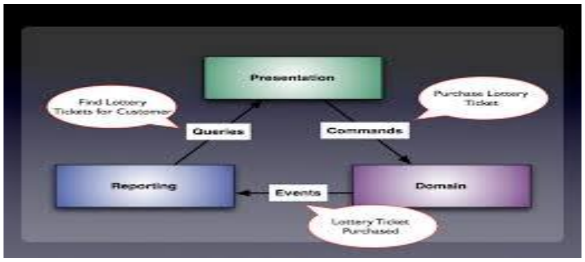
QDA Query Driven Approach to Entity Resolution

R.Venkataramanaiah[1], Dr. E.Kesavulu Reddy[2]
[1]*Student, Department of Computer Science, S.V.University, Tirupati. – INDIA*
[2]*Assistant Professor, Department of Computer Science, S.V.University, Tirupati. – INDIA*

***Abstract*- This paper explores "on-the-fly" data cleaning in the context of a user query. A novel Query-Driven Approach (QDA) is developed that performs a minimal number of cleaning steps that are only necessary to answer a given selection query correctly. The comprehensive empirical evaluation of the proposed approach demonstrates its significant advantage in terms of efficiency over traditional techniques for query-driven applications. The significance of data quality research is motivated by the observation that the of data-driven technologies such as decision support tools, data exploration, analysis, and scientic discovery tools is closely tied to the quality of data to which such techniques are applied. It is well recognized that the outcome of the analysis is only as good as the data on which the analysis is performed. That is why today organizations spend a substantial percentage of their budgets on cleaning tasks such as removing duplicates,correcting errors, and lying missing values, to improve data quality prior to pushing data through the analysis pipeline.Given the critical importance of the problem, many efforts, in both industry and academia, have explored systematic approaches to addressing the cleaning challenges.**

1. INTRODUCTION

This paper addresses the problem of query-aware data cleaning, wherein the needs of the query dictates which parts of the data should be cleaned. Query-aware cleaning is emerging as a new paradigm for data cleaning to support today's increasing demand for (near) real-time analytical applications. Modern enterprises have access to potentially limitless data sources, e.g., web data repositories, social media posts, clickstream data, etc. Analysts usually wish to integrate one or more such data sources (possibly with their own data) to perform joint analysis and decision making. As a result of merging data from different sources, a given real-world object may often have multiple Representations, resulting in data quality challenges. In this paper, we focus on the Entity Resolution (ER) challenge [16], [19], [29].



Entity resolution is a well-known problem and it has received significant attention in the literature over the past few decades. A thorough overview of the existing work in this area can be found in surveys. We classify the ER techniques into two categories as follow: Generic ER. A typical ER cycle consists of several phases of data transformations that include: normalization, blocking, similarity computation, clustering, and merging, which can be intermixed. In the normalization phase, the ER framework standardizes the data formats. The next phase is blocking which is a main traditional mechanism used for improving ER efficiency. The primary motivation of this paper is query on online data. A key concept driving the QDA approach is that of vestigiality. A cleaning step (i.e., call to the resolve function four pair of records) is called vestigial (redundant) if QDA can guarantee that it can still compute a correct answer without knowing the outcome of this resolve. We formalize the concept of vestigiality in the context of a large class of SQL selection queries and develop techniques to identify vestigial cleaning steps.

2 LITERATURE SURVEY

We firstly review the literatures concerning Customer satisfaction, and then the query driven approach problem in cloud computing. To estimate the service demand of a service provider, it is critical to measure its customer satisfaction. In business management, there have been many specialists who focus on the researches of the definition of customer satisfaction. The concept of customer satisfaction is firstly proposed by Cardozo in 1965 and he believed that high customer satisfaction produces purchase behavior again.

In recent years, cloud computing has become a booming service industry. How to increase ease of access is an important issue for cloud service providers. Many works have been done to research this issue. There are some researches focusing on the Query Driven Approach problem of the service providers. Chaisiri took into consideration the uncertainty of the customers demand, and proposed a stochastic programming model with two-stage recourse to solve the Query Driven Approch problem for the service providers. There are some works in cloud computing which consider customer satisfaction. Chen adopted utility theory leveraged from economics and developed an utility model for measuring customer satisfaction in cloud. In the service model, consumer satisfaction is relevant to two factors: ease of access and less response time. They assumed that consumer satisfaction is decreased with longer time to request query processing and longer response time.

