Spectral clustering algorithm based on Hadoop cloud platform research and implementation

LiSheng Zhang, Ling Hou，DaJiang Lei

[1]Chongqing University of Posts and Telecommunications, Chongqing 400000, China;

[2]Chongqing University of Posts and Telecommunications, Chongqing 400000, China.

Chongqing University of Posts and Telecommunications, Chongqing 400000, China.

zhangls@cqupt.edu.cn,houlingstudent@163.com, leidj@foxmail.com

Keywords: cloud computing, MapReduce programming model, Hadoop cloud platform, spectral clustering.

Abstract. Spectral clustering algorithm based on the theory of spectrum, its meaning is the optimal clustering problem into graph partitioning problem is a point of clustering algorithms can be high-dimensional data set cluster after dimensionality reduction. Greatly reducing the time of clustering. Compared with the traditional clustering algorithm, spectral clustering which can have the advantage of clustering and converge to the global optimal solution in the sample space of arbitrary shape. However, the prevalence of large data sets are in the real world, when we want to clustering the spectral of large data sets, because the data is too large, the convergence rate will slow down, if not impossible to obtain results within the stipulated time we give us a lot of problems cluster. Thus, this paper based on Hadoop cloud platform to achieve large-scale clustering high-dimensional data sets. Experiments show that: spectral clustering algorithm after the parallel deployments running on Hadoop clusters, with good speedup and good scalability.

1. Introduction

Cluster analysis is an important research on data mining, and therefore has been development quickly. Because of its simple, spectral clustering algorithm has been implementation simply, having a strong theoretical foundation and it is capable of high-dimensional data dimensionality reduction and effective stand. The main idea of it is to be seen as a data point vertices in the graph, the similarity reduction and effective stand. The main idea of it is to be seen as a data point vertices in the graph, the similarity between points is seen as edges in the graph, complete data set clustering theory based on spectrum division.

In the face of a large amount of data, spectral clustering[1] algorithm convergence speed becomes very slowly, even it can’t get the clustering result in the effective time, paralleling spectral clustering algorithm can greatly improve the spectral clustering algorithm clustering algorithm can greatly improve the spectral clustering algorithm clustering efficiency when it is in large-scale data environment. Traditional high-performance parallel computing models such as OpenMP and MPI, etc.[2-3], although they could improve the efficiency of spectral clustering algorithm, but still there are many defects, such as abstract level is not high, developers need to be familiar with the underlying parallel configuration and implementation details and so on. Because of its simply and abstractly, Hadoop users only need to focus on their own parallel tasks to be solved without the need to understand the details too much, distributed programming has been simplifies greatly, but it also been reduced the cost greatly, so this paper shows the spectral clustering algorithm parallelization based on Hadoop cloud platform. Experimental results show that compared with serial, spectral clustering algorithm which been achieved based on Hadoop has better efficiency, performance, and scalability.

2. Basic thinking in spectral clustering algorithm.

Spectral clustering algorithm, as a kind of point pair clustering algorithm, utilizes spectrum division theory, transforming clustering problem in the traditional sense into graph partitioning problem. It regards data point as vertex V in the graph, and the similarity between the data as weighted value W at the edge of the graph. Thus, we get an undirected weighted graph G= (V, E) based on the similarity between data points. Result after the graph partitioning is to make the highest similarity between points within the sub-graph, and the lowest similarity between points in the sub-graph. We use similarity matrix namely, adjacency matrix, to represent undirected weighted graph, figuring out the eigenvectors clustering corresponding to the first k of the smallest eigenvalues, representing the division of the graph. As a result, spectral clustering algorithm is associated to the number of data points, and independent of the dimension of data points, carrying out the dimensionality reduction effectively. Spectral clustering algorithm includes calculating the similarity matrix, calculating the diagonal matrix, calculating the Laplace matrix, calculating the first k smallest eigenvalues of the Laplace matrix and their corresponding eigenvectors, K-Means clustering.

Spectral clustering is suitable for handling a multidimensional vector, and dimension reduction purpose can be achieved by processing. Optimal division criteria based on graph theory is to make the highest inner similarity of the two divided sub-graphs, and the lowest similarity between sub-graphs. The quality of division criteria affects the merits of the clustering results directly[4].

Steps of spectral clustering algorithm.

Step1. Calculating the similarity matrix

We use a weighted undirected graph to represent the data set, vertex of undirected weighted graph to represent data point V of the data set, and the weighted value W at the edge of the graph to represent the similarity between points. Then, an undirected weighted graph G(V, E) is obtained based on sample similarity. In this way, clustering problems turn into graph partition problems. And then we use the adjacency matrix to represent the information in the graph.

