classifiers to handle data with uncertain information. Value uncertainty arises in many applications during the data collection process. Example sources of uncertainty measurement/quantization errors, data staleness, and multiple repeated measurements. Rather than abstracting uncertain data by statistical derivatives, such as mean and median, the accuracy of a decision tree classifier can be improved much if the complete information of a data item is used by utilizing the Probability Density Function (PDF). In particular, an attribute value can be modelled as a range of possible values, associated with a PDF. The PDF function has only addressed simple queries such as range and nearestneighbour queries. Queries that join multiple relations have not been addressed with PDF. Despite the significance of joins in databases, we address join queries over uncertain data. We propose semantics for the join operation, define probabilistic operators over uncertain data, and propose join algorithms that provide efficient execution of probabilistic joins especially threshold. In which we avoid the semantic complexities that deals with uncertain data. For this class of joins we develop three sets of optimization techniques: item-level, page-level, and index-level pruning. We will compare the performance of these techniques experimentally.

GJCST-C Classification: *H.2.8*

Strictly as per the compliance and regulations of:

Query Join Processing Over Uncertain Data for Decision Tree Classifiers

V. Yaswanth Kumar [α] & G. Kalyani [σ]

Abstract - Traditional decision tree classifiers work with the data whose values are known and precise. We can also extend those classifiers to handle data with uncertain information. Value uncertainty arises in many applications during the data collection process. Example sources of uncertainty measurement/quantization errors, data staleness, and multiple repeated measurements. Rather than abstracting uncertain data by statistical derivatives, such as mean and median, the accuracy of a decision tree classifier can be improved much if the complete information of a data item is used by utilizing the Probability Density Function (PDF). In particular, an attribute value can be modelled as a range of possible values, associated with a PDF. The PDF function has only addressed simple queries such as range and nearest-neighbour queries. Queries that join multiple relations have not been addressed with PDF. Despite the significance of joins in databases, we address join queries over uncertain data. We propose semantics for the join operation, define probabilistic operators over uncertain data, and propose join algorithms that provide efficient execution of probabilistic joins especially threshold. In which we avoid the semantic complexities that deals with uncertain data. For this class of joins we develop three sets of optimization techniques: item-level, page-level, and index-level pruning. We will compare the performance of these techniques experimentally.

I. Introduction

Classification rules can be represented as below. Consider the information about the insurance company information.

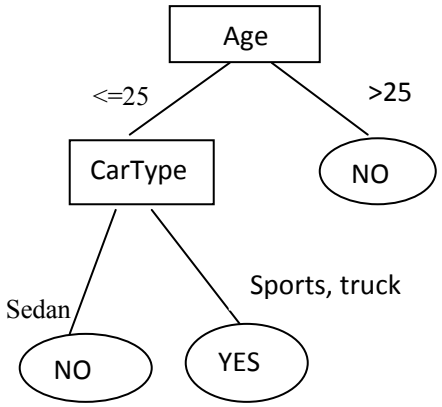
Insurance info (age: **integer**, cartype: **string**, highrisk: **boolean**) if age is between 16 and 25 and cartype is either sports or truck, then the risk is high.

Consider the information about the insurance customers.

Age	Cartype	Highrisk
23	Sedan	False
30	Sports	False
36	Sedan	False
25	Truck	True
30	Sedan	False
23	Truck	True
30	Truck	False
25	Sports	True
18	Sedan	False

Author α : Student, DVR & Dr HS MIC College of Technology, Kanchikacherla, Krishna(dt).
Author σ : Assoc. professor, DVR & Dr HS MIC College of Technology, -Kanchikacherla, Krishna(dt).

Trees that represent classification rules are called classification trees or decision trees.



Data uncertainty arises naturally in many applications due to various reasons. We briefly discuss three categories here: dimension errors, data mustiness, and repeated dimensions.

a) **Dimension Errors**: Data obtained from measurements by physical devices are often imprecise due to dimension errors.
b) **Data mustiness**: In some applications, data values are continuously changing and recorded information is always out of date.
c) **Repeated dimensions**: Perhaps the most common source of uncertainty comes from repeated dimensions. For example, a patient's body temperature could be taken multiple times during a day.

Type-1 Probabilistic Relations

Type-1 uncertainty refers to confidence if a tuple belongs to a relation or not. Consider the table represents a part of my personal address book. It is not really likely that my address book contains the phone number of the Dutch Queen, where it
is very likely that the address book contains the phone number of one of my fellow students, Ruud van Kessel.

Type-2 Probabilistic Relations

With Type-2 uncertainty, the value of the key-attribute is deterministic but values of other attributes in the relation may be uncertain. Table shows a relation which depicts where Kings and Queens of different countries around the world live. There is no uncertainty about the country where the King or Queen lives. Since the attribute values of the field "town" for Queen Beatrix

and King Carl XVI Gustaf represents uncertainty, it is not possible to tell in which village they live with complete certainty, based on this list.

