Exploring the record's Standard In a Relation for better understanding

Shilpa S.Devasthali

[1] M.E.IIyr Student,Computer Department,JSPM's BSIOTR,Wagholi,Pune,Maharashtra,India

ABSTRACT

For the given set of ranked records, we are facing the problems that which condition is to be selected for the raising the standered of the record. we have to find the qualified record that shows its ranking as high as possible among the qualified tuples.For that we study the standing maximization problem , this will give the approximate solution for the given problem. . It will show the object promotion and characterization. we also show the hardness of problem and for that solution proposed the greedy methods for high accuracy. Our solution on real database will confirm the effectiveness and efficiency.

Keyword : - *SMP(Standing Maximization Problem),Relational Database,NP-Hardness.*

1. Introduction

To rank the record as per user preferences there are certain types of operations are used. These operations include top-k and skyline queries. The top-k operation and the skyline query are used to calculate the highest grade. To determine the top k objects, that is, k objects with the highest overall grades, the naive algorithm must access every object in the database, to find its grade under each attribute. The skyline query and operator were introduced to the database context by applying the problem of finding the maxima of a set of points.. Recently, due to application of skyline in multi-criteria decision-making and mobile service applications, it has gained popularity and now widely used in database literature. With the help of these queries superior object can be defined.

Table 1:A relation with CS PhD students.

Name	Age	Location	Expertise	Publications
Brown	30	N.America	System	14
Smith	27	N.America	Database	8
Suzuki	32	Asia	Theory	9
Muller	28	Europe	Theory	15
Dubolis	26	Europe	System	12
Martin	31	Europe	Database	17
Kim	28	Asia	Database	10
Chen	26	Asia	Theory	12
Gupta	26	Asia	System	13

Consider the example Table 1 as shown- a relational data for the CS PhD graduates. A relation with CS PhD Graduates. This table contains the attributes as name, age, location, expertise and publication. For measuring the quality of graduates, consider publications as the measuring attribute. If we go accordingly then kin does not have a good ranking. But if we restrict the relation with (age<30) and (expertise ='databases'), then kin's ranking is 1st.

2. LITERATURE SURVEY:

2.1) Ordering the attributes of query results

Authors: G. Das, V. Hristidis, N. Kapoor, and S. Sudarshan.

Automatic selection of attributes is required to deal with different requirements of different users. The optimization problem of choosing the most “useful” set of attributes, that is, the attributes that are most influential in the ranking of the items. Here, in this approach, it returns the top attribute from each variant.

2.2) Query by output

Authors: Q. T. Tran, C.-Y. Chan and S. Parthasarathy

The novel data-driven approach is presented here. In this approach the hidden relationships in the given database is considered. And according to that it generates instance-equivalent queries. It introduces set of criteria to rank order output queries. Query by output introduces investigation of a problem of finding SQL statement that produces the result which include given set of input tuple.

2.3) Region-based online promotion analysis

Authors: T. Wu, Y. Sun, C. Li, and J. Han

This approach identifies the top-k interesting region for most effective promotion of object given by the user, where a region is defined over continuous ranged dimensions. for this it requires large space and aggregation operations. Materialisation algorithm is used in this approach but this is expensive method.

2.4) Incremental discovery of prominent situational facts

Authors: A. Sultana, N. Hassan, C. Li, J. Yang, and C. Yu

In the Incremental discovery of prominent situational facts, there is ever-growing append-only table. The entry in that table is done when it follows the specified criteria. and hence the tuple in the table becomes the skyline object. In this method it maintains the set of sky line tuple and compares each new tuple with the skyline tuple.

2.5) Interactive query refinement

Authors: C. Mishra and N. Koudas

In this approach the investigation is done on how to refine the predicates of query as a result of which satisfies the user specified cardinality constrain. This attempt to query output by size.

