Data Analysis in SJ Clustering and Mary-Data Query System Analysis using Jasper Server Report 5.5 Tool

S.J. MaryJasper\textsuperscript{1,2}, V. SugaPriya\textsuperscript{2}, R. KrishnaMoorthy\textsuperscript{3}

\textsuperscript{1,2}Assistant Professor, Department of Management Studies, Alpha College of Engineering, Chennai, Tamilnadu, India.

\textsuperscript{3}P.G Student, Department of Management Studies, Alpha College of Engineering, Chennai, Tamilnadu, India.

ABSTRACT

This paper identifying records that produces compatible results using Fast Clustering Selection Algorithm. A selection algorithm may be evaluated from both the efficiency and effectiveness points of view. While the efficiency concerns the time required to find a record, the effectiveness is related to the quality of the record. The selection algorithm fetches the result with the help of register number. The Selection algorithm works in two steps. In the first step, the register number fetches the result from the server. The record for every individual will be obtained by hit method. The sender sends the request to the server. In the second step, the most representative record that is strongly related to target classes is fetched from database. The record fetches from the database by the register number. The string generation algorithm is guaranteed to generate the optimal result \( k \) candidates. We analyses the results of students using Selection Algorithm. We need to define compatible operation analogs by introducing max-min operation & min-max operation. It automatically collects data from the web to enrich the result. The analysis of result for huge students make more time. The accuracy of the result has to be considered. We need to fetch the result individually by their register number. It leads to time inefficiency. In a proposed system, we obtain the result for a group of students. The Selection method fetches the result for a student according to their register number which is entered in between a range. The result for the student automatically fetched from the server. Once the result for the candidate has been fetched from the server, it stored in the client database. Then we sort the result of the student as group. It increases the accuracy and makes the efficient one. It reduces the burden of the people who analyze the result. The result analysis is performed within a short period. We can generate the report based on the GRADE system. Our experimental evaluation shows that our approach generates superior results. Extensive experiments on large real data sets demonstrate the efficiency and effectiveness. Finally we sort the results of students using JASPER FAST CLUSTERING SELECTION algorithm.

Keywords : Jasper, Minmax and Maxmin Operation.

I. INTRODUCTION

Students play a major role in Educational field. Students are evaluated under different categories: By choosing their institution, studying well, gaining good knowledge, and getting good marks. Result analysis of each student paves the way for their higher education as well as their improvement in future. Percentage marks prior to the grade scheme were converted into grades for ease of comparison. The reliability of the new scheme was again studied using statistical analysis of data obtained from both
the old and new schemes. Some assessment schemes use a grading category index (GCI) instead of actual mark for each assessment criterion. GCIs usually have a smaller number of options to choose from when awarding results.

For example, the GCI may gave eight levels with the highest being awarded to exceptional students and the lowest being awarded to students of inadequate performance.

This reduced level of categories has been shown to result in less variability between assessors compare to systems which use marking ranges between 0 and 100. The Results of the students are analyzed using Fast Clustering Selection Algorithm.

In this paper, we are analyzing the results of students using clustering methods with the help of filtering by introducing max-min operation &min-max operation. The filter method is usually a good choice when the number of records is very large. The SELECTION algorithm works in two steps.

In the first step, the register number fetches the result from the server. The record for every individual will be obtained by hit method. The sender sends the request to the server. In the second step, the most representative record that is strongly related to target classes is fetched from database. It consists of three components: query generation, and data selection and presentation. This approach automatically determines information. It then automatically collects data from the web. By processing a large set of data; it is able to deal with more complex queries. In order to collect result, we need to generate informative queries. The queries have to be generated for every individual student. It increases the time to fetches the result and inefficiency. In order to overcome this, the queries are generated along with unique identification number i.e. register number. Based on the generated queries, we vertically collect image data with multimedia search engines. We then perform re-ranking and duplicate removal to obtain a set of accurate and representative results.

II. RELATED WORK

Selection can be viewed as the process of identifying and removing as many irrelevant and redundant record as possible. This is because: (i) irrelevant records do not contribute to the predictive accuracy, and (ii) redundant features do not redound to getting a better predictor for that they provide mostly information which is already present. Selection focused on searching for relevant records.

Irrelevant data, along with redundant data, severely affect the accuracy. Thus, selection should be able to identify and remove as much of the irrelevant and redundant information as possible.

QUERY GENERATION

To collect result from the web, we need to generate appropriate queries before performing search. We accomplish the task with two steps. The first step is query extraction. We need to extract a set of informative keywords from querying. The second step is query selection. This is because we can generate different queries: one from retrieve, one from display, and one from the combination of retrieve and display.

In query generation, given an input string Qi, we aim to generate the most likely k output strings so that can be transformed from Qi and have the largest probabilities.

DATA SELECTION AND PRESENTATION

We perform search using the generated queries to collect the result of the student. The result of the
student is fetched from the server by three processes. Before query generation, the register number for the students is fetched from the database. The register numbers are grouped based upon the department. The register number for each group is partitioned and stored as arrays of objects. In query generation, the register number is added with the query and it performs the request to server. The results are built upon text based indexing. Therefore, re-ranking is essential to reorder the initial text-based search results. A query-adaptive re-ranking approach is used for the selection of the result. We first decide whether a query is text related or image related, and then we use different features for re-ranking.

