FBR SYSTEM: USER DIRECTED FILTERING OF IMPRECISE QUERIES

Sarika Sarode [1], K. V. Metre[2]

[1] *Department of Computer Engineering, MET's IOE, Maharashtra, India*
[2] *Department of Computer Engineering, MET's IOE, Maharashtra, India*

ABSTRACT

The rapid expansion of World Wide Web has made a large number of databases like the bibliographies, scientific databases etc. So user not able to express their need explicitly and it results in to queries that lead to unsatisfactory results. The FBR (Feature Based Retrieval) system allows user to use imprecise queries to express their uncertainty. The traditional way of searching the data requires specifying the queries clearly. More time is needed to retrieve the data with traditional approach. FBR system computes the sensitivity of the output if user modifies certain conditions. The new conditions to improve the quality of result will also be explored by the FBR system. FBR system is designed in such a way that it can handle the probabilistic queries containing uncertainty. To support interactive response time, FBR system allows user to set threshold value. In large databases, to reduce the searching time there is need to search database scientifically which will lead to faster information retrieval. FBR System provides facility to reprocess the query output which is not provided by the existing System. FBR system also explores the suitable interfaces for users to express their uncertainty, and how to turn this input to probability distribution; FBR system provides facility to user to state his uncertainty in terms of probability value. It is possible to reprocess the query output using FBR system.

Keyword: *Feature Based Retrieval System, Cluster, Probability Distribution, Sensitivity*

1. INTRODUCTION

FBR system is a new way for scientific search in large database.FBR system is designed in way to process the uncertainty and inexplicitly defined queries.FBR system checks the impact of uncertainty over query result.FBR is implemented by using fast algorithms, which provide output to user within real time constraint. So when the user is uncertain about the input data, there is idea that user input should be converted into probability values. This thing motivates the design of FBR System, which will convert the user input into the probability distribution.

To understand the FBR's working, consider the example, suppose there is one large database of Car, which include the information related to various car's with their features. Suppose user want to search the information related any car he has seen. Then the FBR system first analyzes the database of car and then it shows the various features of car to user. Then user gives input to the system. The features of car are like Color, Mac wheel, Type of fuel etc.FBR allows user to express their uncertainty by providing three options i.e. Sure, Not sure, Pretty sure etc. The front end of FBR system converts these user input into the probability values. After processing the user input FBR system generates the report which contains three tables showing the models of car's and attribute score with sensitivity report. FBR's another duty is to help user to check the impact of giving an input about which user is not sure. In

some situations the impact of changing the input condition is high on result. So it is beneficial for user not to change the initial search conditions. To give risk calculation facility FBR provide sensitivity analysis.FBR system provide user a facility to reprocess the query output. FBR system is advantageous when user is uncertain about input conditions. It allows user to define a query with explicit conditions. FBR system would be able to evaluate a query containing probabilistic data. The response time of FBR system would less, it allows user to set threshold value. FBR system is a completely automated approach for the Many-Answers Problem which beverages data and workload statistics and correlations. Exploratory search with imprecise conditions could benefit many other applications, including product search and online medical advice. For product search, suppose a user wants to leverage the wisdom of the crowd for deciding about a camera purchase. Crowd-sourced camera data will contain a mix of objective properties (e.g., megapixels and price) and subjective user evaluations (e.g., if the camera is good for sport photography). In the medical domain, a database of diseases, their symptoms, potential causes (e.g., family history and lifestyle choices), and remedies would similarly be consulted by people not feeling well. As sites like WebMD's symptom checker (http: //symptoms. webmd.com) show, there is great interest in this kind of application. In general, FBR System can be applied to any relational database of interest, helping a database user fine-tune imprecise conditions for exploratory analysis. Now days, Database systems have to play a key role in managing this data because of their abilities in storing, querying, and updating large data collections. Today, it is hard to imagine a single major company or other organization not running a database system for coping with its administrative, financial or corporate data. One of the basic assumptions in database systems is that the data is certain: a data record is either a perfect implementation of some real-world truth or, if it does not hold, absent in the database. In reality, however, a large amount of data is not deterministic, but rather uncertain. Sources of Probabilistic Data a sensor, for instance, comes with a limited precision, and hence its measurements are inherently uncertain with respect to the precise physical value. For example, if we are interested in the birth place of Albert Einstein and we only know that he was born in Germany, then we are unsure about the city of birth. Moreover, ambiguity can cause uncertainty. For instance, most sentences in natural language allow more than one interpretation, sometimes leaving the reader in doubt. Likewise, if we lack information, maybe due to a minimization process reporting an age interval rather than the actual age of a person, we are confronted with several alternatives which in turn generate uncertainty. Similarly, when we observe inconsistent information such as two differing locations of a conference venue, then we can no longer be certain about the true information. Finally, there will always be uncertainty about the future, as can be noticed for instance in every day's weather forecast. So when the user is uncertain about the input data, there is idea that user input should be converted into probability values. This thing motivates the design of FBR System, which converts the user input into the probability distribution.