2.6) Identifying the most influential data objects with reverse top-k queries

Authors: A. Vlachou, C. Doulkeridis, K. Nørva_ g, and Y. Kotidis

This approach deals with Reversing top-k queries leads to a query type that instead returns the set of customers that find a product appealing. It address the challenging problem of processing queries that identify the top-m most influential products to customers, where influence is defined as the cardinality of the reverse top-k result set

3. DISADVANTAGES OF EXISTING SYSTEM

3.1) The systems are designed to study the problem of small set of attributes.
Considering the large amount of data these systems takes more time to solve the problem

3.2) It works with single attribute selection only.

4. SYSTEM ARCHITECTURE

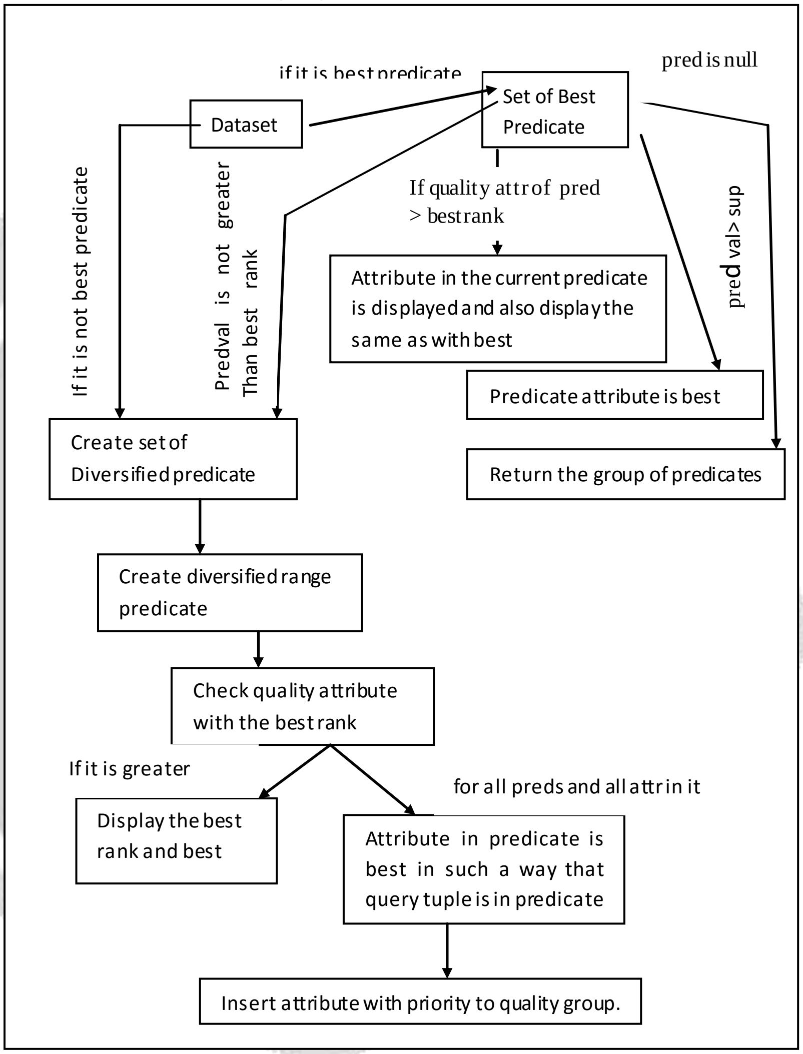
For a given set of record in a relation we have to find the record that stand out among the qualified tuples.

For this system we take the relational dataset as an input to our system. Next we have to check whether the given query tuple is *Best predicate* or not. This means that, if it is the *Best predicate* then we have to create the set of all *Best predicate* and if it is not the *Best predicate* then we have to create the set of diversified predicates.