However, the user satisfaction here is defined as how much the requirements specified in a request are satisfied. Morshedlou and Meybodi defined the users' satisfaction level based on expected value of user's utility that an user attaches to a certain monetary amount. However, the existing formulas measuring customer satisfaction of cloud computing cannot properly reflect the definition of customer satisfaction, and they did not take into account user's psychological differences.

To address this problem, we use the definition of query driven approachment for entity resolution leveraged from economics and develop a formula to measure customer satisfaction in cloud. And then, how cloud configuration affects customer satisfaction and how customer satisfaction and customer security affects the performance of cloud service providers are analyzed. Based on these works, a query driven approach problem considering customer satisfaction with secured data maintenance is formulated and solved such that the optimal configuration is obtained.

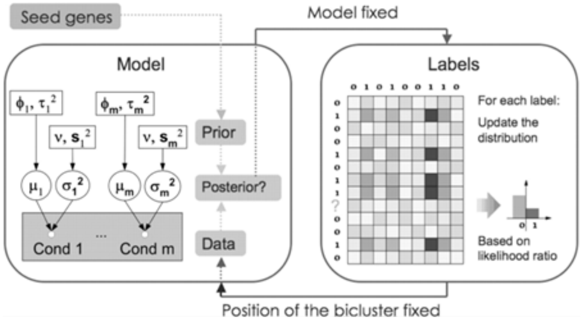
3. ALGORITHMS

3.1 General modeling framework

The core of the probabilistic framework resembles that of Sheng et al. (2003), the main ingredients being column-wise probability distributions and hidden labels (g) for the genes and (c) for the conditions to indicate bi-cluster membership. Assume each column j of the (n × m) expression data matrix X represents an experimental condition and each row i represents a gene. Expression values xij for which both the corresponding gene and condition are assigned to the bi-cluster (gi = 1 and cj = 1) are then modeled by the bi-cluster distribution (superscript 'bcl') of the corresponding condition. All other expression values are modeled by the background distribution (superscript 'bgd') of their corresponding condition. The use of condition-wise background distributions allows compensating for between-array differences in expression level variance.

Of course, we do not know in advance which genes, conditions and expression values xij belong to the bi-cluster; the model therefore depends on hidden gene (g) and condition labels (c). Figure 1 shows a conceptual scheme of the framework.

Fig1



For the column-wise (condition-wise) statistical probability distributions, we use Gaussian distributions with parameters θj = (μj, σj) and conjugate Normal – Inverse χ2 priors. The (full conditional) label probabilities are given by Bernoulli distributions with Beta priors.

3.1.1 Prior distributions

One of the strengths of the Bayesian probabilistic framework is the possibility of using well-chosen prior distributions on the model parameters. We utilize conjugate Normal–Inverse χ2 priors on the column-wise Gaussian probability distributions (Gelman et al., 2004):

If xij indicates the expression level of gene i in experimental condition j, the corresponding distributions for bi-cluster and background can then be written as

The parameterization of the above formulas is justified by the interpretation of the corresponding full conditional distributions in Section 3.1.2 (where parameters κ and ν are equivalent to a number of prior observations). In addition to the prior distributions on the model parameters, Beta priors B (ξg1, ξg0) and B(ξc1, ξc0) on the parameters of the Bernoulli label distributions can be used to specify a prior believe that a gene or condition belongs to the bi-cluster (bi-cluster size).

We postpone the discussion of parameter choices for the priors to Section 3.3.