2. RELATED WORK

Bahar Qarabaqi and Mirek Riedewald, discussed, the new method to calculate the coefficient of correlation. Traditional decision tree classifiers work with data whose values are known and precise. They extend such classifiers to handle data with uncertain information, which originates from measurement/quantization errors, data staleness, multiple repeated measurements, etc. The value uncertainty is represented by multiple values forming a probability distribution function (pdf). They discover that the accuracy of a decision tree classifier can be much improved if the whole pdf, rather than a simple statistic, is taken into account. They extend classical decision tree building algorithms to handle data tuples with uncertain values. Since processing pdfs is computationally more costly, they propose a series of pruning techniques that can greatly improve the efficiency of the construction of decision trees [1]. E. Yilmaz, J. A. Aslam et al discussed the minkowskies and average precision methods of calculation of the distance between two rankings in detail. Takes approach toward a new rank correlation coefficient, AP correlation (ape) that is based on average precision and has a probabilistic interpretation [2].

D. Suciu, D. Olteanu, C. Re, and C. Koch discussed the Imprecise data design and methods, Highlighted a number of ongoing research challenges related to PDBs, and kept referring to an information extraction (IE) scenario as a

running application to manage uncertain and temporal facts obtained from IE techniques directly inside a PDB setting[3].Y. Diao, et al., proposed, An Automatic Interactive Data Exploration framework, that iteratively steers the user towards interesting data areas and predicts a query that retrieves his objects of interest. AIDE leverages relevance feedback on database samples to model user interests and strategically collects more samples to refine the model while minimizing the user effort. [4].

K. Dimitriadou, O. Papaemmanoui, and Y. Diao presented the The need for effective IDE l only increase as data are being collected at an unprecedented rate how an automated navigation assistant service, DBNav, for interactive exploration of large datasets. DBNav would help a user quickly navigate through a complex large data spaces. Profiles of both user interest and application characteristics are the key ingredients to realizing this vision. Profiles can be either supplied by the user or learned from interaction logs[5].

B. Qarabaqi and M. Riedewald discussed about the a great deal of interest in the past few years on ranking of results of queries on structured databases, including work on probabilistic information retrieval, rank aggregation, and algorithms for merging of ordered lists. In many applications, for example sales of homes, used cars or electronic goods, data items have a very large number of attributes. When displaying a (ranked) list of items to users, only a few attributes can be shown. [6].

F. Olken and D. Rotem, Presented a novel relevance feedback scheme based on classifier combination and a method to automatically tune weights in a distance function for content-based image retrieval which can be incorporated into most distance-based image retrieval system. The weighting scheme is integrate schemes from the literature and leads to clear improvements in all. All in all the classifier combination scheme in combination with the weight learning out performs all other methods which was evaluated on two different databases, one purely content-based task, and another one incorporating textual and visual information[7].

