1K-1

ビジネスデータに向けたR*-TREEへの新しいクラスタリング標準の導入
(A HYBRID CLUSTERING CRITERION FOR R*-TREE ON OLAP DATA)

Yaokai FENG Zhibin WANG and Akifumi MAKINOUCHI
Graduate School of Information Science and Electrical Engineering, Kyushu University

1 INTRODUCTION

There is increasing requirement for processing multidimensional range queries on business data usually stored in relational tables. In order to obtain good performance for such multidimensional range queries, multidimensional indices (e.g., R*-tree is famous) are helpful, in which the tuples are clustered among the leaf nodes to restrict the nodes to be accessed for a query.

In this paper, first, it is pointed out that, when the R*-tree is used for indexing business data, the clustering pattern of tuples among the leaf nodes is a decisive factor on range search performance. But, there exist many very slender leaf nodes when R*-tree is used to index business data, which greatly degrades query performance. Slender nodes means those having a very narrow side (even the side length is zero) in some dimension. Clearly, slender nodes have very small, even 0, areas (volumes). According to our discussion in this paper, the reason of so many slender leaf node existing becomes clear. The insert algorithm of R*-tree, especially, its criterion choosing subtrees for new coming objects, determines the clustering pattern of the tuples among the leaf nodes. After that, we make it clear that the present clustering criterion in the insert algorithm of R*-tree is not suitable to R*-tree applied to business data. Instead, a hybrid clustering criterion for the insert algorithm of R*-tree is proposed. Our discussion and experiments indicate that query performance of R*-tree on business data is improved much by the new clustering creation.

2. SLENDER NODES AND THEIR EFFECT

Because of the particularity of business data, some new features occur when R*-tree is used to index business data.

As a feature of business data, the data ranges of the attributes are very different from each other. For instance, the data range of “Year” from 1990 to 2003 is only 13 while the amount of “Sales” for different ``Product" may be up to several hundreds of thousands. Another typical example of such domains with small cardinalities is Boolean attribute, which has inherently only two possible values.

According to our observations, there are many slender leaf nodes, or even 0-area leaf nodes when R*-tree is applied to business data. Again, slender nodes mean those having a very narrow side (even zero side) in some dimension. Some examples are those MBRs roughly shaped as line segments in 2-dimensional spaces and roughly shaped as plane segments in 3-dimensional spaces.

The basic reason that so many slender leaf nodes exist is the distribution of the tuples in the index space. Because the possible different values in some index dimensions (attributes) are few. And, the existing of slender leaf nodes is a “positive feedback”. That is, once some slender leaf nodes exist, they will become more and more as the new tuples are inserted. In other words, the existing of slender leaf nodes promotes generation of many new such nodes, which greatly deteriorates search performance.

The existing of slender nodes leads to some problems.

Let us consider the insertion algorithm of R*-tree, using the example depicted in Figure 1 (a). Point p is to be newly inserted. Certainly it should be inserted in Node B since it is so nearer to Node B than to Node A. However, according to the insert algorithm of the R*-tree, p will be inserted to Node A in this case. This is because the area increment of doing so is smaller than that of inserting p to Node B. In other words, the new-coming tuples tend to be inserted into the existing slender nodes. This will lead to a bad clustering of tuples among the leaf nodes, which greatly cut down query performance.

Let us to see another case shown in Figure 1(b). There are two MBRs shaped as line segments, A and B. Let assume p is a point to be inserted. Intuitively, p should be included in Node B whose MBR is a line segment. Actually, p may be inserted in Node A, although this enlarges the overlap (between A and B) and also leads to a long node A. This is because the insertion algorithm of the R*-tree cannot determine which node, A or B, should be selected since both volume increment and overlap increment of selecting A and selecting B are 0. As a result, either Node A or Node B is selected as default without consideration of actual overlap. Here, we assume that Node A is selected. When a new point with the same y-axis coordinate as p is inserted again, the same process is repeated and the point is also inserted into Node A.

