Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Query Optimization for Intelligent Data Mining to Obtain Abstract User Views

**Sachin Thapa*, Praveen Singh, Shubhendu Dutta, Manish Kumar**
BVDUCOE, Pune, Maharashtra,
India

***Abstract— several web applications provide interface for dynamic query formation for better user experience and efficient data mining. However, several applications fail to measure the need of user session constraints. We propose a self-learning dynamic query builder which can measure session constraints to enhance the efficiency of dynamic query formation. The system is capable to initially target an abstract user dataset and in further stages, abstract the user session constraints to throw user-centric data abstraction. A user session not only provides information against anticipated dataset but also for undesirable data.Traditional algorithms measure the likeness of user queries. However, integrating a ranking constraint for plummeted data enhance the efficiency of dynamic queries. In our work, we have consumed session constraints to calculate the degree of acceptance and rejection of query components.***

***Keywords— Data Mining, Query Optimization, Database Views, RDBMS, Distributed Database***

## I. INTRODUCTION

Information with its multidimensional mapping is getting denser for application users to refine meaningful data. To provide an effective data mining for responsive user experience, dynamic query forms has been widely accepted. An algorithmic solution [1]to overcome the challenges by minimizing the tedious manual effort required to browse data has been proposed. The research presented a study of system performance based on application oriented queries. Another research work [2] proposed a dynamic query form mechanism with the featureof dynamically generating query forms. The model use a probabilistic model to refine form components based on user session preferences. It has been established that dynamic forms provides higher success rate.Another notable work proved that probabilistic tactics can be castoff to scheme smart data entry forms that endorse superior data mining. The model, USHER [3], employs the standard to dynamically adjust the form based on received values. After entry, the systemspontaneouslyrecognizes contextualized faults and reworks to verify their correctness. A prior research concludes that merelyexhibitingcallback forms as a flat list may not be appropriate [4]. However none of the models focused on the rejected query components which has been comparatively less referred by the user requests. Without consideration of this dataset, initially unattended data may get further backlogged resulting in complete denial as a part of query. Users may never be able to get the insight of such datasets. Hence, we propose a model which takes into account the rejection session components which can be used for improving the query prediction at later stages.

## II. SYSTEM ARCHITECTURE

The proposed system works around four step evaluation process as shown in Fig 1.

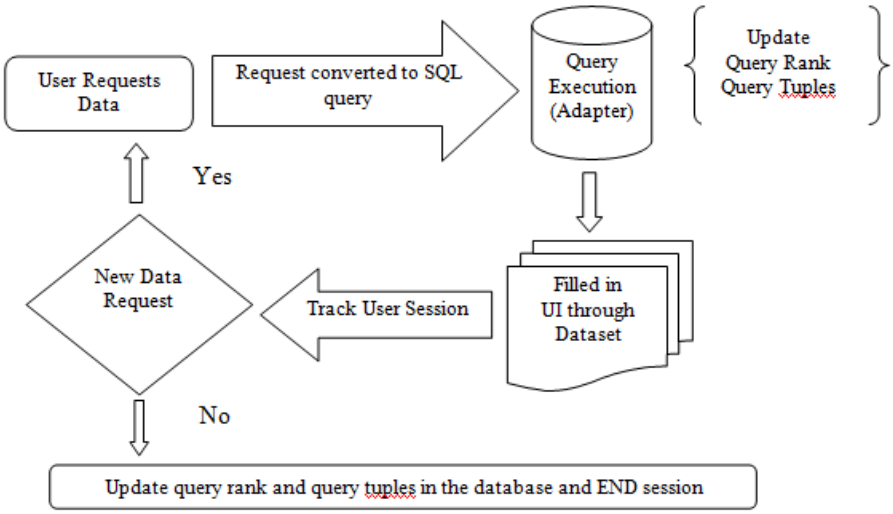


Fig 1: System Architecture

 

At the adapter level, the query ranks are reworked depending on the fired queries. The ranking will be based on relativity of accepted and rejected components which will provide training dataset for the system. The dataview responsible for user interface interaction provides the query resultant values. In addition, it also triggers client side cookies for tracking the user feedback against the responded data. It can provide insights on accepted components as well as rejected components. Irrespective of the users response, the tuples and queries will be re-ranked based on acceptance or rejection of the queries, hence, the training phase has been considered as an iterative process in the proposed system. Client level debugging can expose the APIs which needs to be abstracted in the future work. The front end has been implemented through JavaScript (backbone) which consumes WCF services and forwards the request to the business logic layer which further redirects it to the data access layer. The data-access layer handles the query execution and prepares the response data transfer object. The proposed model has been tested for CRUD operations on the assessment database.