Gautam Das, Vagelis Hristidis,discussed the problem of selecting the top m attributes from the view point of helping a user understand what factors most influenced a ranking system in its ranking decisions. They presented several variants of the problem, showed that several of these variants are NP-hard, and presented efficient greedy heuristics. They performed a user study demonstrating the benefits of a hybrid approach that returns the top attributes from each of these variants[8].

3. SYSTEM ARCHITECTURE

Fig-1 illustrates the architecture of FBR system. FBR system response to the feature based user input specified in terms of probability by calculating its rank according to entities and attributes. It will first trains the models of entity ranker and attribute ranker using training dataset. Then it becomes able to calculate the respective rankings. Sensitivity analysis can be done only if user demands. It suggests new conditions that will useful to improve the quality of result. It computes the sensitivity of the result. If user modifies the input conditions then FBR system provides the sensitivity of output, so user can avoid the modification in to the input if it has high impact on result. In the following architecture the user can give an input in terms of attributes. User can select attributes. Then the FBR System processes the attributes and turns the user input into query. Then the query will processed by the FBR System. And the Report will generated by the FBR system. The Attributes and Entities will be ranked by the Attribute and Entity Models respectively.

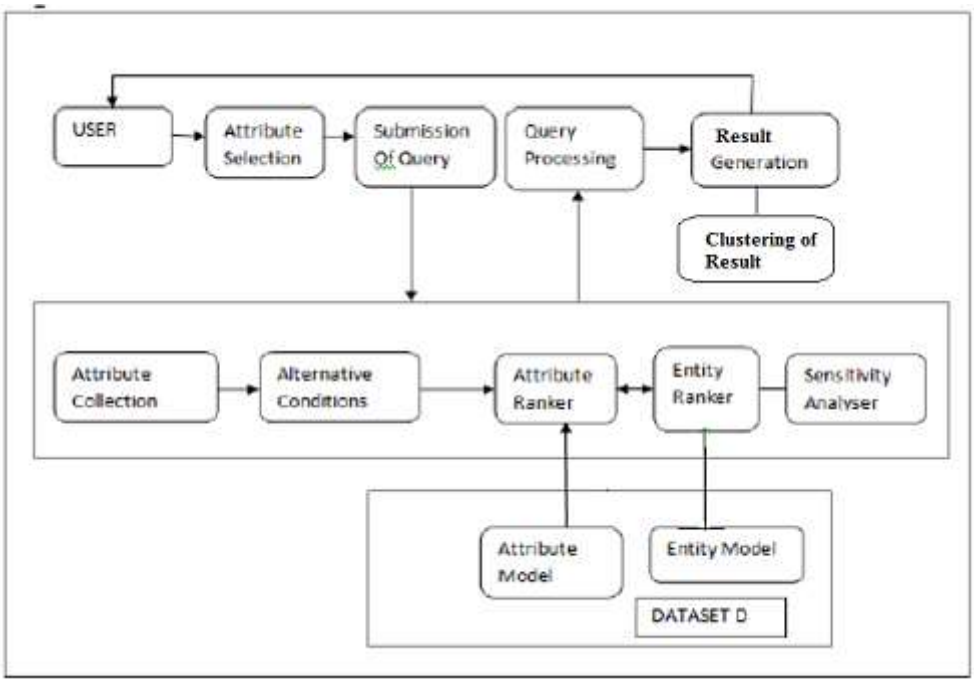

Fig -1: System Architecture of FBR System.

1. Entity ranking

Entity Ranker uses Entity Model to rank the entities at the query time. Entity Model will train by training dataset which can be precise or probabilistic.

2. Attribute ranking

Attribute Ranker uses Attribute Model to rank the attributes at the query time. Attribute Model will trained by training dataset which can be precise or probabilistic.

3.Total computation time

Total computation time is the summation of time required to rank the entities and time required to rank the attributes.

