ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Determinization of Uncertain Objects Based On Query Response

Miryala Harish Kumar
M.Tech Student
Vignanabharathi Institute of Technology And Sciences,
Ghatkesar.

V.Sridhar Reddy
Associate Profesor
Vignanabharathi Institute of Technology And Sciences,
Ghatkesar.

ABSTRACT:

This paper considers the problem of determinizing probabilistic data to enable such data to be stored in legacy systems that accept only deterministic input. Probabilistic data may be generated by automated data analysis/enrichment techniques such as entity resolution, information extraction, and speech processing. The legacy system may correspond to pre-existing web applications such as Flickr, Picasa, etc. The goal is to generate a deterministic representation of probabilistic data that optimizes the quality of the end-application built on deterministic data. We explore such a determinization problem in the context of two different data processing tasks—triggers and selection queries. We show that approaches such as thresholding or top-1 selection traditionally used for determinization lead to suboptimal performance for such applications. Instead, we develop a query-aware strategy and show its advantages over existing solutions through a comprehensive empirical evaluation over real and synthetic datasets.

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.

EXISTING SYSTEM:

- Many approaches to the determinization problem can be designed. Two basic strategies are the Top-1 and All techniques, wherein we choose the most probable value / all the possible values of the attribute with non-zero probability, respectively.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

- For instance, a speech recognition system that generates a single answer/tag for each utterance can be viewed as using a top-1 strategy. Another strategy might be to choose a threshold τ and include all the attribute values with a probability higher than τ .
- Existing system works address a problem that chooses the set of uncertain objects to be cleaned, in order to achieve the best improvement in the quality of query answers.
- There are several related research efforts that deal with the problem of selecting terms to index document for document retrieval. A term-centric pruning method described in existing system retains top postings for each term according to the individual score impact that each posting would have if the term appeared in an adhoc search query.

DISADVANTAGES OF EXISTING SYSTEM:

- Often lead to suboptimal results.
- They explore how to determinize answers to a query over a probabilistic database. In contrast, we are interested in best deterministic representation of data (and not that of a answer to a query) so as to continue to use existing end-applications that take only deterministic input.
- Their goal is to improve quality of single query, while ours is to optimize quality of overall query workload.

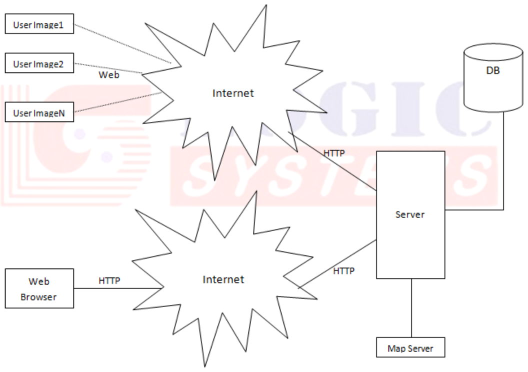
PROPOSED SYSTEM:

- In this paper, we study the problem of deteminizing datasets with probabilistic attributes (possibly generated by automated data analyses/enrichment).
- Our approach exploits a workload of triggers/queries to choose the “best" deterministic representation for two types of applications – one, that supports triggers on generated content and another that supports effective retrieval.
- Interestingly, the problem of determinization has not been explored extensively in the past. The most related research efforts are, which explore how to give deterministic answers to a query (e.g. conjunctive selection query) over probabilisitc database.
- Unlike the problem of determinizing an answer to a query, our goal is to determinize the data to enable it to be stored in legacy deterministic databases such that the determinized representation optimizes the expected performance of queries in the future. Solutions cannot be straight forwardly applied to such a determinization problem.

ADVANTAGES OF PROPOSED SYSTEM:

- We introduce the problem of determinizing probabilistic data. Given a workload of triggers/queries, the main challenge is to find the deterministic representation of the data which would optimize certain quality metrics of the answer to these triggers/queries.
- Solves the problem of determinization by minimizing the expected cost of the answer to queries.
- We develop an efficient algorithm that reaches near-optimal quality.
- The proposed algorithms are very efficient and reach high-quality results that are very close to those of the optimal solution. We also demonstrate that they are robust to small changes in the original query workload.