3.1.2 Full conditional distributions

As illustrated in Supplementary Material File 1, the full conditional distributions for the gene labels are

Bernoulli distributions:

In this expression, ξg1 and ξg0 are parameters of the Beta prior distribution B(ξg1, ξg0) on the probability that a gene belongs to the bi-cluster, n is the total number of genes, θ the set of model parameters (μ, σ), ψ the total set of hyper parameters (κ, ν, s, ϕ, ξg0, ξg1, ξc0, ξc1) and ||g≠i||1 the one norm of the current (binary) gene label vector, except for label i. In other words, the second factor depends on the number of genes currently in the bi-cluster as well as prior knowledge on the bi-cluster size. The first factor corresponds to the likelihood ratio of bi-cluster versus background model.

A similar model holds for the full conditional distribution of the condition labels c (see Supplementary Material File 1).

Given the choice of the Normal – Inv χ2 priors, the full conditional distributions for the model parameters are given by

The prior parameters κ and ν can be interpreted as ‘pseudo counts’ or the number of ‘prior observations’ for the estimates of the mean and variance, respectively. The resulting estimates for means (variances) are weighted means of the observed sample mean (variance) and the prior mean (variance). For brevity, we omitted the formulas for the background model parameters, which are similar. Details on the derivations can be found in Supplementary Material File 1.

3.1.3 Joint posterior distribution

Given the data and a particular choice for the prior distributions, the joint posterior distribution p(θ, g, c | X, Ψ) indicates the statistically most interesting simultaneous assignments for the labels and the model parameters. Unfortunately, it is not possible to use this joint posterior distribution directly because we are unable to describe it analytically. We next discuss a strategy to detect its local maxima using information on the corresponding full conditional probability distributions only.

3.2 Algorithm

Conditional Maximization or CM (Gelman et al., 2004) consists of alternatively maximizing a set of full conditional distributions. These maximization steps are repeated until the procedure converges to a local mode of the corresponding joint probability distribution. Figure 1 illustrates the alternating procedure; the full conditionals and joint posterior distributions were introduced in Sections 3.1.2 and 3.1.3, respectively. In the query-driven context under study, convergence to local modes in the posterior landscape will often be sufficient because the query is introduced through strong priors on the model parameters (see Section 3.3). Strong priors act as a powerful zoom lens to magnify interesting regions in the likelihood landscape. Therefore, they tend to give rise to rather simple posterior distributions, even when the corresponding likelihood landscape is complex and contains many modes. Furthermore, the knowledge represented by the seed genes can be used for clever initialization (Supplementary Material File 1).

4. PRELIMINARY INVESTIGATION

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. And the main objective of this investigation is to establish efficient communication systembetween the client and server by providing the cloud space by the admin of the cloud.

Cloud admin can provide the cloud space to the author based on a security key which is generated by the admin after submitting the complete details of the author. Cloud admin can also provide the access to the user or client based on his query selection among the possible choices of queries. Before accessing the file from the cloud they must have a private key provided by the cloud admin.

It can be possible by providing the enhanced communication between the client and server.

Some of the basic principles and rules are defined in order to establish the client-server architecture.

Client Server:

With the varied topic in existence in the fields of computers, Client Server is one, which has generated more heat than light, and also more hype than reality. This technology has acquired a certain critical mass attention with its dedication conferences and magazines. Major computer vendors such as IBM and DEC, have declared that Client Servers is their main future market. A survey of DBMS magazine revealed that 76% of its readers were actively looking at the client server solution. The growth in the client server development tools from $200 million in 1992 to more than $1.2 billion in 1996.

Client server implementations are complex but the underlying concept is simple and powerful. A client is an application running with local resources but able to request the database and relate the services from separate remote server. The software mediating this client server interaction is often referred to as MIDDLEWARE.

The typical client either a PC or a Work Station connected through a network to a more powerful PC, Workstation, Midrange or Main Frames server usually capable of handling request from more than one client. However, with some configuration server may also act as client. A server may need to access other server in order to process the original client request.

5. MODULES DESIGN ANALYSIS

There are mainly four types of modules defined after performing the careful analysis of a system existence. They are as follows

1. User module
2. Admin module
3. Chart module
4. Query driven approach module

5.1 User module:

User module, the new user should register Application form, before enter the particular site, after login, user should create the profile for that particular login user, user can search any author details, they can view also related author details.

