B. Sunil Kumar, B. RAJASEKHAR, S Chandana / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 2,Mar-Apr 2012, pp.1269-1273

A Frame Work on Nutshell using Unsupervised ConciseRangeQueries

B. Sunil Kumar, B. RAJASEKHAR, S Chandana

Abstract - Wireless communication technology is more advance in 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. 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 and 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. 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.

Keywords - Unsupervised Learning, Histogram, spatial databases, Wireless sensor networks.

INTRODUCTION I

In several real time applications like spatial databases is an important role including geographical information systems, decision support intelligent transportation and resource allocation. Spatial database become a powerful tool is its ability to manipulate such as model, index and query but not simply store spatial data objects. Efficient query processing for spatial data objects has received considerable attention from the database research community, spatial queries include range query nearest neighbor search, spatial join and closer pair query. In various fields there is a need to manage geometric, geographic, or spatial data, which means data related to space. The space of interest can be, for example, the two-dimensional abstraction of the surface of the earth [1] that is, geographic space, the most prominent example, a man-made space like the layout of a VLSI design, a volume containing a model of the human brain, or another 3d-space representing the arrangement of chains of protein molecules. At least since the advent of relational database systems there have been attempts to manage such data in database systems. Characteristic for the technology emerging to address these needs is the capability to deal with large collections of relatively simple geometric objects, for example, a set of 100 000 polygons. This is somewhat different from areas like CAD databases (solid modeling etc.) where [2] geometric entities are composed hierarchically into complex structures, although the issues are certainly related. Several terms have been used for database systems offering such support like pictorial, image, geometric, geographic, or spatial database system. The terms “pictorial” and “image” database system arise from the fact that the data to be managed are often initially captured in the form of digital raster images (e.g. remote sensing by satellites, or computer tomography in medical applications). The term “spatial database system” has become popular during the last few years, to some extent through the series of conferences “Symposium on Large Spatial Databases (SSD)” held bi-annually since 1989 [Buch89, GünS91, AbO93], and is associated with a view of a database as containing sets of objects in space rather than images or pictures of a space. Indeed, the requirements and techniques for dealing with objects in space that have identity and well-defined extents, locations, and relationships are rather different from those for dealing with raster images. It has therefore been suggested [3] to clearly distinguish two classes of systems called spatial database systems and image database systems, respectively [GünB90, Fra91]. Image database systems may include analysis techniques to extract objects in space from images, and offer some spatial database functionality, but are also prepared to store, manipulate and retrieve raster images as discrete entities. In this survey we only discuss spatial

database systems in the restricted sense. Several papers in this special issue address image database problems and so complement the survey.

SECTION II

2. Related Work on other Alternatives

2.1. Histogram: In histogram construction specially consists of several buckets each of which stores the frequency of data points in it. The histogram has been widely used as a tool for selectivity estimation for example in the construction of V-optimal /Min Skew histogram aims to find partitions to minimize the error of selectivity estimation which is defined as summing up the multiplication of frequency and the statistical variance of the spatial densities of all points grouped within that bucket. Difference between MinSkew does not allow overlapping buckets resulting in disjoint partitions of the data set. Note, however, that here the three obtained buckets only minimize the spatial-skew, rather than the information loss (which is defined as summing up the multiplication of frequency and the extent of each bucket).

2.2. Random Sampling: It is easy to see that it is inferior to our result in the sense that in order to give the user a reasonable idea on the data set a sufficient number of samples need to be drawn especially for skewed data distributions example using k=3 bounding boxes roughly corresponds to taking six random samples with a high concentration of points in the down town area. Indeed random sampling is a very general solution that can be applied on any type of queries. In fact the Seminal work of proposed to use a random sample as an approximate representation of the results of a join and designed nontrivial algorithms to compute such a random sample at the early stages of the query execution process. The fundamental difference between their work and ours is that the results returned by a range query in a spatial database are strongly correlated by the underlying geometry. For instance, if two points p and q are returned, then all the points in the database that lie inside the bounding box of p and q must also be returned. Such a property does not exist in the query results of a join. Thus, it is difficult to devise more effective approximate representations for the results of joins than random sampling. On the other hand, due to the nice geometric and distributional properties exhibited by the range query results, it is possible to design much more effective means to represent them concisely.

