IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 12, Issue 5 (Jul. - Aug. 2013), PP 45-48
www.iosrjournals.org

K Means Clustering Algorithm for Partitioning Data Sets Evaluated From Horizontal Aggregations

R. Rakesh Kumar[1], A. Bhanu Prasad[2]
[1](*M.Tech, Software Engineering, Vardhaman College of Engineering/ JNTU-Hyderabad, India)*
[2](*Associate Professor, Department of IT, Vardhaman College of Engineering/ JNTU-Hyderabad, India)*

Abstract: *Data mining refers to the process of analyzing the data from different perspectives and summarizing it into useful information that is mostly used by the different users for analyzing the data as well as for preparing data sets. A data set is collection of data that is present in the tabular form. Preparing data set involves complex SQL queries, joining tables and aggregate functions. Traditional RDBMS manages the tables with vertical format and returns one number per row. It means that it returns a single value output which is not suitable for preparing a data set. This paper mainly focused on k means clustering algorithm which is used to partition data sets after horizontal aggregations and a small description about the horizontal aggregation methods which returns set of numbers instead of one number per row. This paper consists of three methods that is SPJ, CASE and PIVOT methods in order to evaluate horizontal aggregations. Horizontal aggregations results in large volumes of data sets which are then partitioned into homogeneous clusters is important in the system. This can be performed by k means clustering algorithm.*
Keywords: *Aggregations, SQL, pivoting and K-means clustering.*

I. Introduction

Data mining is the process of extracting knowledge from large volumes of data. It has attracted a great deal of attention in the information industry and in society as a whole in recent years due to the wide availability of huge amount of data and the imminent need for turning such data into useful information and knowledge. Data can be stored in different kinds of databases and information repositories. One such data repository architecture that has emerged is the data warehouse. Data warehouse technology includes OLAP (Online Analytical Processing), that is, analysis technique with functionalities such as summarization, consolidation and aggregation. Data aggregation is a process in which information is gathered and expressed in a summary form, and which is used for purposes such as statistical analysis. Aggregation is normally associated with data reduction in relational databases. The aggregate functions available in SQL are MIN, MAX, AVG, SUM and COUNT. All these functions returns a single number as output. This is called vertical aggregation. The output of vertical aggregations is helpful in calculation. Most of the data mining operations require a data set with horizontal layout with many tuples and one variable or dimension per column.
This paper contains three fundamental methods that are used to evaluate Horizontal aggregations: they are case, SPJ (Select Project Join) and pivot: case, SPJ (Select Project Join) and pivot.

Case method:

This method uses the "case" programming construct available in SQL. The case statement returns a selected value based on Boolean expressions from a set of values.

SPJ method:

The SPJ method is work based on the relational operators only. The basic concept in SPJ method is to build a table with vertical aggregation for each resultant column. And then join all those tables to produce another table.

Pivot method:

It is a built-in operator offered by some DBMS (Data Base Management System) concepts, which transforms row to columns. This method internally needs to determine how many columns are needed to store the transposed table and it can be combined with the GROUP BY clause.

Clustering:

Clustering is the process of grouping a set of objects into clusters in such a way that objects in the same cluster are more similar to each other than to those in other clusters. It is a common technique for statistical data analysis which is used in many fields like machine learning, pattern recognition, image analysis, information retrieval, and bio informatics. Clustering is a main task of exploratory data mining. Clustering can be achieved

by various algorithms which differ significantly in their notion of what constitutes a cluster and how to efficiently find them. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and also intended use of the results.

The horizontal aggregations provide several unique features and advantages. There advantages include they

- They represent a template to generate SQL code from a data mining tool. This SQL code automates writing SQL queries, optimizing them, and testing them for correctness. SQL code reduces manual work in the data preparation phase in a data mining.
- SQL code is more efficient than SQL code written by an end user as it is automatically generated. As a result, data sets can be created in less time.
- The data set can be created entirely inside the DBMS. In modern database environments, it is common to export denormalized data sets to be further cleaned and transformed outside a DBMS in external tools (e.g., statistical packages). Unfortunately, exporting large tables outside a DBMS is very slow and it creates inconsistent copies of the same data and effects database security.