5.2 Admin module:

Admin is a super user. they can view all the author details. Admin can view the chart based on author details, admin can view the user details. A typical ER cycle consists of several phases of data transformations that include: normalization, blocking, similarity computation, clustering, and merging which can be intermixed.

5.3 Query driven approach module:

Process of Query Driven Approach. Mentioned below is process of query driven data warehousing approach − When a query is issued to a client side, a metadata dictionary translates the query into the queries, appropriate for the individual heterogeneous site involved.

5.4 Chart module:

Query driven approach project we will generate a chart based on author details , so user admin can easily find out which author can realize the most number of articles .

6. THE QUERY DRIVEN APPROACH PROBLEM

These rules can be discovered from existing high quality data such as master data or manually identified data. Inspired by the swoosh method, each cluster is then merged into a composite record via a merge function. Finally a traditional ER method, denoted by T-ER, can be applied to identify the new data set. Moreover, in order to identify more records, the current ER result can be used as the training data to discover new ER-rules. The training data can also be Obtained by using techniques, such as relevant feedback, crowd sourcing and knowledge extraction from the web. Therefore, with the accumulated information, ER-rules for more entities can be discovered.

Invalid rule. A rule r is invalid if there exist records that match LHS(r) but do not refer to RHS(r) . Invalid rules might be discovered when the information of entities is not comprehensive. For example, suppose the training data set involves the records. The rule r:

(name ¼“wei Wang”)^(coa 2“zhang”)) e1 can be generated. For o31, it matches LHS(r) but does not refer to e1. Therefore, r is an invalid rule.

Incomplete rule set. An ER-rule set R of entity set E is incomplete if there are records referring to entities in E that are not covered by R. Both the incomprehensive information of entities and continuous changes of entity features would cause a rule set become incomplete. To solve these problems, we develop some methods to identify candidate invalid rules and candidate useless rules and discover new effective ER-rules.

7. CONCLUSION

In this paper, we have studied the Query-Driven Entity Resolution problem in which data is cleaned \on-the-y" in the context of a query. We have developed a query-driven entity resolution framework which efficiently issues the minimal number of cleaning steps solely needed to accurately answer the given selection query. We formalized the problem of query-driven ER and showed empirically how certain cleaning steps can be avoided based on the nature of the query.

This research opens several interesting directions for future investigation. While selection queries (as studied in this paper) are an important class of queries on their own, developing QDA techniques for other types of queries (e.g. Joins) is an interesting direction for future work. Another direction is developing solutions for efficient maintenance of a database state for subsequent querying.

Future enhancement:

In future, we will future develop our algorithm in the following aspects: This research opens several interesting directions for future investigation. While

Selection queries (as studied in this paper) are an important class of queries on their own, developing QDA techniques for other types of queries (e.g., joins) is an interesting direction for future work. Another direction is developing solutions for efficient maintenance of a database state for subsequent querying.

REFERENCES

Related Articles

[1] Face recognition with radial basis function (RBF) neural networks Meng Joo Er; Shiqian Wu; Juwei Lu; Hock Lye Toh

[2] Competitive learning with floating-gate circuits D. Hsu; M. Figueroa; C. Diorio

[3] A state-of-the-art survey on software merging T. Mens

[4] A modified Chi2 algorithm for discretization F.E.H. Tay; Lixiang Shen

[5] Constructive feed forward ART clustering networks. I A. Baraldi; E. Alpaydin

[6] Cluster number selection for a small set of samples using the Bayesian Ying-Yang model Ping Guo; C.L.P. Chen; M.R. Lyu

[7] Constructive feed forward ART clustering networks. II A. Baraldi; E. Alpaydin

[8] The multi synapse neural network and its application to fuzzy clustering Chih-Hsiu Wei; Chin-Shyurng Fahn

[9] Learning sensory maps with real-world stimuli in real time using a biophysically realistic learning rule M.A. Sanchez-Montanes; P. Konig; P.F.M.J. Verschure

[10] A study of concurrency control in real-time, active database systems A. Datta; S.H. Son