International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol.2, Issue.1, Jan-Feb 2012 pp-522-524 ISSN: 2249-6645

# EFFICIENT QUERY PROCESSING IN SPARSE DATABASE BY AVOIDING SUBSPACE

**S.Dhiyanesh*, Mrs.DeviSelvam****

*II M.E CSE,Sri shakthi Institute Of Engineering and Technology,Anna University,Coimbatore
**Asst.prof CSE,Sri shakthi Institute Of Engineering and Technology,AnnaUniversity,Coimbatore

## Abstract
Sparse data are common and available in many real life applications. The Sparse data sets are used in e-commerce application. An E-commerce dataset may have thousands of attributes, but most of the values are null and only a few of while apply to a particular queries to a database. In existing RDBMS objects are conventionally stored using horizontal representation, vertical representation and HoVer representation in this paper. According to the dimension correlation of sparse datasets, a novel mechanism has been developed to conduct query for sparse datasets by improving the 'Hover' technique. Therefore the original data objects are represented in a database format in respective subspaces.

**Index Terms:** Sparse database, query processing, correlation, subspace, Hover.

## I. INTRODUCTION
With continuous advances in the network and storage technology, there is dramatic growth in the amount of very high-dimensional sparse data from a variety of new application domains, such as bioinformatics, time series, and perhaps, most importantly e-commerce [1], [3], which pose significant challenges to RDBMSs.
The main characteristics of these sparse data sets may be summarized as follows:

**High dimensionality**: The dimensionality of feature vectors may be very high, i.e., the number of possible attributes for all objects is huge. For example, in some e-commerce applications, each participant may declare their own idiosyncratic attributes for the products, which results in data sets that have thousands of attributes [1].

**Sparsity:** Each object may have only a small subset of attributes, which is called active dimensions, i.e., significant values appear only in few active dimensions;
In addition, different objects may have different active dimensions. For example, an e-commerce data set may have thousands of attributes, but most of which are null and only a few of which apply to a particular product.

**Correlation:** Since each object may have only few active dimensions, more likely, similar objects share same or similar active dimensions. For example, in recommendation systems, it is important to find homogeneous groups of users with similar ratings in subsets of the attributes. Therefore, it is possible to find certain subspaces shared by similar objects. In existing RDBMSs, objects are conventionally stored using a horizontal format called the horizontal representation in this paper.

## II. RELATED WORK
In RDBMSs, sparse data sets are typically represented by the horizontal representation. The horizontal representation is straightforward and can be easily implemented; however, it may suffer from schema evolution, column number limitation, and poor storage and query performance incurred by sparsity [2]. Besides the horizontal representation, there are two approaches which can be used to store and conduct query for sparse data sets in an unmodified RDBMS, i.e., the vertical representation [1] and the decomposition storage model [4].

In [1], an alternative of the horizontal representation, i.e., the vertical representation, was investigated. The vertical representation has been used to represent sparse data sets for

Querying [1] as well as representing dense data sets for data mining [5]. In the vertical representation, a single row in the horizontal representation is split into multiple rows. Each row contains an object identifier, an attribute name, and a value.

In addition, the Resource Description Framework (RDF) data of the Semantic Web is commonly stored using this format.

The vertical representation can scale to thousands of attributes, avoid storage of null values, and support evolving schemas. However, writing queries over this representation is cumbersome and error-prone. An inspection of a single row in the horizontal representation becomes a multiway self-join over the vertical table. Another problem of the vertical representation is that it is difficult to support multiple data types. One approach is to create a separate table for each data type, but the approach makes the query rewriting more complicated.

There are some other works about vertical partition of horizontal tables where the subspaces are generated according to the characteristics of the fixed query workload over the data set. On the contrary, our work explores inherent properties of sparse data sets, and our solution is data-oriented, i.e., subspaces are generated according to the distribution of data.