Therefore, we provide a more efficient, better integrated and more secure solution compared to external data mining tools. Horizontal aggregations just require a small syntax extension to aggregate functions called in a SELECT statement. Alternatively, they can be used to generate SQL code from a data mining tool to build data sets for data mining analysis.

II. Proposed Methodology

Clustering is a method of partitioning a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria. Data mining applications frequently involve categorical data. The biggest advantage of these clustering algorithms is that it is scalable to very large data sets.

We need different attributes from multiple fact tables in order to construct a new data set from the range of discrete points of known data sets. In many applications one often has a number of data values, obtained by experimentation, which stored on limited number of databases. It is often required to extract particular attributes that are useful from the different fact tables and perform aggregation.

2.1. K means algorithm

K-means is one of the simplest unsupervised learning algorithms which is used to solve the well known clustering problem. The procedure employs a simple and easy way to classify a given data set through a certain number of clusters fixed a priori. We assume K clusters. The main idea of this algorithm is to define k centroids, one for each cluster. These centroids should be placed in a way that different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step in the algorithm is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroid as the clusters resulting from the previous step. After we have these k new centroids, a new binding is performed between the same data set points and the nearest new centroid. As a result, a loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

Finally, this algorithm aims at minimizing an *objective function*, which is a squared error function. The objective function is given by the following equation.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(J)} - c_j \|^2$$

Where $\| x_i^{(J)} - c_j \|^2$ is a chosen distance measure between a data point $x_i^{(J)}$ and the cluster centre c_j , is an indicator of the distance of the *n* data points from their respective cluster centers.

K-means algorithm which is based on classification technique uses horizontal aggregation as input. Pivot operator is used to calculate the aggregation of particular data values from distinct multiple fact tables. Optimization provides for PIVOT for large number of fact table. The database connectivity and choosing different tables with .mdb extension is the first step in this system.

K means algorithm consists of the following four steps. They are

1. Place K points into the space represented by the objects which are data sets that are being clustered. These points represent initial group centroids.
2. Assign each data object to the group that has the closest centroid.
3. When all objects have been assigned to different clusters, recalculate the positions of the K centroids.

4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into clusters from which the metric to be minimized can be calculated.

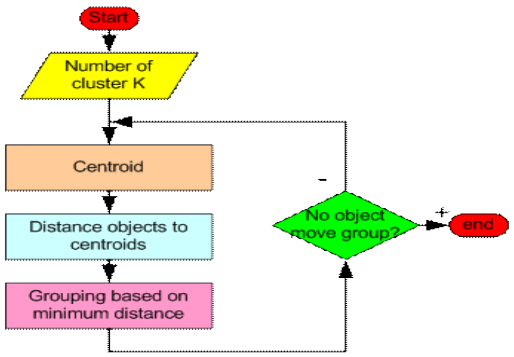

Fig: flow chart of k means algorithm

Step by step example of k means algorithm:

The k means algorithm is implemented on large data sets that are resulted after the horizontal aggregation in order to partition into different clusters. An example with step by step procedure applied on a data set is given below.

As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of seven individuals:

Subject	A	B
1	1.0	1.0
2	1.5	2.0
3	3.0	4.0
4	5.0	7.0
5	3.5	5.0
6	4.5	5.0
7	3.5	4.5

This data set which is given above in the table is to be grouped into two clusters. As a first step in finding a sensible initial partition, let the A & B values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster,

	Individual	Mean Vector (centroid)
Group 1	1	(1.0, 1.0)
Group 2	4	(5.0, 7.0)

The remaining data sets are now examined in a sequence and are allocated to different cluster to which they are closest, which is calculated terms of Euclidean distance to the cluster mean. When mean vector is recalculated, each time a new member is added. This gives the following series of steps:

