© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Query Aware Determinization of Uncertain Objects by Indexing

Sakthi Priyadharshan V *, Dr. Sivasubramaniam S

Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai, India

ABSTRACT

This project considers the problem of determinizing uncertain data data to enable to facilitate the data data storage 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 represent already existing web applications such as Flickr, Picasa, etc. The idea is to create a deterministic representation of probabilistic data that improves the quality of the user end-application built on deterministic data. We study and solve such a determinization issue in the context of two different data processing tasks—triggers and selection queries. It is known that methods 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.

Keywords: Optimization, Image Identification, Indexing, Categorization

I. INTRODUCTION

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

How Data Mining Works?

Transactions and analytical systems have been growing in the higher technology level at an unprecedented rate. Data Mining helps bridge the gap between the two. Data mining software checks patterns and relationships from different sources including transactional and structured data. Several types of analytical software are available: statistical, machine learning, and neural networks. Generally, any of four types of relationships are sought:

Classes: Stored data is used to locate data in predetermined groups. For example, a restaurant chain could mine customer purchase data to determine when customers visit and what they typically order. This information could be used to increase traffic by having daily specials.

- Clusters: Data items are grouped according to logical relationships or consumer preferences. For example, data can be mined to identify market segments or consumer affinities.
- Associations: Data can be mined to identify associations. The beer-diaper example is an example of associative mining.
- Sequential patterns: Data is mined to anticipate behaviour patterns and trends. For example, an outdoor equipment retailer could predict the likelihood of a backpack being purchased based on a consumer's purchase of sleeping bags and hiking shoes.

II. IMPLEMENTATION

In the first module we develop the data owner module. It is infeasible when the number of tags is large, which is the case in multiple data sets. Hence, we develop a branch-and-bound algorithm to solve EDCM approximately. We empirically demonstrate that our approximate solution reaches the quality that is very close to the exact one, while being orders of magnitude more efficient. First, in this module, Data Owner New user should register then only the data owner can Login to Application. After the Registration is completed, then admin approved user only to login. Then only user can login to home page. User first we search the image file. We use any of queries that processed to get output in Grid view. Grid view shows related search items. User selects that particular image then Download it. Also user can change our Password and Update their Details. Home Page has about the project Details.

Naive enumeration-based algorithm which finds the exact solution to the problem. This naive algorithm is, however, exponential in the number of tags associated with the object. In this module, we also develop the Admin functionalities. Admin verify the register details and approve the Users. Upload the Images to Cloud and give different titles for the images. Admin views User Details and can lock the particular User. Admin views the Upload files Details. Chart view shows the user performance in this application. Chart shows the File searching and execution time. Chart shows the number of the searching images. Chart shows the number of images user downloaded to Cloud. Chart shows the efficient and more accessing images details. All over user actions and performance view an admin.

The Branch and bound Algorithm performance can be heavily improved further by performing query-level improvements and optimizations. In any definite sequence node Xs of a node S, The cost information can be exactly determined. Instead of checking each and every item in the dataset we can automatically use branch and bound technique. The approach discovers answer sets in a greedy fashion so that answer sets with lower cost tend to be discovered first. Each node v in the tree corresponds to a partial tag selection where decisions of whether to include certain tags in the answer set or not has been made. We capture the partial selection of tags using the concept of sequence defined

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. We test our solution over several synthetic query workloads where parameters of workloads follow certain distributions.

III. INPUT AND OUTUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

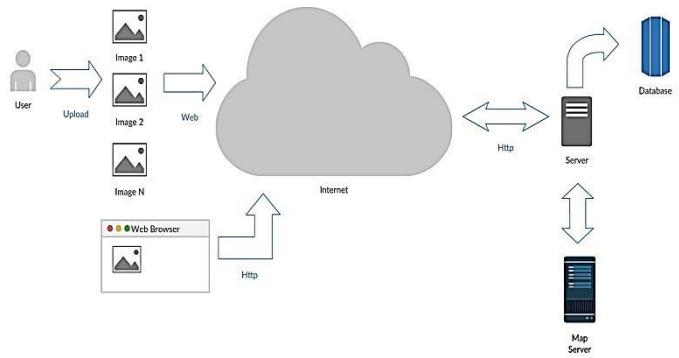
- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of

screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting information.
3. Create document, report, or other formats that contain information produced by the system. The output form of an information system should accomplish one or more of the following objectives.
4. Convey information about past activities, current status or projections of the Future.
5. Signal important events, opportunities, problems, or warnings.
6. Trigger an action.
7. Confirm an action.



The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional Test: Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems / Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.
Test strategy and approach
Field testing will be performed manually and functional tests will be written in detail.
Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

IV. CONCLUSION

In this work we have considered the problem of determinizing uncertain objects to enable such data to be stored in pre-existing systems, such as Flickr, that take only deterministic input. The goal is to generate a deterministic representation that optimizes the quality of answers to queries/triggers that execute over the deterministic data representation. We have proposed efficient determinization algorithms that are orders of magnitude faster than the enumeration based optimal solution but achieve almost the same quality as the optimal solution. As future work, we plan to explore determinization techniques in the context of applications, wherein users are also interested in retrieving objects in a ranked order.

V. REFERENCES

[1] D. V. Kalashnikov, S. Mehrotra, J. Xu, and N. Venkatasubramanian, "A semantics-based approach for speech annotation of images," IEEE Trans. Knowl. Data Eng., vol. 23, no. 9, pp. 1373–1387, Sept. 2011.

[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] 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.

[6] J. Li and A. Deshpande, "Consensus answers for queries over probabilistic databases," in Proc. 28th ACM SIGMOD-SIGACTSIGART Symp. PODS, New York, NY, USA, 2009.

[7] M. B. Ebarhimi and A. A. Ghorbani, "A novel approach for frequent phrase mining in web search engine query streams," in Proc. 5th Annu. Conf. CNSR, Frederlcton, NB, Canada, 2007.

[8] S. Bhatia, D. Majumdar, and P. Mitra, "Query suggestions in the absence of query logs," in Proc. 34th Int. ACM SIGIR, Beijing, China, 2011.

[9] C. Manning and H. Schutze, Foundations of Statistical Natural Language Processing, Cambridge, MA, USA: MIT Press, 1999.

[10] D. V. Kalashnikov and S. Mehrotra, "Domain-independent data cleaning via analysis of entity-relationship graph," ACM Trans. Database Syst., vol. 31, no. 2, pp. 716–767, Jun. 2006.