2.3. Unsupervised Learning: Our analysis could be interpreted as a new clustering problem if we return the underlying partitioning P instead of the concise representation R. similarly for existing clustering problems one could return instead of the actual clusters only shapes of the clusters and the numbers of points in the clusters. This will deliver a small representation of the data set. Consider the example in fig 1 which shows a typical distribution of interesting pints such as restaurants near a city found in a spatial database. There are a large number of points in a relatively small downtown area. The suburbs have a moderate density while the points are sparsely located in the countryside.

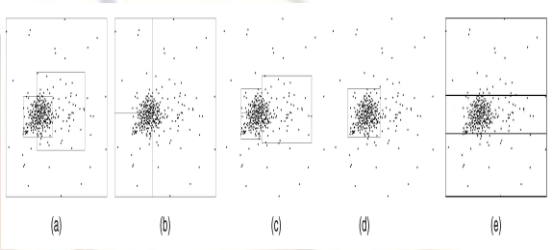

Figure 1 shows the clustering technique to find the queries in spatial databases

In Fig. 1a omit the counts here, the downtown area is summarized with a small box with many points. The suburb is grouped by a larger box that overlaps with the first box note that its associated count does not include those points contained in the first box and all the outliers from the countryside are put into a very large box. One can verify that such a solution indeed minimizes the information loss (1). The intuition is that in order to minimize (1), we should partition the points in such a way that small boxes could have a lot of points while big boxes should contain as few as possible. If adding a new point to a cluster increases the size of its bounding box, then we need to exercise extra care, as it is going to increase the "cost" of all the existing points in the cluster. In other words, the cost of each point in a cluster C is determined by the "worst" points in C. It is this property that differentiates our problem with all other clustering problems, and actually makes our definition an ideal choice for obtaining a good concise representation of the point set. the modified k-means approach is shown in Fig. 1b. Here, we also use the bounding box as the "shape" of the clusters using the center, radius pair would be even worse.) Recall that the objective function of k-means is the sum of distance or squared of each point to its closest center. Thus, in this example, this function will be dominated by the downtown points, so all the three centers will be put in that area, and all the bounding boxes are large. This obviously is not a good representation of the

point set: It is not too different from that of, say, a uniformly distributed data set.

One may argue that the result in Fig. 1b is due to the presence of outliers. Indeed, there has been a lot of work on outlier detection, and noise-robust clustering [6]. However, even if we assume that the outliers can be perfectly removed and hence the bounding boxes can be reduced, it still does not solve the problem of putting all three centers in the downtown (Fig. 1c). As a result, roughly 1/3 of the downtown points are mixed together with the suburban points. Another potential problem is, what if some of the outliers are important? Although it is not necessary to pinpoint their exact locations, the user might still want to know their existence and which region they are located in. representation (Fig. 1a) with k = 3 only tells the existence of these outliers. But as we increase k, these outliers will eventually be partitioned into a few bounding boxes, providing the user with more and more information about them. Fig. 1d shows the result obtained by a density based clustering approach. A typical density based clustering, such as CLARANS [7], discovers the clusters by specifying a clustering distance _. After randomly selecting a starting point for a cluster, the cluster starts to grow by inserting neighbors whose distance to some current point in the cluster is less than _. This process stops when the cluster cannot grow any more. This technique, when applied to our setting, has two major problems. First, we may not find enough clusters for a given k, assume that there is a support threshold on the minimum number of points in one cluster). In this example, we will always have only one cluster. Second, the clusters are quite sensitive to the parameter Є. Specifically, if we set Є small, then we will obtain only the downtown cluster (Fig. 1d); if we set Є large, then we will obtain the cluster containing both the downtown and the suburb. Neither choice gives us a good representation of the point set.

SECTION III

3. Problem Definition: Wireless Communication becomes most widely used techniques and common for people to view maps or get related services from the handheld devices such as mobile phones and PDAs. To view the range queries as one of the most commonly used tools are often posed by the users to retrieve needful information from spatial databases. 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. To view the range of queries we present 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.