	Cluster 1		Cluster 2	
Step	Individual	Mean Vector (centroid)	Individual	Mean Vector (centroid)
1	1	(1.0, 1.0)	4	(5.0, 7.0)
2	1, 2	(1.2, 1.5)	4	(5.0, 7.0)
3	1, 2, 3	(1.8, 2.3)	4	(5.0, 7.0)
4	1, 2, 3	(1.8, 2.3)	4, 5	(4.2, 6.0)
5	1, 2, 3	(1.8, 2.3)	4, 5, 6	(4.3, 5.7)
6	1, 2, 3	(1.8, 2.3)	4, 5, 6, 7	(4.1, 5.4)

Now the initial partition is changed, and the two clusters at this stage having the following characteristics:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2, 3	(1.8, 2.3)
Cluster 2	4, 5, 6, 7	(4.1, 5.4)

But we are not sure that each individual has been assigned to the right cluster. In order to know that each individual is assigned to the right cluster, we compare each individual's distance to its own cluster mean and to that of the opposite cluster.

Individual	Distance to mean (centroid) of Cluster 1	Distance to mean (centroid) of Cluster 2
1	1.5	5.4
2	0.4	4.3
3	2.1	1.8
4	5.7	1.8
5	3.2	0.7
6	3.8	0.6
7	2.8	1.1

Hear only individual 3 is nearer to the mean of the opposite cluster (Cluster 2) than its own (Cluster 1). In other words, each individual's distance to its own cluster mean should be smaller than the distance to the other cluster's mean (which is not the case with individual 3). Thus, individual 3 is relocated to Cluster 2 resulting which results in the new partition:

	Individual	Mean Vector (centroid)
Cluster 1	1, 2	(1.3, 1.5)
Cluster 2	3, 4, 5, 6, 7	(3.9, 5.1)

The iterative relocation would now continue from this new partition until no more relocation occur. However, in the example given above each individual nearer its own cluster mean than that of the other cluster and the iteration steps, choosing the latest partitioning as the final cluster solution. I this way the k means algorithm is performed on the data sets that are evaluated from horizontal aggregations.

III. Conclusion

The paper extended the horizontal aggregation methods with k means clustering algorithm. Conventional horizontal aggregation gives data from single fact table as an input. But in this paper the input to the algorithm is the data from the multiple fact tables. In order to partition the data set into different clusters we used the Euclidean distance computation, pivoting a table to have one dimension value per row. Pivot is a data manipulating operator which is easy to compute wide set of values. It is an extension of Group By with unique restrictions and optimization opportunities, and this makes it easy to introduce incrementally on top of existing grouping implementation. The paper also consists of a simple example explaining the step wise procedure of k means clustering algorithm. Optimized k-means is significantly faster because of small data set run clustering outside the DBMS.

References

[1] C. Ordonez and Z. Chen. Horizontal aggregations in SQL to prepare data sets for data mining analysis.IEEE Transactions on Knowledge and Data Engineering (TKDE), 24(4), 2012.
[2] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and subtotal. In ICDE Conference.
[3] G. Bhargava, P. Goel, and B.R. Iyer. Hypergraph based reordering of outer join queries with complex predicates. In ACM SIGMOD Conference, pages 304–315, 1995.
[4] J.A. Blakeley, V. Rao, I. Kunen, A. Prout, M. Henaire, and C. Kleinerman. .NET database programmability and extensibility in Microsoft SQL Server. In Proc. ACM SIGMOD Conference, pages 1087–1098, 2008.
[5] J. Clear, D. Dunn, B. Harvey, M.L. Heytens, and P. Lohman. Non-stop SQL/MX primitives for knowledge discovery. In ACM KDD Conference, pages 425–429, 1999.
[6] E.F. Codd. Extending the database relational model to capture more meaning. ACM TODS, 4(4):397–434, 1979.
[7] C. Galindo-Legaria and A. Rosenthal. Outer join simplification and reordering for query optimization. ACM TODS, 22(1):43–73, 1997.

Text Book:

[1] H. Garcia-Molina, J.D. Ullman, and J. Widom. Database Systems: The Complete Book. Prentice Hall, 1st edition, 2001.
[2] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco, 1st edition, 2001.