If the probability that the join pair meets the join condition exceeds the threshold, it is included in the result, otherwise the pair is not included. This threshold can either be user specified or a system parameter. The tuple pairs when their probabilities exceed a certain threshold as Probabilistic Threshold Join Queries (PTJQ) we focus on threshold joins and develop various techniques for the efficient (in terms of I/O and CPU cost) algorithms for PTJQ. In particular, we develop three pruning techniques:

Name	Country	Town
BeatrixvanOranje	The Netherlands	The Hague/0.9 Amsterdam/0.1
Carl XVI Gustaf	Sweden	Stockholm/0.5 Malmo/0.5

(a) Type-1 Probabilistic Relations

Name	Phone number	Probability
Beatrixvan Oranje	+31701234567	0.01
Ruud van Kessel	+316 12345678	0.99

(b) Type-2 Probabilistic Relations

1) **item-level pruning**, where two uncertain values are pruned without evaluating the probability.
2) **page-level pruning**, where two pages are pruned without probing into the data stored in each page.
3) **index-level pruning**, where all the data stored under a subtree is pruned. Two useful types of join operations specific to uncertain attributes: value join (v-join) and distribution join (d-join). V-join is a natural extension of the join operation on deterministic data. The PDF (probability Density Function) can be used to calculate the Range of values to an attribute which contains attribute uncertainity. PDF also calculates probability of matching uncertain tuples present in different relations while performing join operation. Each join-pair is associated with a probability to indicate the likelihood that the two tuples are matched. We use the term **Probabilistic Join Queries** (PJQ). For join conditions over uncertain data, the result is generally not boolean, but probabilistic.

II. Related Work

The model for managing uncertain data is proposed in moving-object environments and in sensor networks. Recently, the Trio System has been proposed to handle such uncertainty. Another representation of data uncertainty is a "probabilistic database", where each tuple is associated with a probability value to indicate the confidence of its presence. Probabilistic databases have also been recently extended to semi-structured data and XML. Probabilistic queries are classified as value-based (return a single value) and entity-based (return a set of objects). Probabilistic join queries belong to the entity-based query class.

Aggregate value queries and nearest neighbor evaluation algorithms are presented. To our best knowledge, probabilistic join queries have not been addressed before. Also these works did not focus on the efficiency issues of probabilistic queries. Although examine the issues of query efficiency, their discussions are limited to range queries. There is a rich vein of work on interval joins, which are usually used to handle temporal and one-dimensional spatial data. Different efficient algorithms have been proposed, such as nested-loop join, partition-based join, and index-based join. Recently the idea of implementing interval joins on top of a relational database. All these algorithms do not utilize probability distributions within the bounds during the pruning process, and thus potentially retrieve many false candidates. We demonstrated how our ideas can be applied easily to enhance these existing interval join techniques.

III. Implementation

The INLJ (IndexedNestedLoopJoin) algorithm can recover I/O performance by organising the pages in a Tree structure. Let R and S denote the two relations that are being joined, and assume that R has fewer tuples than S. If neither join input has an index on the joining attribute, the indexed nested loops join algorithm first builds an index on the smaller input R.

The index is built by extracts the *key–pointerinformation* for each tuple. The key–pointer information is then spatially sorted based on the MBR. We can develop the efficient query join processing technique by the following sequence of operations.

a) Data Refinement

Take any Real-world Data which is possible to containing Uncertainity. Clean the data i.e. removing unnecessary data for Our project and Represent the most appropriate Data.

b) Formulating Range values using PDF function (Probability Density Function)

PDF summarizes how odds/probabilities are distributed among the events that can arise from a series of trials.By using PDF function we can replace the uncertainity values as Ranges.

c) Similarity matching between Uncertainty tuples

(By using probabilistic Joining Queries) Calculate the probability of joining the two uncertainity

tuples. Each join-pair is associated with a probability to indicate the likelihood that the two tuples are matched.

d) Removing Uncertainity by using INLJ

Although uncertainty tables can be used to improve the performance of page-based joins, they do not improve I/O performance, simply because the pages still have to be loaded in order to read the uncertainity tables.In INLJ we can use Interval Index.

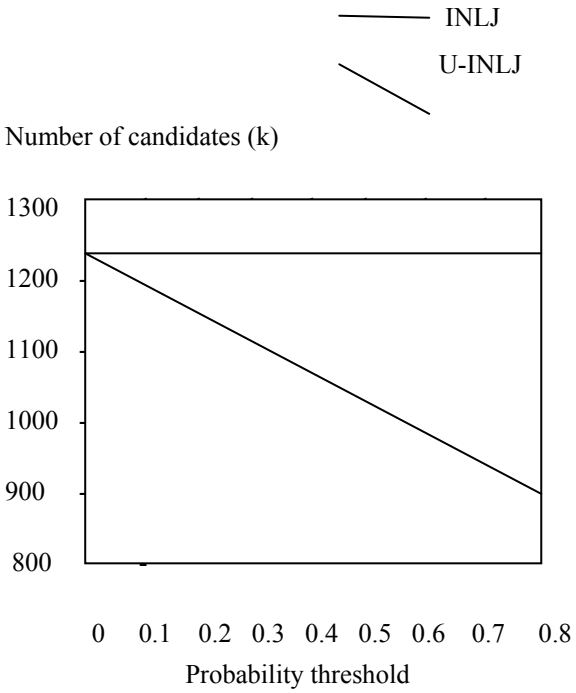
Conceptually, each tree node still has an uncertainty table, but now each uncertainty interval in a tree node becomes a Minimum Bounding Rectangle (MBR) that encloses all the uncertainty intervals stored in that MBR. Page-level pruning now operates on MBRs instead of uncertainty intervals.