After this we have to check each predicate value with the best rank value or with the support value. This is because we have to find the most qualified tuple for the predicate set. When we compare this predicate value with the *Bestrank* or with the *support* value there are four types of return we get. First, if pred_value is null. At this time the system will return the group of predicate. Second if the pred_value is greater than *support_value* is found, then we can say that the predicate attribute is the best and the system will return the same. Third, if the quality attribute of the predicate is found greater than the *Bestrank*, that time the system will display the attribute in the current predicate and display the same as best. Fourth, if the predicate value is not greater than the *Bestrank*, this time the system creates the set of diversified predicate. Then the range is specified. And according to that range, the diversified range predicate is created. For this range predicate set, each quality attribute is compared with the *Bestrank*. If it found greater, the system will display the quality attribute as the best.

If it is found it is not greater, the comparison is done for all predicates and for all attributes in it. This time the system will insert the quality attribute into the set of diversified predicate after checking two conditions: first is, there must exist the query tuple in the predicate set and second is, value of attribute is greater than support value.

Fig: System Architecture


if it is best predicate
pred is null
Set of Best Predicate
Dataset
If it is not best predicate
Predval is not greater Than best rank
If quality attr of pred > bestrank
pred val> sup
Attribute in the current predicate is displayed and also display the same as with best
Predicate attribute is best
Create set of Diversified predicate
Return the group of predicates
Create diversified range predicate
Check quality attribute with the best rank
If it is greater
for all preds and all attr in it
Display the best rank and best
Attribute in predicate is best in such a way that query tuple is in predicate
Insert attribute with priority to quality group.

In the proposed system the main focus on the time requirement to solve the problem as well as the proposed system presents greedy method which explores the search space partially. Also it identifies the sub-optimal SMP solution with high quality.

4.1 ADVANTAGES OF PROPOSED SYSTEM

- This system is fast and it explores very limited part of search spaces
- It considers small number of predicates.
- This system finds the solution close to optimal**.**

5. METHODOLOGY

To solve the SMP problem, there is straight forward approach that is search with depth first manner, search out all possible subspaces. And report the subspaces where the query object has highest percentile rank. This basic method is known as Naive algorithm. But this approach is having high cost. Hence there are number of greedy approaches that can be used. First method is Browsing Algorithm.

Browsing Algorithm:

This algorithm extracts the classification rules from set of records. it iteratively selects the sub range which

(i) includes t_q,

(ii) includes at least sup.|R| records when applied together with the predicate Selected so far, where sup is the minimum support constraint, and

(iii) Maximizes the ratio of positive to all tuple covered by the rule (i.e., range).

Browsing algorithm compares the records with the t_q whether it is less than or equal to or it is greater than equal to. With the help of this it may decide that whether it is positive dimension or negative dimension. It works definitely; but the working of BA is slow.

Hence there is another solution that works relatively faster than the BA. This algorithm is known as Diversified – Path Browsing Algorithm (DBA).

Algorithm 1. Naive Algorithm

```
G := R; Preds = ϕ; bestrank := qual(G);
bestG := G; bestPreds :ϕ;
procedure NAIVERANGE(G, Preds)
    if all attributes are in Preds then
        if qual(G) > bestrank then
            bestrank :=qual(G);
            bestG := G; bestPreds :=Preds;
    else
        Pick any attribute A not in Preds
        A:preds := all possible predicates on A for G
            such that |σpredG| ≥sup .|R|;
    for each pred ϵA.preds do
        G' = σpredG;
        Preds' :Preds ∪ {pred};
        NAIVERANGE(G', Preds');
return {bestG, bestPreds};
```

Diversified Algorithm (DBA):

It is faster than Naive algorithm. This algorithm works on diversified predicate for single attribute.

