The World in a Nutshell:Concise Range Queries

Abstract

With the advance of wireless communication technology, it is quite common for people to view maps or get related services from the handheld devices, such as mobile phones and PDAs. Range queries, as one of the most commonly used tools, are often posed by the users to retrieve needful information from a spatial database. However, due to the limits of communication bandwidth and hardware power of handheld devices, displaying all the results of a range query on a handheld device is neither communication efficient nor informative to the users. This is simply because that there are often too many results returned from a range query.

In view of this problem, we present a novel idea that a concise representation of a specified size for the range query results, while incurring minimal information loss, shall be computed and returned to the user. Such a concise range query not only reduces communication costs, but also offers better usability to the users, providing an opportunity for interactive exploration.

The usefulness of the concise range queries is confirmed by comparing it with other possible alternatives, such as sampling and clustering. Unfortunately, we prove that finding the optimal representation with minimum information loss is an NP-hard problem. Therefore, we propose several effective and nontrivial algorithms to find a good approximate result. Extensive experiments on real-world data have demonstrated the effectiveness and efficiency of the proposed techniques.

Existing System

Facing the huge amount of spatial data collected by various devices, such as sensors and satellites, and limited bandwidth and/or computing power of handheld devices, how to deliver light but usable results to the clients is a very interesting, and of course, challenging task. For our

purpose, light refers to the fact that the representation of the query results must be small in size, and it is important for three reasons.

First of all, the client-server bandwidth is often limited. This is especially true for mobile computing and embedded systems, which prevents the communication of query results with a large size. Moreover, it is equally the same for applications with PCs over the Internet. In these scenarios, the response time is a very critical factor for attracting users to choose the service of a product among different alternatives, e.g., Google Map versus Mapquest, since long response time may blemish the user experience. This is especially important when the query results have large scale.

Second, clients' devices are often limited in both computational and memory resources. Large query results make it extremely difficult for clients to process, if not impossible. This is especially true for mobile computing and embedded systems.

Third, when the query result size is large, it puts a computational and I/O burden on the server. The database indexing community has devoted a lot of effort in designing various efficient index structures to speed up query processing, but the result size imposes an inherent lower bound on the query processing cost. If we return a small representation of the whole query results, there is also the potential of reducing the processing cost on the server and getting around this lower bound. As we see, simply applying compression techniques only solves the first problem, but not the latter two. none of the clustering techniques work well for the concise range query problem since the primary goal of clustering is classification. An important consequence of this goal is that they will produce clusters that are disjoint.

Proposed System

We focus on the problem of finding a concise representation for a point set P with minimum information loss. We show that in one dimension, a simple dynamic programming algorithm finds the optimal solution in polynomial time. However, this problem becomes NP-hard in two dimensions. Then, we settle for efficient heuristic algorithms for two or higher dimensions. The

above BFS traversal treats all nodes alike in the R-tree and will always stop at a single level. But, intuitively, we should go deeper into regions that are more "interesting," i.e., regions deserving more user attention. These regions should get more budgets from the k bounding boxes to be returned to the user. Therefore, we would like a quantitative approach to measuring how "interesting" a node in the R-tree is, and a corresponding traversal algorithm that visits the R-tree adaptively. In the algorithm R-Adaptive, we start from the root of the R-tree with an initial budget of k, and traverse the tree top-down recursively. Suppose we are at a node u with budget _, and u has b children u1; . . . ; ub hose MBRs are either completely or partially inside Q. Let the counts associated with them be n1; . . . ; nb. Specifically, if BR(ui) is completely inside Q, we set ni ¼ nui ; if it is partially inside, we compute ni proportionally as in