4. Sensitivity Analysis

Sensitivity will be analyzed only on when user demands. All these are the background processes.

FBR system related to the software which can able to retrieve the data by processing uncertain conditions. So it provides fast retrieval within less time from the large database. So this software can be used with any kind of database. The FBR system is designed to do following tasks

1. To find the precise result the user is looking for.
2. To combine the two approaches so that the user can start with imprecise conditions.
3. To rank the results for database queries.
4. To gain wider acceptance for big data analysis.

3.1 Algorithms

Algorithm for finding Probability Score.

I**nput: A=Number** of fields and their most frequently data item counts in the data set.

Output: P=Probability score of every field in the dataset.

if T (A) == 1 then /* Single Attribute */

{

Probability (A) = Number of Outcome Favourable /Total numbers of outcomes

}

else if T (A) > 1 then /* Multiple Attribute */

P (A1 and A2) =P (A1) P (A2 after A1)*

Return P(A);

}

Algorithm for Ranking the result using Average Precision.

Input: Q: All extracted result set tuple and their Probability score.
Output: Li: Ascending or descending the result set depends on their Pr score.

1. Invoke the interface S (Q, A) with output stream L and store top-n tuple of L in list L0

2. A = n; /*A is the set of attributes selected so far*/

3. for each j in 1. . . m do

{/*repeat until we find top-m attributes*/ }

4. for each ai in A − A do

{/*Invoke M − |A| pipelining interfaces*/ }

5. Invoke pipelining interfaces S (Q, A [{ai}) with output streams Li

6. P (Li) = 0

/*P (Li) is the prefix of Li retrieved so far*/

7.Return output stream Li

}

Algorithm for creating clusters of Result.

/*Clustering or cluster analysis involves assigning data points to clusters such that items in the same cluster are as similar as possible, while items belonging to different clusters are as dissimilar as possible. Clusters are identified via similarity measures. These similarity measures include distance, connectivity, and intensity. Different similarity measures may be chosen based on the data or the application.*/

Input: Dataset d, Type_Of_Attributes.

Outputs'' Clusters created

1.Choose the attribute type.

2. Choose a number of clusters (manual/dynamic)

3. Assign randomly to each point coefficients for being in the clusters.

4. Repeat until the algorithm has converged (that is, the coefficients' change between two iterations is no more than, the given sensitivity threshold)

5. Compute the centroid for each cluster.

6. For each point, compute its coefficients of being in the clusters.

7.Return the cluster

3.2 Mathematical Model

Let S be the Feature Based Retrieval system, which can be represented in terms of input, functions and output.

$$S = \{I, F, O\}$$

I: Input: [I1, I2, I3……In]

I1= Attribute value 1

I2= Attribute value 2

I3=Attribute value 3

In= Attribute value n

F: Functions: [F1, F2, F3, F4, F5, F6]

F1=Query generation from user input

F2=Find Probabilistic Data on Uncertain Data.

F3=Stretching Shrink technique finds relaxations and contractions based on user feedback.

F4=Risk estimation, perform sensitivity quantifies the risk of a condition.

F5=Perform classification on Query Time.

F6= Result Clustering.

O: Output: [O1, O2, O3, O4, O5]

O1=Query Generation from user input (Attribute collection)

O2=Sensitivity value of specified attribute.

O3=cost

O4=Interactive response time to query response.

O5=Result in cluster form.

4. EXPERIMENTAL SETUP

A desktop application for implementing scientific search is developed. The system is implemented on Windows 7 platform. JAVA environment with SQL Server for the backend. Core i3 machine with 2 GB RAM is used for

development and testing. Dataset: Car dataset is used. Car dataset contains specifications of car's details, images, and their reviews.