## III. THE HOVER REPRESENTATION

### *A. THE HoVer REPRESENTATION*

The pure horizontal or vertical representation is introduced that may yield unsatisfactory performance in sparse databases. Therefore, we propose a new representation called HoVer, which can effectively exploit the characteristics of sparse data sets, such as sparsity and dimension correlation. We aim at achieving good space and time performance for storing and querying high-dimensional sparse data sets.

Although the dimensionality of sparse data sets could be very high, up to thousands, a single data object typically has only a few active dimensions, and similar objects have a better chance to share similar active dimensions. A closer inspection of many e-commerce sparse data sets shows that typical e-commerce data sets have a wide variety of items which can be organized into categories and the categories themselves are hierarchically grouped; items that belong to a common category are likely to have common attributes, while those within the same subcategory are likely to have more common attributes. The RDF data [6] also shows that the attributes of similar subjects tend to be defined together. This motivates us to find certain subspaces which are shared by similar data groups, and to split the full space into some lower-dimensional subspaces.

On the other hand, our purpose is to split the full space into subspaces which can yield superior performance for the storage and query of sparse data. These approaches are not suitable for this scenario.

In this section, present sparse data is introduced, these sets using the novel HoVer representation. First, we design an efficient and effective approach to find correlated dimensions. After that, the original full space into subspaces and store the original sparse data set using multiple tables where each table corresponds to a certain subspace.

### *B. CORELATED DEGREE REPRESENTATION*

Before subspace selection, we first consider how to measure the correlation between two dimensions

**Definition 1 (Correlation Table):** The correlation table represents the correlation of dimensions in a sparse data set, which is a super triangle matrix.

**Definition 2 (Correlated Degree):** The correlated degree is used to measure the correlation between two dimensions in which dimension i(j) is active, and tuples in which dimensions i and j are active simultaneously. According to set theory, it characterizes the number of tuples in which at least one of the two dimensions i and j is active.

### *C. SUBSPACE SELECTION*

An optimal subspace partitioning should enjoy two properties, i.e., all dimensions are highly correlated intersubspaces while being highly unrelated intersubspaces. If the number of subspaces determined by the user is smaller, dimensions which are not highly correlated may be clustered into the same subspace; hence, the subspace tables are still very sparse.

According to our above analysis, the number of subspaces should be determined by the subspace selection algorithm according to the dimension correlations of the sparse data set. Because the underlying storage and query processing details of the RDBMS may have some influence on the performance [3], there may not exist perfect subspace clustering typically.

Our subspace selection problem can be mapped to the Minimum Clique Partition problem. While mapping each dimension in the sparse data set to a node in the graph, and if the correlated degree between two dimensions is no less than the correlated degree threshold.

Add an edge to link the two corresponding nodes in the graph, and our subspace selection problem is exactly same as the Minimum Clique Partition problem. Unfortunately, the Minimum Clique Partition problem is NP-complete [7], which means that heuristic algorithm is used to approximate optimal partitions which tries to group together correlated dimensions. With a smaller threshold, fewer subspaces will be generated, but the nonnull density of each subspace will be smaller. Actually, the optimal correlated degree threshold varies for different data sets.

### *D. SCHEMA EVALUATION*

When a new column is added, a new subspace which only contains the new column will be created, and the correlation table should also be updated accordingly. Since the correlation table is incrementally maintained, the new column may be merged to a subspace when subspaces are reorganized. When a column is deleted, then there is a need to delete the column from the corresponding subspace and update the correlation table accordingly.

## IV. QUERY PROCESSING IN HOVER

### *A. QUERY REWRITING*

Our ultimate purpose is to define horizontally represented views over the HoVer representation. Users typically issue traditional SQL queries over the horizontal view, which can be rewritten into queries over the underlying HoVer representation.

In our work, the dimensions in the original sparse data space are clustered into subspaces, and a horizontal vertically partitioned into subspace tables. In many real-life applications, the dimensions with a high correlated degree are likely to characterize similar topics and have high probability of being accessed together [8]; hence, they should be stored in the same subspace table. Then take advantage of this characteristic and access as few subspace tables as possible in query evaluation. A query rewriting table which records the relationships between the dimensions and the subspaces is essential for query rewriting.