Here we regard the prediction of whether a query is text related as a classification task. We can choose to match each query term with a result list. But it will not be easy to find a complete list. In addition, it will be difficult to keep the list updated in time.

We adopt a method that analyzes results. Thus, we perform a duplicate removal step to avoid information redundancy. The result which is fetched from the server may increases the time if there is large amount of data. To increases the time efficiency we need to process the query in a different manner. The results are grouped with the help of group id.

EVALUATION OF QUERY GENERATION

The generated query is first passed as a string to the server. The server searches the result with the register number. Once the result is found for the particular register number, the server sends the respond to the query client. The result received for a particular student is stored in the database with help of the register number. The results can be printed for a group of students by simply selecting the results from database with the group id. The group id is set for a group of students based upon their department id. The department id is a unique constraint for the identification of the record. In query generation the records are fetched from the server and stored in the client database by the department id and group id.

EVALUATION OF RERANKING

We use the query adaptive ranking to perform query classification and then adopt query-adaptive re-ranking accordingly. It is our proposed approach and it is denoted as “proposed”. After re-ranking, we perform duplicate removal and irrelevant removal of result.

III. ALGORITHM AND ANALYSIS

The proposed FAST algorithm logically consists of two steps:
(i) removing irrelevant record, (ii) removing redundant record.
1) Irrelevant records have no/weak correlation with target concept;
2) Redundant records are assembled in a cluster and a representative data can be taken out of the cluster.

ALGORITHM

For every result
Calculate the average queue size(avg)
\text{if} \ minth<\text{avg}<\text{maxth} \\
\{ \\
\text{Calculate probability } p_a \\
\text{With probability } p_a: \\
\quad \text{if } \text{register no. is valid and} \\
\quad \text{if the result is not already fetched} \\
\quad \{ \\
\quad \quad \text{Mark the result} \\
\quad \quad \text{Send request to the sender and save the result} \\
\quad \} \\
\text{else} \\
\quad \text{Drop the request to the server} \\
\} \\
\text{else if } \text{maxth} \leq \text{avg} \\
\text{Store the result in database} \\
\text{Send acknowledgment to the server.} \\

\text{Fig. 1. gives the flowchart of the algorithm FAST Algorithm} \\

\text{Diagram 1: Jasper Server - Architecture} \\

\text{The FAST algorithm fetches the result of the student with the help of the register number.} \\

\text{The algorithm checks whether the given register number is valid or invalid. The register number is a collection of college code and student code.} \\

\text{The college code is used to identify the result of the particular college.} \\

\text{The FAST algorithm calculates the probability of finding the result of the student from the server. Then it identifies the results from the server using the request and response method. The } \text{avg}<\text{max condition is checked first to find whether the result of the student is pass out student or current student. If it becomes true then it checks the server whether the result of the student is released by the university. If the university does not release the result means then the request process is stopped and the response message is passed to the user that the university still}
now did not release the result for the current semester of the particular register number. If the university release the result means then the request for the result is send to the server. The server identifies the result belonging to the particular student with the help of the student register number.

Once the result is identified by the server then it responds to the client by sending the reply. Once the client receive the response its store the result in the client database. This increases the time efficiency, accuracy and cost benefit.

**SELECTIVITY OF RANGE QUERIES**

Selectivity estimation of range queries is a much harder problem. Several methods were available. However, they are only able to estimate the number of records in the range. None can be efficiently adapted to estimate the number of results in the range. One naive solution is to treat information as record by removing the irrelevant information. This clearly increases the space consumption significantly (and affects the efficiency) since the number of points is typically much larger than the number of existing nodes. When generating the query workload for our datasets we had to address two main challenges. We had to generate a work load, with an attribute distribution representing the user interests in a realistic way. Second, we had to create queries of the form attribute-value.

Query reformulation involves rewriting the original query with its similar queries and enhancing the effectiveness of search. Most existing methods manage to mine transformation rules from pairs of queries in the search logs. One represents an original query and the other represents a similar query.

1) Select the length of the query l by sampling from a uniform probability distribution with lengths varying from 1 to 3.

2) Select an attribute A1 using the popularity that they have on the vector

3) Select the next attribute A2 using the co-occurrence ratio with the previous attribute A1.

4) Repeat from Step 2, until we get l different attributes.

**DATABASE SIZE EFFECT**

We check the effect of the size of the database on the precision of attribute suggestions and the number of query matches. We consider subsets of the database of documents of different sizes. As expected the proposed strategies increase their quality when we increase the data size. The size of the result is based on the method of us storing it. We storing the data which is retrieved from sever to the client database which increases the time efficiency and minimum storage capacity.

![Diagram 2: Jasper Server – Graphical Suite](image1)

![Diagram 3: Jasper Server – Login Page](image2)

The results are stored in the database by the student register number which requires less storage and increases the efficiency of accessing the information.
IV. CONCLUSION

In this paper, we have presented a clustering-based selection algorithm for result analysis. The algorithm involves (i) removing irrelevant records, (ii) removing redundant records. We can do the result analysis but it makes more time to get the result of every student. For that we are using a selection algorithm which removes the redundancy of the result and using it we can fetch the result of large group of people. We have adopted a method to remove duplicates, but in many cases more diverse results may be better. In our future work, we will further improve the scheme, such as developing better query generation method and investigating the relevant segments from the result.

V. REFERENCES