5. RESULT AND ANALYSIS

This section examines the effectiveness of the proposed FBR system for building the probabilistic queries over attributes. FBR system calculates the probability score and sensitivity score of the different attributes. It calculates the time required to search the particular record stored in the database. The probability score is calculated when the user is uncertain about his input. The probability score indicates the possible probable matches in percent. The following graph shows the probability score of all the attributes like as Brand, Price, Name, Model etc...Input is given to the FBR system in terms of attribute value. With the values color =red with precision=1 and price=320000 and 1170000, the probability score calculated by the system is shown in chart 1

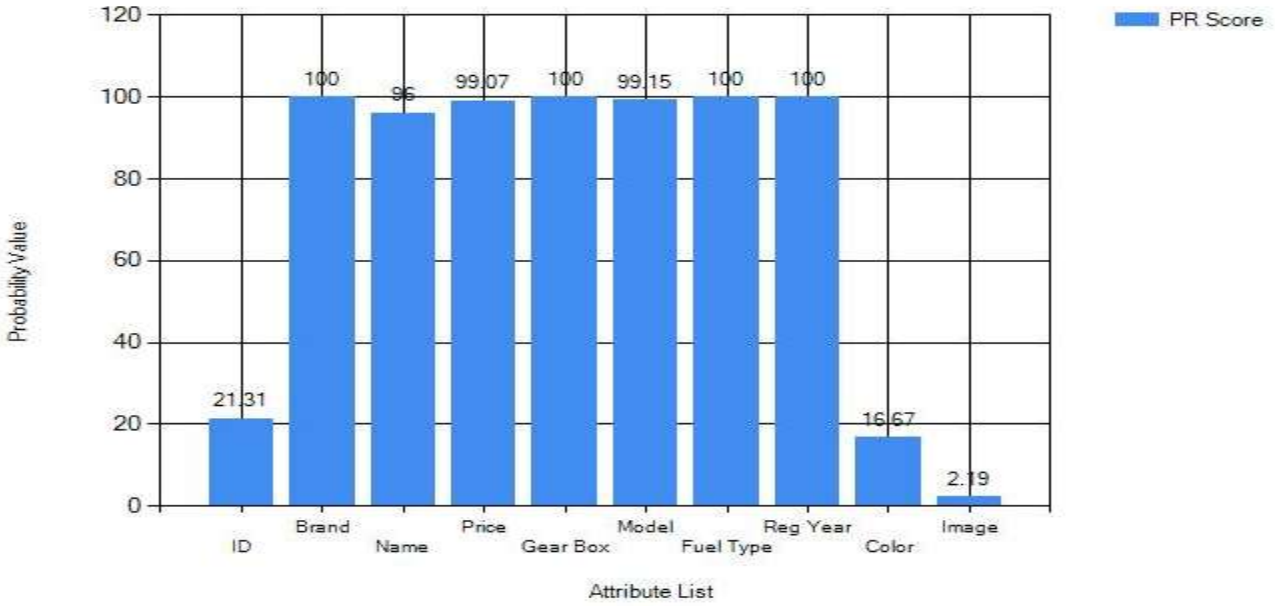

Chart -1: Graph for probability score of attributes.

Chart 2 shows the graph which presenting the time required to search the relevant records.
According to user input, the time is varying for number of attributes. Time is given in milliseconds. As the no of attributes increases the time required to search the relevant records is also increases. The graph shows the time required for the input values color =red with precision=1 and price=320000 and 1170000.