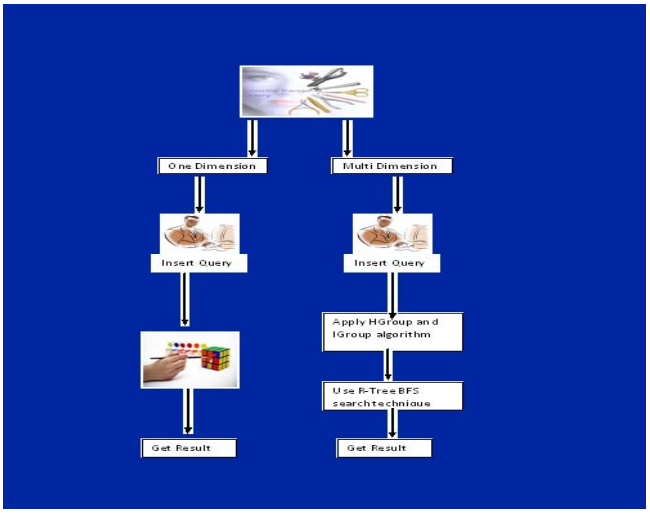
$$n_u \cdot \frac{Area(\text{MBR } (u) \cap Q)}{Area(Q)},$$

Solving Techniques:

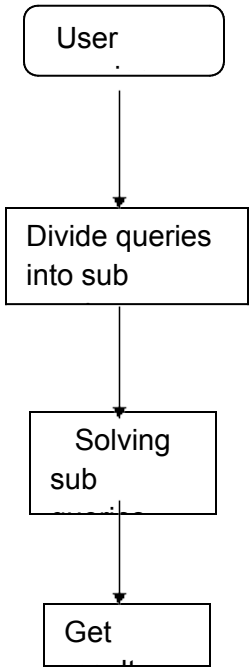
In one dimension, a simple dynamic programming algorithm finds the optimal solution in polynomial time. However, this problem becomes NP-hard in two dimensions. Then, we settle for efficient heuristic algorithms for the problem for two or higher dimensions.

Diagrams

Architecture



Optimal Solution Technique Process for one dimension



Module Description:

1) Input data

Spatial databases have witnessed an increasing number of applications recently, partially due to the fast advance in the fields of mobile computing and embedded systems and the spread of the Internet. For example, it is quite common these days that people want to figure out the driving or walking directions from their handheld devices (mobile phones or PDAs). However, facing the huge amount of spatial data collected by various devices, such as sensors and satellites, and limited bandwidth and/or computing power of handheld devices, how to deliver light but usable results to the clients is a very interesting, and of course, challenging task**.** Collected spatial data are provided as an input**.**

2) One dimension

In one dimension, a simple dynamic programming algorithm finds the optimal solution in polynomial time. We first give a dynamic programming algorithm for computing the optimal concise representation for a query.

3) Multi dimension

Since our problem is also a clustering problem, it is tempting to use some popular clustering heuristic, such as the well-known k-means algorithm, for our problem as well. However, since the object function makes a big difference in different clustering problems, the heuristics designed for other clustering problems do not work for our case. The k-anonymity problem does share the same object function with us, but the clustering constraint there is that each cluster has at least k points, while we require that the number of clusters is k. These subtle but crucial differences call for new heuristics to be tailored just for the concise representation problem.then we use R-Tree BFS searching algorithm.

4) View result

The search results are viewed. The query result size significantly reduced as required by the user. The reduced size saves communication bandwidth and also the client's memory and computational resources, which are of highest importance for mobile devices. Second, although the query size has been reduced, the usability of the query results has been actually improved. The concise representation of the results often gives the user more intuitive ideas and enables interactive exploration of the spatial database. Finally, we have designed R-tree-based algorithms so that a concise range query can be processed much more efficiently than evaluating the query exactly, especially in terms of I/O cost.

SYSTEM SPECIFICATION:

HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.

- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Colour.
- Mouse : Logitech.
- Ram : 512 MB.

SOFTWARE REQUIREMENTS:

- Operating system : Windows XP Professional.
- Coding Language : ASP .Net,C#
- Database : Sql Server 2005.

Reference:

Ke Yi, Xiang Lian, Feifei Li and Lei Chen, "The World in a Nutshell: Concise Range Queries", **IEEE Transactions on Knowledge and Data Engineering, Vol. 23, No.1, January 2011.**