## III. PROPOSED MODEL

Every user request is traditionally considered as an opportunity for analysing the user behaviour. However, it's also the counter-response for server response, which can serve the same purpose. For the sake of simplicity, we initiated pre-defined user session and targeted it to the data access layer. Based on the user feedback, the ranking of tuples and queries was calculated recursively. Table no 1 shows the various parameters obtained during the multiple iterations of session tracking.

TABLE I PARAMETERS OF TEST RUN

| ITERATION | NO OF COLUMNS | ACCURACY OF PROPOSED SYSTEM | TIME (IN NANOSECONDS) | NO OF CONSTRAINTS USED IN QUERY |
|---|---|---|---|---|
| 1 | 31 | 46% | 0.0034 | 12 |
| 2 | 31 | 47% | 0.0031 | 17 |
| 3 | 31 | 45% | 0.0039 | 11 |
| 4 | 31 | 44% | 0.0034 | 13 |

For any given accepted query tuple $f(na)$:

$$\mathrm{f(na)} = \sum_{\mathrm{n=1}}^{\mathrm{n}} \frac{(\mathrm{z}-1)^{(\mathrm{n}-1)}(\mathrm{a})}{(\mathrm{z}-\mathrm{a})} (\mathrm{z})^{\mathrm{n}}$$

where, z is the no of table tuples and a is the tuples considered for $f(na)$. Similarly, for a rejected tuple $f(nr)$:

$$\mathrm{f(nr)} = \frac{1}{2\mathrm{f(na)}} \frac{\mathrm{f(z)}}{\mathrm{z}-\mathrm{a}}$$

Both the functions provides a probabilistic value for consideration or rejection of tuples in the future transactions.In our test cases, query forms are able to produce different queries by altered inputs and attain diverse precisions which can be compared with the traditional approaches (Fig 2).

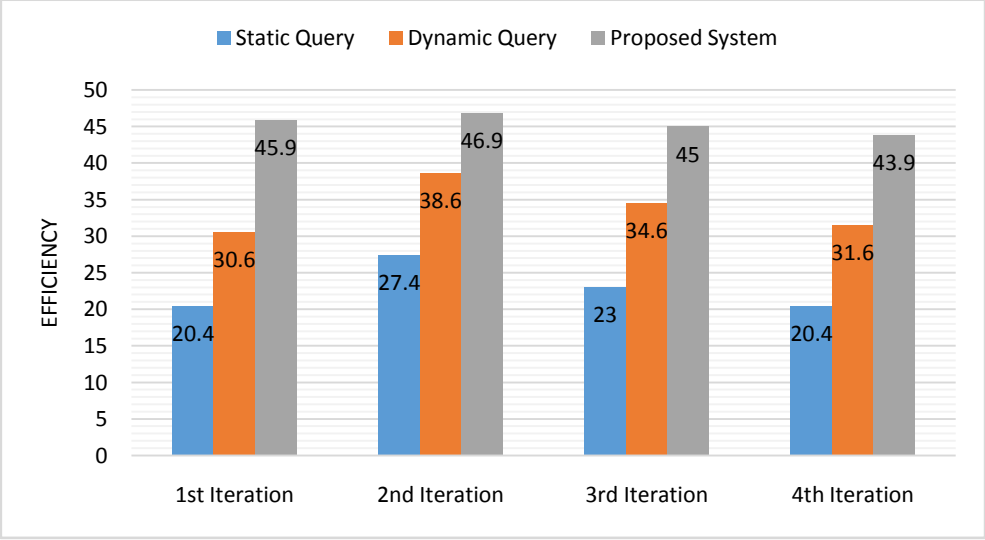


Fig 2: Efficiency comparison in our test dataset (in percentage)

The proposed algorithm when tested on the same dataset as that of dynamic forms and static queries, proved to be significantly providing better prediction of user preferences. It is worth mentioning that the results may be recorded better with application oriented datasets.After each iteration, the system implicitly produces rankings of form components and the user then supplements the anticipated form components in the query form.

