Volume 2, Issue 12, December-2015, pp. 904-907 ISSN (O): 2349-7084

International Journal of Computer Engineering In Research Trends

Available online at: www.ijcert.org

Query Aware Determinization of Uncertain Objects

[1]P.Jhancy, [2]K.Lakshmi ,[3]Dr.S.Prem Kumar

[1]*Pursuing M.Tech, CSE Branch, Dept of CSE*

[2]*Assistant Professor, Department of Computer Science and Engineering*

[3]*Professor & HOD, Department of computer science and engineering, G.Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh, India.*

Abstract:- The main aim of this paper is to think about the trouble of determining probabilistic data to allow such data to be stored in legacy systems that agree only deterministic input. Probabilistic data may be produced by mechanized data analysis methods such as entity resolution, information extraction, and speech processing etc. The target is to make a deterministic depiction of probabilistic data that optimizes the excellence of the end-application built on deterministic data. We discover such a determinization problem in the background of two dissimilar data processing jobs – selection and triggers queries. Here approaches such as thresholding or top-1 selection usually used for determinization lead to suboptimal presentation for such applications. As an alternative, we expand a query-aware strategy and demonstrate its rewards over existing solutions through a complete empirical evaluation over real and synthetic datasets.

Keywords – uncertain data, query workload, data quality, branch and bound algorithm.

1. INTRODUCTION

Through the arrival of cloud computing and the increase of web-based applications, users frequently store their data in various active web applications. Repeatedly, user data is generated mechanically through a variety of signal processing, data analysis techniques before being stored in the web applications. For example, modern cameras will have the features such as vision analysis to produce tags such as landscape, portrait, indoors, outdoors, night mode etc. And also have the feature of microphones for users to speak out a expressive sentence which is then processed by a speech recognizer to generate a set of tags to be associated with the photo. The photo along with set of tags can be streamed in real-time via wireless connectivity to Web applications such as Flicker. It is an image hosting and video hosting website, and web services suite .It is a popular website for users to share and insert personal photographs. This paper will consider the problem of mapping probabilistic data into the corresponding deterministic representation as the determinization problem. Many solutions to the determinization problem can be planned. Here we use the strategy called Top-1. In this we choose the most feasible value / all the probable values of the attribute with non-zero probability, correspondingly. For example, a speech recognition system that produces a single answer/tag for each declaration can be viewed as using a top-1 strategy. Here we explore how to determinate answers to a query over a probabilistic database.

2. RELATED WORK

Many advanced probabilistic data models were used in proposed systems. Here the centre of attention however was determinizing probabilistic objects, such as speech output and image tags, for which the probabilistic attribute model meet the requirements. It is to be noted that determining probabilistic data stored in more advanced probabilistic representation such as tree structures is also used. Several related research

efforts that contract with the problem of selecting terms to index document for document retrieval. A term-centric pruning method explains in keeps top postings for each term according to the individual score impact that each posting would have if the term appeared in an temporary search query. Here we propose a scalable term selection for text classification, is nothing but which is based on coverage of the terms. The centre of these research efforts is on significance – that is, getting the right set of terms that are most relevant to this paper. In our problem, a set of probably appropriate terms and their significance to the document are already specified by other data processing techniques. Thus, our objective is not to explore the significance of terms to documents, but to select keywords from the given set of terms to represent the paper, such that the quality of answers to triggers or queries is optimized. The main advantage of our proposed system is it will resolve the problem of determinization by reducing the expected cost of the answer to queries. Here we develop an efficient algorithm that achieves near-optimal quality. The algorithms which we are advice are very capable and reach high-quality results that are very close to those of the optimal solution.

3. DETERMINIZATION FOR THE COST-BASED METRIC

3.1. Branch and Bound Algorithm

As an alternative of performing a brute-force enumeration, we can make use of a faster branch and bound (BB) technique. The move towards will discovers response sets in a greedy fashion so that answer sets with lower cost tend to be discovered first. A branch-and-bound algorithm consists of a systematic enumeration of candidate solutions by means of state space search: the set of candidate solutions is notion of as forming a rooted tree with the full set at the root. The algorithm investigates branches of this tree, which symbolize subsets of the solution set. Before specifying the candidate solutions of a branch, the branch is checked against upper and lower estimated bounds on the optimal solution, and is leftover if it cannot produce a better solution than the best one found so far by the algorithm.

The algorithm depends on the capable estimation of the lower and upper bounds of a region/branch of the search space and approaches comprehensive enumeration as the size (n-dimensional volume) of the region tends to zero. Table 1 précis the notations we will utilize to demonstrate the future BB algorithm.

3.2. Outline of the BB algorithm

The benefit of a unique model for all types of discrete optimization problems is that a general purpose Branch and Bound method is available. The two basic stages of a general Branch and Bound method: • Branching: splitting the problem into sub problems • Bounding: calculating lower and/or upper bounds for the objective function value of the sub problem The branching is performed in the following algorithm by separating the current subspace into two parts using the integrality requirement. Using the bounds, unpromising sub problems can be eliminated. LP-relaxation is formed by discarding the integer requirements. For binary variables, add bounds 0 # x # 1. i The LP-minimum gives a lower bound for the ILP-minimum: min f # min f . LP ILP In the following Branch and Bound method, the best IP-solution given by the solved sub problems is stored as an incumbent solution, a record holder. The incumbent objective value is an upper bound for the minimum value. A list P of candidate sub problems is maintained and updated. A sub problem is fathomed (totally solved) and removed from the list, when • it has an integer solution that is best so far and becomes the new incumbent solution, or, • its optimum LP-solution objective is worse than the current incumbent value, or, • the LP-problem is infeasible. Notation: f* = minimum value of the objective function for the current LP-sub problem. f = incumbent minimum value, given by a feasible integer solution min xmin