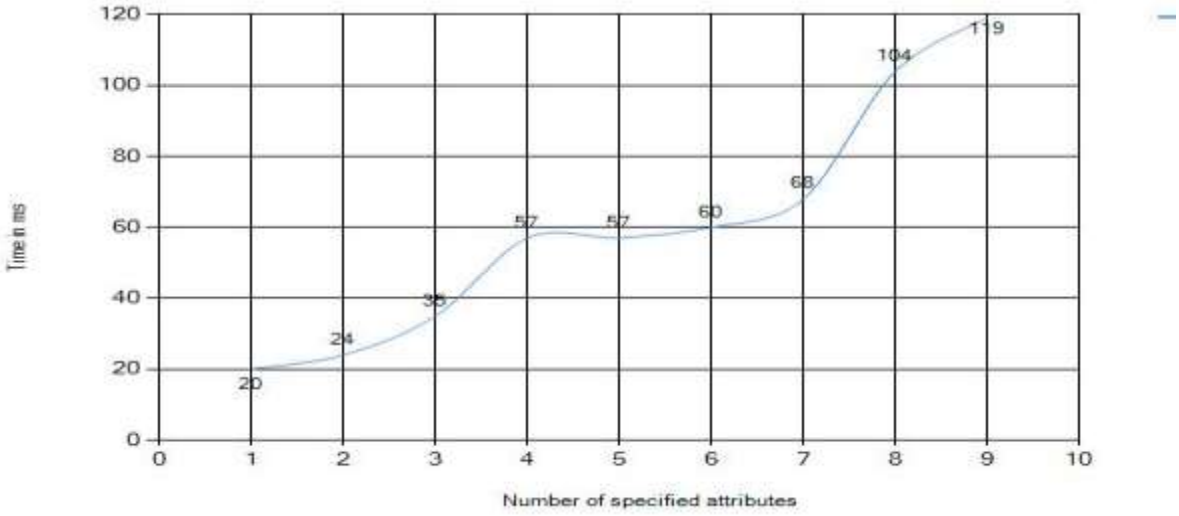

Chart -3: Graph showing the time required to searching.

CrowdOp is the similar system which is used to retrieve the data from the database on the basis of query plans. User gets the best optimized plan based on the attributes. Here multiple operators are optimized which is very complex. Declarative crowdsourcing query can be evaluated in many ways, the choice of execution plan has a significant impact on overall performance, which includes the number of questions being asked, the types/difficulties of the questions and the monetary cost incurred. Where FBR system process the user input which contains the uncertainity, which is not provided by CrowdOp.

6. CONCLUSION

The proposed system is a completely automated approach for handling the uncertainty of user about query input. While doing big data analysis the database has to support many users finding information they are looking for. The FBR System is proposed to allow user to explicitly express their uncertainty through probabilities. FBR is a new way to scientifically search the large database.FBR system reduces the time required to retrieve the data. It is a smart way to search the data which reduces the user effort.

7. REFERENCES

[1].Bahar Qarabaqi and Mirek Riedewald, Member, IEEE, \"Merlin: Exploratory Analysis with Imprecise Queries ", In IEEE Symposium on Knowedge And Data Engineering, VOL. 28, NO. 2, , 30 Oct. 2015

[2] E. Yilmaz, J. A. Aslam, and S. Robertson, "A new rank correlation coeffcient for information retrieval", in Proc. 31st Annu. ACM SIGIR Conf.

[3]. D. Suciu, D. Olteanu, C. Re, and C. Koch, "Probabilistic Databases". San Rafael, CA, USA: Morgan Claypool.

[4] Y. Diao, et al., "AIDE: An automatic user navigation service for interactive data exploration (demo)," Proc. VLDB Endowment, vol. 8, no. 12, pp. 1964–1967 2015.

[5]. K. Dimitriadou, O. Papaemmanoui, and Y. Diao, "Explore-byexample: An automatic query steering framework for interactive data exploration," in Proc. ACM SIGMOD Int. Conf. Manage. Data,2014, pp. 517–528.

[6] B. Qarabaqi and M. Riedewald, "User-driven refinement of imprecise queries", in Proc. IEEE 30th Int. Conf. Data Eng.,2014, pp. 916927.

[7]. F. Olken and D. Rotem, "Random sampling from databases – a survey, in Statistics Comput., vol. 5, no. 1, pp. 25–42, 1994.

[8]. Gautam Das , Vagelis Hristidis, "Ordering the Attributes of Query Results" *SIGMOD 2006,* June 27–29, 2006, Chicago, Illinois, USA,ACM 1595932569/06/0006