 

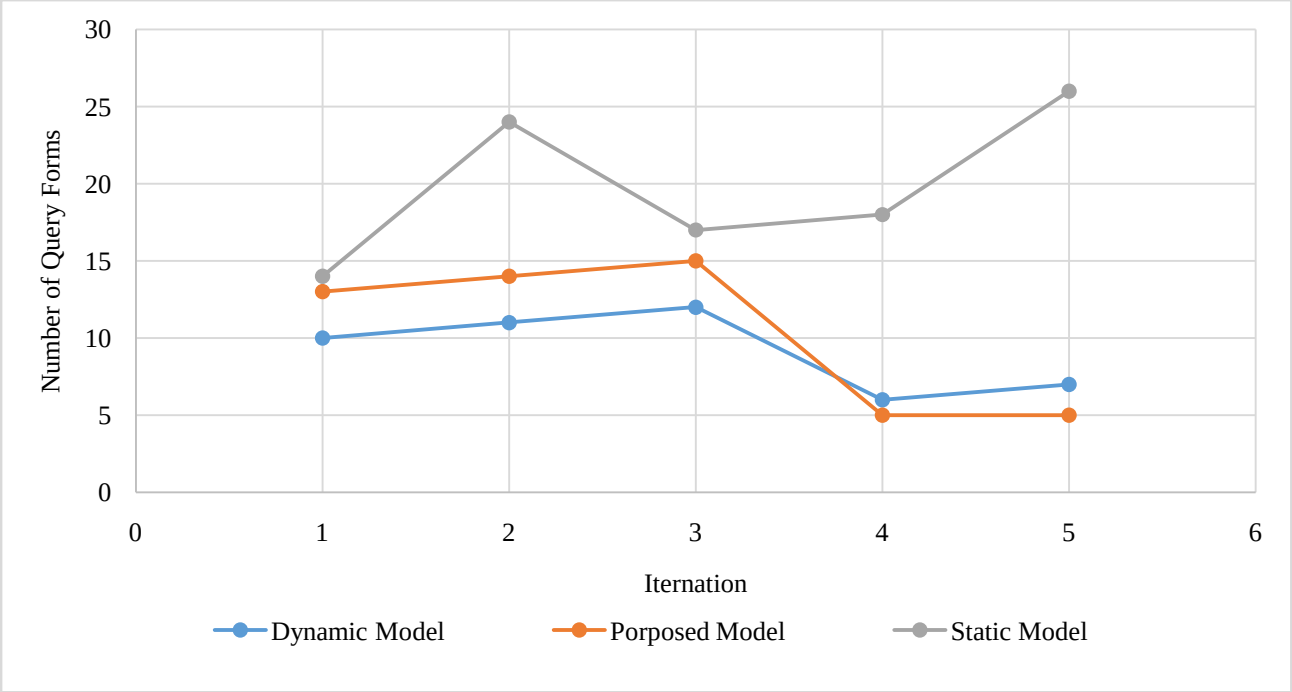


Fig 3: Comparison of Query forms

Correspondingly, a query form is dynamically enhanced till a set of user accepted query results are achieved.The ranking of query components makes it effortless for its users to alterthe dynamic query forms.

## IV. CONCLUSIONS

In our proposed system, we have modified the traditional approach of dynamic query forms. This enhances the accuracy of the field prediction for dynamic queries by also considering the rejection parameters. It will help rank the query constraints in a more rational way for personalized sessions. As a part of the future enhancements, we will upgrade the system to remove the bottlenecks which may arise due to computation overhead of query constraints on every session request. Our system is conceptually better but needs more optimization on implementation scenario.

## REFERENCES

[1] Magesh Jayapandian and H.V. Jagadish, *"Automating the Design and Construction of Query Forms"*,IEEE transactions on knowledge and data engineering, VOL. 21, NO. 10, October 2009.

[2] Liang Tang, Tao Li, Yexi Jiang, and Zhiyuan Chen,"*Dynamic Query Forms for Database Queries"*, in proceedings of TKDE.2013.62,pages 1041-4347,U.S.A,June 2013.

[3] K. Chen, H. Chen, N. Conway, J. M. Hellerstein, and T. S. Parikh. *Usher: Improving data quality with dynamic forms*, in Proceedings of ICDE conference, pages 321–332, Long Beach,California, USA, March 2010.

[4] Eric. Chu, A. Baid, X. Chai, A. Doan, and J. F. Naughton. *Combining keyword search and forms for ad hoc querying of databases*, in Proceedings of ACM SIGMOD Conference, pages 349–360, Providence, Rhode Island, USA, June 2009.

 