SYSTEM ARCHITECTURE:



ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

DETERMINIZATION FOR THE COST-BASED METRIC

Branch and Bound Algorithm

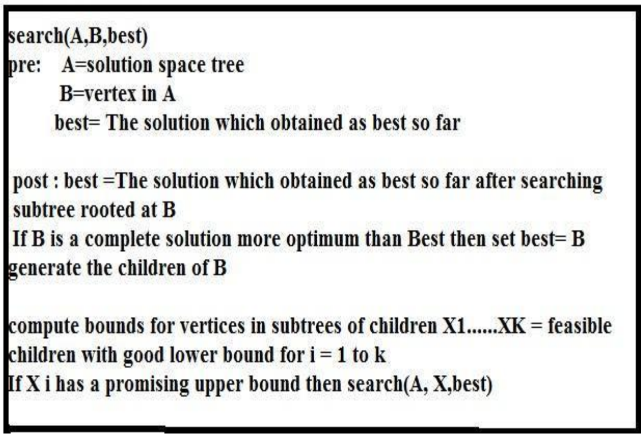
As an alternative of performing a brute-force enumeration, we can make use of a faster branch and bound (BB) technique. The move towards will discovers response sets in a greedy fashion so that answer sets with lower cost tend to be discovered first. A branch-and-bound algorithm consists of a systematic 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.

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.



```
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).

CONCLUSIONS

We have considered problem of determinizing uncertain objects in order to organize and store such data in already existing systems example Flickr which only accepts deterministic value. Our aim is to produce a deterministic depiction that optimizes the quality of answers to queries/triggers that execute over the deterministic data representation .As in future work, we plan to perform project on efficient Determinization algorithms that are orders of scale faster than the enumeration based best solution but achieves almost the same excellence as the optimal solution and search Determinization techniques as per the application context, wherein users are also involved in retrieving objects in a ranked order.

REFERENCES

[1] Jie Xu, Sharad Mehrotra," Query Aware Determinization of Uncertain Objects" ,IEEE

Transactions on knowledge and data engineering, VOL. 27, NO. 1, January 2015.

[2] J. Li and J. Wang, "Automatic linguistic indexing of pictures by a statistical modeling approach," IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 9, pp. 1075–1088, Sept. 2003.

[3] C. Wangand, F. Jing, L. Zhang, and H. Zhang, "Image annotation refinement using random walk with restarts," in Proc. 14th Annu. ACM Int. Conf. Multimedia, New York, NY, USA, 2006.

[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," in Proc. ICASSP, 2007.

[5] Jian Pei, Ming Hua," Query Answering Techniques on Uncertain and Probabilistic Data" In VLDB, pages 1151- 1154, 2006.

[6] Umesh Gorela1, Bidita Hazarika2, Abhinesh Tiwari3, Priti Mithari," Survey on Query Aware Strategy for Determining Uncertain Probabilistic Data", in (IJSETR), Volume 4, Issue 10, October 2015 3510

[7] R. Nuray-Turan, D. V. Kalashnikov, S. Mehrotra, and Y. Yu, "Attribute and object selection queries on objects with probabilistic attributes," ACM Trans. Database Syst., vol. 37, no. 1, Article 3, Feb. 2012.

[8] V. Jojic, S. Gould, and D. Koller, "Accelerated dual decompositionfor MAP inference," in Proc. 27th ICML, Haifa, Israel, 2010.

[9] D. Sontag, D. K. Choe, and Y. Li, "Efficiently searching for frustrated cycles in map inference," in Proc. 28th Conf. UAI, 2012.

[10] I. Bordino, C. Castillo, D. Donato, and A. Gionis, "Query similarity by projecting the query-flow graph," in Proc. 33rd Int. ACM SIGIR, Geneva, Switzerland, 2010.

[11] P.Jhancy, K.Lakshmi ,Dr.S.Prem Kumar," Query Aware Determinization of Uncertain Objects" in ijcert Volume 2, Issue 12, December-2015, pp. 904-907

[12] R. Nuray-Turan, D. V. Kalashnikov, S. Mehrotra, and Y. Yu,"Attribute and object selection queries on objects with probabilistic attributes," ACM Trans. Database Syst., vol. 37, no. 1, Article 3, Feb. 2012.

[13] B. Sigurbjornsson and R. V. Zwol, "Flickr tag recommendation based on collective knowledge," in Proc. 17th Int. Conf. WWW, New York, NY, USA, 2008.

[14] A. Rae, B. Sigurbjornsson, and R. V. Zwol, "Improving tag recommendation using social networks," in Proc. RIAO, Paris, France, 2010.

[15] D. Carmel et al., "Static index pruning for information retrieval systems," in Proc. 24th Annu. Int. ACM SIGIR, New Orleans, LA, USA, 2001.