Our general method for branch and bound algorithms involves modeling the solution space as a tree and then traversing the tree exploring the most promising sub trees first. This will continuous until either there are no sub trees into which to advance break the problem, or we have inwards at a point where, if we continue, only inferior solutions will be found. Let us have a look on a general algorithm for branch and bound searching is presented in figure 1.


```
search(A,B,best)
pre:   A=solution space tree
       B=vertex in A
      best= The solution which obtained as best so far

post : best =The solution which obtained as best so far after searching
subtree rooted at B
If B is a complete solution more optimum than Best then set best= B
generate the children of B

compute bounds for vertices in subtrees of children X1.....XK = feasible
children with good lower bound for i = 1 to k
If X i has a promising upper bound then search(A, X,best)
```

Fig1. Branch and bound searching

Let us look at this technique more directly and discover that what is required to explain problems with the branch and bound method.

We first need to define the objects that formulate the original problem and possible solutions to it.

Problem instances. For the knapsack problem this would consist of two lists, one for the weights of the items and one for their values. Here we need an integer for the knapsack capacity. For chromatic numbers (or graph coloring), this is just a graph that could be accessible as an adjacency matrix, or better yet, an adjacency edge list.

Solution tree. This must be an ordered edition of the solution search space, perhaps containing partial and infeasible solution candidates as well as all feasible solutions as vertices. For knapsack we built a depth-first search tree for the coupled integer programming problem with the objects ordered by weight. In the chromatic number solution tree we offered partial graph colorings with the first k nodes colored at level k. These were ordered so that if a node had a particular color at a vertex, then it remained the same color in the sub tree.

Solution candidates. For knapsack, a list of the items placed in the knapsack will be sufficient. Chromatic numbering involves a list of the colors for each vertex in the graph. Other than, it is a little more complex since we use partial solutions in our search, so we must indicate vertices yet to be colored in the list. An necessary rule to be followed in essential solution spaces for branch and bound algorithms as follows.

If a solution tree vertex is not part of a feasible solution, then the sub tree for which it is the root cannot contain any feasible solutions.

This rule assures that if we cut off search at a vertex due to impracticality, then we have not unnoticed any optimum solutions.

Currently, we present the definitions for bounds used in the above algorithm.

Lower bound at a vertex. The Smallest value of the intention function for any node of the sub tree rooted at the vertex.

Upper bound at a vertex. The largest value of the intention function for any node of the subtree rooted at the vertex.

For chromatic number we used the number of colors for the lower bound of a partial or complete solution. The lower bound for knapsack vertices was the current load, while the upper bound was the possible weight of the knapsack in the subtree.

Branch-and-bound may furthermore be a base of various heuristics. For instance, one may desire to prevent branching while the gap among the upper and lower bounds becomes smaller than a certain threshold. This is act as a solution and can greatly reduce the computations required. This type of solution is particularly applicable when the cost function used is noisy or is the result of statistical estimates and so is not known exactly but rather only known to lie within a range of values with a specific probability. The main advantage of Branch & Bound algorithm is it finds an optimal solution (if the problem is of limited size and enumeration can be done in reasonable time).

4. CONCLUSIONS AND FUTURE WORK

In this paper we have measured the problem of determinizing uncertain objects to permit such data to be stored in pre-existing systems, for example Flicker, that capture only deterministic input. The plan is to generate a deterministic demonstration that optimizes the excellence of answers to queries that implement over the deterministic data representation. Planned efficient determinization algorithms that are advice of extent faster than the details based optimal solution but attain approximately the same quality as the optimal solution. The Future Enhancement for this work is to explore determinization methods in the context of request, in which users are also involved in regain objects in a grade order.

REFERENCES

[1] D. V. Kalashnikov, S. Mehrotra, J. Xu, and N. Venkatasubramanian, "A semantics-based approach for speech annotation of images," TKDE'11.
[2] J. Li and J. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," PAMI'03.
[3] C. Wangand, F. Jing, L. Zhang, and H. Zhang, "Image annotation refinement using random walk with restarts," ACM Multimedia'06.
[4] B. Minescu, G. Damnati, F. Bechet, and R. de Mori, "Conditional use of word lattices, confusion networks and 1-best string hypotheses in a sequential interpretation strategy," ICASSP'07.
[5] R. Nuray-Turan, D. V. Kalashnikov, S. Mehrotra, and Y. Yu, "Attribute and object selection queries on objects with probabilistic attributes,"

ACM TODS'11.
[6] J. Li and A. Deshpande, "Consensus answers for queries over proba-bilistic databases," PODS'09.
[7] M. B. Ebarhimi and A. A. Ghorbani, "A novel approach for frequent phrase mining in web search engine query streams," CNSR '07.

[8] S. Bhatia, D. Majumdar, and P. Mitra, "Query suggestions in the absence of query logs," SIGIR '11.
[9] C. Manning and H. Schutze, Foundations of Statistical Natural Lan-guage Processing. MIT Press, 1999.
[10] D. V. Kalashnikov and S. Mehrotra, "Domain-independent data cleaning via analysis of entity-relationship graph," ACM TODS'06.
[11] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis, "On-line index selection for shifting workloads," SMDB'07.
[12] P. Unterbrunner, G. Giannikis, G. Alonso, D. Fauser, and D. Kossmann, "Predictable performance for unpredictable workloads," VLDB'09.
[13] R. Cheng, J. Chen, and X. Xie, "Cleaning uncertain data with quality guarantees," PVLDB'08.
V. Jojic, S. Gould, and D. Koller, "Accelerated dual decomposition for map inference," ICML '1