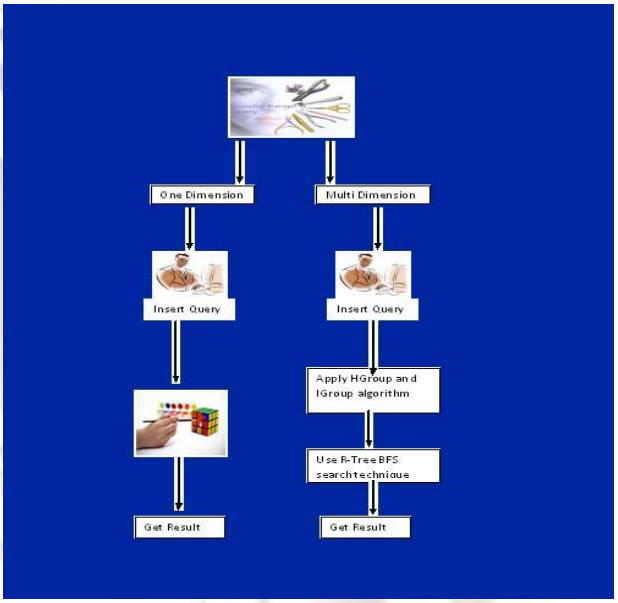

Figure 2 Proposed Architecture

3.1. Input to system: 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**.**

3.2. Dimensional View: 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.

3.3. View Query Results: 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.

SECTION IV

4. Comparative Study: 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. Compare to previous 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 ttention. These regions should get more budgets from the k bounding boxes to be returned to the user. Therefore, we would ike a quantitative approach to measuring how "interesting" a node in the R-tree is, and a corresponding traversal algorithm hat 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

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

CONCLUSION V

A concise range queries, has been proposed in this paper, which simultaneously addresses the following three problems of traditional range queries. First, it reduces the query result size significantly 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. This concept, together with its associated techniques presented here, could greatly enhance user experience of spatial databases, especially on mobile devices, by summarizing "the world in a nutshell."

Reference

[1] Abdelmoty, A.I., M.H. Williams, and N.W. Paton, Deduction and Deductive Databases for Geographic Data Handling. Proc. 3rd Intl. Symposium on Large Spatial Databases, Singapore, 1993, 443-464.

[2] Abel, D.J., SIRO-DBMS: A Database Tool Kit for Geographical Information Systems. *Intl. J. of Geographical Information Systems 3 (1989)*, 103-116.

[3] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger, "The R - Tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD Int'l Conf. management of Data, pp. 322-331, 1990.

[4] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," Proc. Symp. Principles of Database Systems (PODS), 2006.

[5] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A.W.-C. Fu, "Utility- Based Anonymization Using Local Recoding," Proc. ACM SIGKDD, 2006.

[6] C. Bo¨hm, C. Faloutsos, J.-Y. Pan, and C. Plant, "RIC: Parameter- Free Noise-Robust Clustering," ACM Trans. Knowledge Discovery from Data, vol. 1, no. 3, pp. 10-1-10-28, 2007.

[7] R.T. Ng and J. Han, "Efficient and Effective Clustering Methods for Spatial Data Mining," Proc. Int'l Conf. Very Large Data Bases (VLDB), 1994.

[8] A. Guttman, "R-Trees: A Dynamic Index Structure for Spatial Searching," Proc. ACM SIGMOD, 1984.

[9] N. Beckmann, H.P. Kriegel, R. Schneider, and B. Seeger, "The R- Tree: An Efficient and Robust Access Method for Points and Rectangles," Proc. ACM SIGMOD, 1990.

[10] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," Proc. ACM SIGMOD, 1996.

[11] V. Ganti, R. Ramakrishnan, J. Gehrke, and A. Powell, "Clustering Large Datasets in Arbitrary Metric Spaces," Proc. Int'l Conf. Data Eng. (ICDE), 1999.

B. Sunil Kumar completed M.C.A from OU Hyderabad, pursuing M.Tech through JNTU Hyderabad. I am presently working as Assistant Professor in Department of Master of Computer Applications in Jawaharlal Nehru Institute of Technology, Hyderabad. I am having 3years of Teaching Experience. My interested subjects are Data mining, Web services, Web Technologies, Java, C, and C++.....

B. RAJASEKHAR completed B.Tech in CSE and M.Tech in SE from JNTU Hyderabad. He is presently working as Assistant Professor in Department of Computer Science & Engineering in Jawaharlal Nehru Institute of Technology, Hyderabad. I am having 3years of Teaching Experience. His interested subjects are Data mining, Image Processing, algorithms, data structures, c, uml.....

S Chandana B.Tech & pursuing M.Tech from JNTU Hyderabad. Currently she is working as Asst Prof in Jawaharlal Nehru Institute of Technology, research areas include Data mining and Network Security.