e) Construct the Decision Tree for Query processing

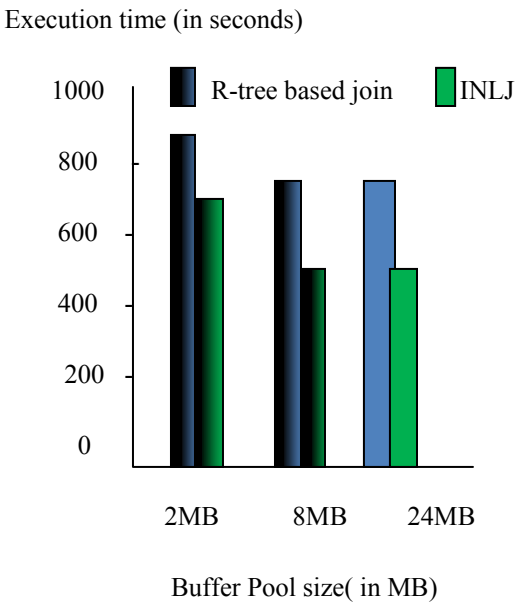
Splitting of an attribute depends on the attribute selection measures (Information gain, Gain ratio, Gini Index).Higher value of an attribute can be selected as splitting one. In this way the output Can be represented in the Decision Tree form by classifying the result into different classes.

IV. Performance

Index-Level Pruning The above problem can be alleviated by organizing intervals with an index. shows that both **INLJ** and **U-INLJ** have a much better performance in *Npair* than **BNLJ** and **U-BNLJ.**



In the above Graph, the comparison between page-level join algorithms with the Index-level join algorithms (INLJ and U-INLJ). In the Index-level join algorithms whenever the Threshold increases the output candidate pairs are Reduced. So, we can join the tables based on most similarity tuples. This leads to high performance in the Results, Next we can compare the Execution time between R-tree based join algorithm and INLJ(Indexed-Nested Loop Join) algorithm are compared in the below graph. The Horizontal row specifies the size of the Datasets and vertical row specifies the Execution time in seconds. In all different type of Datasets the Execution time of INLJ is Better than R-tree based join algorithm.



Finally I can prefer the Indexed-Nested Loop join algorithm as a Probability Threshold Joining Algorithm for Removing the Uncertainity while Joining of multiple table where the joining attribute has uncertain values. so, the Result of joining is efficient and we get the close to Exact Results.

V. Conclusion

Uncertainity management is the mounting topic in Data mining in recent times. In this paper we identify the situation of maintaining uncertain attributes present in the database relations. We suggest a method for getting better join processing of relations in requisites of I/O cost which are having uncertainity attributes present. In this paper we propose the implementation of INLJ, which is capably handle the uncertain values when compared to the earlier uncertainty handlings.

References références referencias

1. J. R. Quinlan, “Induction of decision trees,” Machine Learning, vol. 1, no. 1, pp. 81–106, 1986.
2. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993, ISBN 1-55860-238-0.
3. J. Chen and R. Cheng, “Efficient evaluation of imprecise location dependent queries,” in ICDE. Istanbul, Turkey: IEEE, 15-20 Apr. 2007, pp. 586–595.

4. M. Chau, R. Cheng, B. Kao, and J. Ng, "Uncertain data mining: An example in clustering location data," in PAKDD, ser. Lecture Notes in Computer Science, vol. 3918. Singapore: Springer, 9–12 Apr. 2006,pp. 199–204.
5. R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter, "Efficient indexing methods for probabilistic threshold queries over uncertain data," in VLDB. Toronto, Canada: Morgan Kaufmann, 31 Aug.–3 Sept. 2004, pp. 876–887.
6. R. Cheng, D. V. Kalashnikov, and S. Prabhakar, "Querying imprecise data in moving object environments," IEEE Trans. Knowl. Data Eng.,vol. 16, no. 9, pp. 1112–1127, 2004.
7. T. M. Mitchell, Machine Learning. McGraw-Hill, 1997, ISBN 0070428077 .
8. R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *Proc. SIGMOD*,2003.
9. R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J.Vitter. Efficient indexing methods for probabilistic threshold queries over uncertain data. In *Proc. VLDB*, 2004.
10. D. Zhang, V. Tsotras, andB.Seeger.Efficient temporal join processing using indicies. In *Proc. ICDE*, 2002.