Algorithm 2:DBA Algorithm

```
$G_{best}$ := r; Predsbest := $\phi$; bestrank := qual(G);
procedure DIVERSERANGE(G,  Preds)
        pred := none;
        for each attribute A not in Preds do
            A.pred := best predicate on A for G
              such that $t_q$.A $\epsilon$ A.pred and $|\sigma_{A:pred})| \geq sup.|R|$;
                  if qual($\sigma_{A:pred}$G) > bestrank then
                 pred := A.pred;

       If pred =none then
                 if qual($\sigma_{Preds}$G) > bestrank then
                          bestrank := qual($\sigma_{Preds}$G);
                           $G_{best}$ :=G;
                           $Preds_{best}$ := Preds;
                 else
                           A := the attribute of pred;
                            $DP_A$:=Diversified  predicates on A  of G;
                            for each predicate pred' $\epsilon$ $DP_A$ do
                                if qual$^+$($\sigma_{pred}$'G) $\leq$ bestrank then
                           continue;
                 Preds':= Preds U{pred'}
                 G':=$\sigma_{pred'}$G;
                  DIVERSERANGE(G',  Preds';)
return {$G_{best}$, $Preds_{best}$}
```

Enumerating Diversified –Path Browsing (EDBA)

In BA (Browsing Algorithm it works very iteratively. At each Iteration it picks the most useful attribute. While in this EDB Algorithm, it examines all permutations of the predicate attribute. This algorithm is based on the current percentile of the record. It increases the finding a better percentile rank. By examining all possible permutations of predicate attribute. EDBA takes the prioritized attributes for the work. It arranges according to the improvement made at the each record. Hence the time required is less as compared to the previous method. EDBA gives the optimum solution

Algorithm 3. EDBA Algorithm

```
G :=r; Preds = $\phi$; bestrank := qual(G);
bestG :=G; bestPreds ;= none;
procedure ENUMATTRIBUTE(G, Preds)
    if all attributes are in Preds then
        if qual(G) > bestrank then
            bestrank := qual(G);
            bestG := G; bestPreds := Preds;
        else
            Q := {}                    //priority queue
        for each attribute A not in Preds do
            A:pred := best predicate on A for G such that
            $t_q$:A ϵ A:pred and $|\sigma_{A:pred}G)|\geq sup.|R|$;
        Insert A into Q with priority qual($\sigma_{A:pred}G$);
        while Q is not empty do
        A := next attribute in Q;
        $G_{best}$ :=G;
        $Preds_{best}$ := Preds;
    else
        A := the attribute of pred;
        $DP_A$ := Diversified predicates on A of G;
        for each predicate pred’ ϵ $DP_A$ do
            ENUMATTRIBUTE(G’, Preds’);
return {bestG, bestPreds};.
```

6. COMPARITIVE RESULT

Query Object	Object description and original rank	predicate set and promoted rank			
		NAÏVE	BA	DBA	MA
Yao Ming(2006)	Height=7’6” Weight=310 Born=1980 375/24,524	Weight= [250,315] Born= [1964.1987] 1/1345	Born= [1979,1981] /24,524	Weight= [250,315] Born= [1964,1987] 1/1345	Weight= [250,315] Born= [1964,1987] 1/1345
Nate”Tiny” Archibald(1971)	Height=6’1” Weight=150 Born=1948 17/24,524	Height= (-∞,6’4”] 1/7683	Height= (-∞,6’4”] 1/7683	Height= (-∞,6’4”] 1/7683	Height= (-∞,6’4”] 1/7683
Cheis Bosh(2009)	Height=6’10” Weight=228 Born=1984 177/24,524	Height= [6’9”] Weight= [225,245] 1/1,822	Height= (6’10”.6’11”] Weight= [225,245] 1/1203	Height= (6’9”.6’11”] Weight= [225,235] 1/1326	Height= (6’9”.6’11”] Weight= [225,245] 1/1822
Kevin Durant(2012)	Height=6’9” Weight=215 Born=1988 141/24,524	Height= [6’9”,6’12”] Born= [1964,+∞) 2/4212	Born= [1985,+∞) 2/1559	Born= [1985,+∞) 2/1559	Height= [6’9”,6’12”] Born= [1964,+∞) 2/4212

7. CONCLUSION

We studied the SMP(Standing Maximization problem).This is the problem finding the set of selection predicate on a relation that maximizes the rank of given tuple in the selection result, according to measure attribute. Here we propose the greedy methods namely, Multiatribute solution that finds the approximate solution with less time. This will improve the usability and efficiency of these methods.