The repeated insertion of such points leads to the overflow of Node A. The node is split into Node A' and Node A''. Repeated insertions of points like p leads to node A splitting again and again, which generate a new Node A''', and so on. As a result, the space utilization of such nodes degrades and the total number of nodes tends to increase. Moreover, the heavy overlaps among the leaf nodes also greatly influence the search performance.

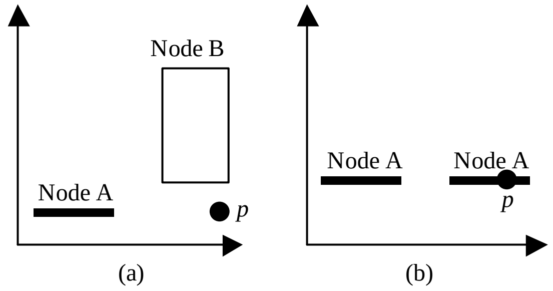

Figure 1. Slender nodes exist.

3. A HYBRID CLUSTERING CRITERION

The present clustering criterion is as follows.

(a) The least enlargement of overlap area, if tie occurs then (b) The least enlargement of MBR area, if tie occurs then (c) The least MBR area.

Our approach to this problem includes the following two points.

(1) Modifying the area calculation.

The reason that "no way to decide a suitable subtree (or leaf node) for new-coming tuples" like p1, p2, p3 in Figure 5 is that the enlargements on overlap area and enlargements on MBR area are zero and comparison can not be made for inserting the tuples to nodes A and B. And whichever of A and B is chosen, the area of the result MBR is zero.

In order to avoid this situation, we change the area calculation. That is, when the area of a rectangle, a node MBR or the overlap region of two node MBRs, is calculated, if exist, all the zero-sides (i.e., the side length is zero) of this rectangle is set to a trivial non-zero positive value (e.g., 10^{-4} in our experiments). dimensionality of the index space.

The modified area calculation of *R* is as follows.

$$Area'(R) = \prod_{i=1}^{d} S_i',$$
$$S_i' = \begin{cases} trivial - value & S_i = 0, \\ S_i & otherwise, \end{cases}$$

where trivial-value is set to 10^{-4} in this paper. Anyway, this trivial value must be less than the unit in this attribute, which is not difficult to guarantee. In this way, many un-comparable situations caused by slender nodes can be avoided. Note that, this modification only changes the clustering pattern of the tuples among the leaf nodes and it has no effect on the correctness of the query result.

(2) Introducing a distance-criterion.

If the above area-criterion cannot decide which subtree or leaf node is most suitable to the new-coming tuples, which means the area-based clustering criterion is no longer in force, the nearest subtree or leaf node to the new-coming tuple is chosen.

4. EXPERIMENTS

Dataset and index attributes: Lineitem table of TPC-H benchmark, which has 16 attributes of various data types including floating, integer, date, string, Boolean. The table used in our experiments has 200,000 tuples. Six of the total 16 attributes are chosen as index attributes, including SHIPDATE(date), QUANTITY(floating point), DISCOUNT(floating point), SHIPMODE(character string), SHIPINSTRUCT(character string), and RETURNFLAG (character string), since they are often used as query attributes in the queries of the benchmark.

The result is included in Table 1.

Table 1. Comparison on the number of accessed different nodes

Query range	R*-tree with *original clustering criterion*	R*-tree with *hybrid clustering criterion*
10%	369.91	95.12
20%	648.90	126.33
30%	603.65	131.31
40%	388.67	137.30
50%	683.29	237.27
60%	489.00	248.10
70%	708.24	231.10
80%	691.89	275.48
90%	571.10	357.62
100%	764.55	358.49

From Table 1, we can know that the hybrid clustering criterion can greatly improve the query performance.

5. CONCLUSIONS

In this paper, a hybrid clustering criterion is introduced to R*-tree applied to OLAP data, which clearly improved query performance of R*-tree.

REFERENCES

Y. Feng, Z. Wang, A. Makinouchi. *A Hybrid Clustering Criterion for R*-tree on Business Data*. Proc. 7th ICEIS Intl. Conf. , 2005. (to appear)