### *B. EFFECT OF CORELATED DEGREE THROSHOLD*

The correlated degree threshold has great influence on the query performance based on the HoVer representation. Using a large threshold, the average size of the subspace tables is small, and the average nonnull density of the subspace tables is large.

In this case, queries over the horizontal representation have high probability of being rewritten into queries over the join of many small subspace tables and the cost of join operations could be high, because the accessed columns have high probability of being distributed into many subspace tables.

In our work, the correlated degree threshold is a tuning parameter and the correlated degree is only characterized by the distribution of the data. Previous work [8], [6] and our experimental study over real-life data sets in Section 5 verify that the correlated degree can well characterize the correlations between dimensions in most of the cases. Besides the subspace partitioning, the diversity of query workload can also significantly impact the overall query performance, as different queries may access various data objects and attributes. Integrating the characteristics of the query workload for subspace selection is a promising alternative to optimize the performance of database for query processing.

## V. CONCLUSION

In this paper, the problem of efficient query processing over sparse databases is addressed. To alleviate the suffering from sparsity and high-dimensionality of sparse data, a new approach is introduced as named HoVer. According to the characteristics of sparse data sets, then vertically partition the high-dimensional sparse data into multiple lower-dimensional subspaces, and all the dimensions in each subspace are highly correlated, respectively. The proposed scheme can find correlated subspaces effectively, and yield superior storage and query performance for conducting queries in sparse databases.

## REFERNCES

[1]. R. Agrawal, A. Somani, and Y. Xu, "Storage and Querying of E-Commerce Data," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 149-158, 2001.

[2]. R. Agrawal, R. Srikant, and Y. Xu, "Database Technologies for Electronic Commerce," Proc. Int'l Conf. Very Large Data Bases (VLDB), pp. 1055-1058, 2002.

[3]. J.L. Beckmann, A. Halverson, R. Krishnamurthy, and J.F. Naughton, "Extending RDBMSs to Support Sparse Datasets Using an Interpreted Attribute Storage Format," Proc. Int'l Conf. Data Eng. (ICDE), p. 58, 2006.

[4]. G.P. Copeland and S. Khoshafian, "A Decomposition Storage Model," Proc. ACM SIGMOD, pp. 268-279, 1985.

[5]. S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating Association Rule Mining with Relational Database Systems: Alternatives and Implications," Data Mining and Knowledge Discovery, vol. 4, nos. 2/3, pp. 89-125, 2000.

[6]. K. Wilkinson, C. Sayers, H.A. Kuno, and D. Reynolds, "Efficient RDF Storage and Retrieval in Jena2," Proc. Int'l Workshop Semantic Web and Databases (SWDB), pp. 131-150, 2003.

[7]. Paz and S. Moran, "Non Deterministic Polynomial Optimization Problems and Their Approximations," Theoretical Computer Science, vol. 15, pp. 251-277, 1981.

[8]. J. Beckham, R. Krishnamurthy, and J.F. Naughton, "The Tradeoff between Horizontal and Vertical Representations of Sparse Data Sets," technical report, http://www.cs.wisc.edu/~sekar/ application/sekar_ecommerce.pdf, 2003.

## AUTHORS

**Mr. S.Dhiyanesh** received B.E degree in CSE from Anna University, Chennai and Currently pursuing M.E degree in Computer Science and Engineering in Sri Shakthi Institute of Engineering and Technology, under Anna University of Technology, Coimbatore. His research interest includes Data Mining.

**Mrs. DeviSelvam** received the M.E. and B.E. degree in CSE from Avinashilingam University, Coimbatore. She is currently working as Assistant Professor in Department of CSE in Sri Shakthi Institute of Engineering and Technology, Coimbatore. Previously she got a good experience in SRM University, Chennai. She has presented papers in National and International Conferences. She is member of IEEE. Her main research interests include Computer Networks, Mobile computing and Data mining.