8. ACKNOWLEDGEMENT

I take this special opportunity to express my sincere gratitude towards professor and all the people who supported me during my entire project work. I would like to express my gratitude to my guide and also the project coordinator Prof. Prashant Jawalkar for providing special guidance. I would also like to thank our HOD Prof. G.M.Bhandari who always has enough time to solve student's problems at any time. Finally thanks to all teachers who are always supportive at us.

9. REFERENCES

[1] W. W. Cohen, "Fast effective rule induction," in Proc. Int. Conf.Mach. Learn., 1995, pp. 115–123.

[2] S. B€orzs€onyi, D. Kossmann, and K. Stocker, "The skyline operator,"in Proc. Int. Conf. Data Eng., 2001, pp. 421–430.

[3] R. Fagin, A. Lotem, and M. Naor, "Optimal aggregation algorithms for middleware," in Proc. ACM Symp. Principles Database Syst., 2001, pp. 102–113.

[4] G. Das, V. Hristidis, N. Kapoor, and S. Sudarshan, "Ordering the attributes of query results," in Proc. ACM SIGMOD Int. Conf. Manage.Data, 2006, pp. 395–406.

[5] M. Miah, G. Das, V. Hristidis, and H. Mannila, "Standing out in a crowd: Selecting attributes for maximum visibility," in Proc. Int.Conf. Data Eng., 2008, pp. 356–365.

[6] C. Mishra and N. Koudas, "Interactive query refinement," in Proc.12th Int. Conf. Extending Database Technol.: Adv. Database Technol.,2009, pp. 862–873

[7] Q. T. Tran, C.-Y. Chan, and S. Parthasarathy, "Query by output,"in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2009, pp. 535–548.

[8] T. Wu, D. Xin, Q. Mei, and J. Han, "Promotion analysis in multidimensional space," Proc. VLDB Endowment, vol. 2, no. 1, pp. 109–120, 2009.

[9] A. Vlachou, C. Doulkeridis, K. Nørva_ g, and Y. Kotidis,"Identifying the most influential data objects with reverse top-k queries," Proc. VLDB Endowment, vol. 3, no. 1, pp. 364–372, 2010.

[10] T. Wu, Y. Sun, C. Li, and J. Han, "Region-based online promotion analysis," in Proc. Int. Conf. Extending Database Technol.: Adv. Database

Technol., 2010, pp. 63–74

[11] M. Das, S. Amer-Yahia, G. Das, and C. Yu, "MRI: Meaningful interpretations of collaborative ratings," Proc. VLDB Endowment,vol. 4, no. 11, pp. 1063–1074, 2011.

[12] Y. Zhang, Y. Jia, and W. Jin, "Promotional subspace mining with EProbe framework," in Proc. ACM Conf. Inf. Knowl. Manage., 2011,pp. 2185–2188

[13] T. Lappas, G. Valkanas, and D. Gunopulos, "Efficient and domain-invariant competitor mining," in Proc. ACM SIGKDD Int.Conf. Knowl. Discovery Data Mining, 2012, pp. 408–416

[14] A. Arvanitis, A. Deligiannakis, and Y. Vassiliou, "Efficient influence-

based processing of market research queries," in Proc. ACM Conf. Inf. Knowl. Manage., 2012, pp. 1193–1202.

[15] E. Wu and S. Madden, "Scorpion: Explaining away outliers in aggregate queries," Proc. VLDB Endowment, vol. 6, no. 8,pp. 109–120, 2013.

[16] A. Sultana, N. Hassan, C. Li, J. Yang, and C. Yu, "Incremental discovery

of prominent situational facts," in Proc. Int. Conf. Data Eng., 2014, pp. 112–123.