Using Gauss similarity function to calculate the similarity between two points

$$w_{ij}=\exp(-||x_i-x_j||^2/2\sigma^2) \quad (1)$$

In this formula, O is the scale parameter, and needs to be entered manually, its size affecting the result of obtained similarity. We take the similarity matrix as W, and each element in W is the similarity between the point and other points. Wij represents the similarity between point i and point j, so, Wij = Wji. By similar calculation, we can get the similarity between each point and other points, combining together as similar matrix. Similarity matrix records the similarity between points.

Step2. Calculating the diagonal matrix

The main diagonal element of line i of the diagonal matrix is the sum of all elements of the i-th row of matrix W, because the i-th row of matrix W represents the similarity between point i and other points. Therefore, the main diagonal of line i of the diagonal matrix represents the sum of similarity between point i and other points. We note diagonal matrix as D. Dii represents the sum of similarity between point i and other points, and elements are 0 except for those at the diagonal position, $D_{ii}=W_{i0}+W_{i1}+W_{i3}+\cdots\cdots+W_{in}$. According to this process, we can obtain the diagonal matrix.

Step3. Calculating the Laplace matrix

Laplace matrix can be divided into non-normalized Laplace matrix and normalized Laplace matrix. In this paper, we adopt the non-normalized Laplace matrix, denoted by L, L=D-W, that is the subtraction of elements of the diagonal matrix and their corresponding elements of the similarity matrix. Thus, the i-th row of Laplace matrix reserves similarity between point i and other points, as well as the total sum of similarity, to retain all the information in the graph.

Step4. Calculating the first k smallest eigenvalues and their corresponding eigenvectors of the Laplace matrix

In this paper, we adopt QR decomposition approach to solve the first k smallest eigenvalues and their corresponding eigenvectors of the Laplace matrix.

The essence is: for any n-order real symmetric matrix, there is an orthogonal matrix Q making Q^TLQ equals to a diagonal matrix, the values on the diagonal equal to eigenvalues of the matrix L, and column vectors of Q are eigenvectors corresponding to their eigenvalues. So we decompose $L_k=Q_kR_k$, where K is the orthogonal matrix, R_k is an upper triangular matrix, $L_k=Q_kR_k$, $L_{k+1}=Q^T_k L_kQ_k=R_kQ_k$. Through continuous iteration, L eventually equal to a diagonal matrix, the values on diagonal is the eigenvalues of L, and each column of R is its corresponding eigenvector. We only need to compare the size of the eigenvalues on the diagonal, and find out the smallest k eigenvalues and their corresponding eigenvectors.

Step5. K-Means clustering

We use the eigenvectors corresponding to the first k smallest eigenvalues to arrange into a N * K matrix in terms of column, and see each row of the matrix as a vector in K-dimensional vector space. Thus, we effectively reduce the dimension. Now we're going to cluster these N n-dimensional vectors. The category of each row in the clustering result each row is the category which the initial first N data points belong respectively. K-Means clustering process is, first of all, to choose m initial centers of the randomly inputted dada, calculating distances between each point and the center point, and then assigning to the nearest clustering center, updating the k clustering centers. If the change of the center point is less than the threshold, end the cycle.

3. Spectral clustering algorithm parallelization

Through our research on the spectral clustering algorithm[5], we found that in the calculate the similarity matrix, calculated the k smallest eigenvalues of Laplace matrix and the corresponding eigenvectors and K-Means clustering, they can be paralleled.

3.1 Parallel computing similarity matrix W

There are n points in the data set, we use HBase to store these n points, so we can use a table to store the n points, with two table two to store the similarity of the data which we calculate. the i-th data point is stored in the i-th row of table one and the i-th row of table two memory similarity between i-th point and other points. We remove any two data points from Table one, We calculate their similarity of them by Gaussian function and then re-storing them on the second table. Since W is a symmetric matrix, after removal point i, i just need to calculate the first i + 1 points, the first similarity i + 2……section i + n points on it, so, from point one more backward point and other points to calculate the similarity of work less and less. So, we deal with each map in the experiment two points, in order to balance the workload, the first map and a first handle n points, and the first two points of the first n-1 second map processing……

3.2 Computing the k smallest eigenvalues and the corresponding eigenvectors by paralleled

When we use the QR algorithm for the k smallest eigenvalue of Laplacian matrix L and its corresponding eigenvector[6], we found that the first step seeking Q is can't be paralleled. That because when we schmidt orthonormal the column vector L, it will be used on the next step of the cycle results. But when we computing $R_k = Q^T_k L_k$ that can be paralleled, each row of Q is multiplied by each column L, independently of each other; in the iterative process, $L_{k+1} = R_kQ_k$ can be paralleled, for the same reason.

3.3 Parallelization K-Means

In the K-Means algorithm[7], all points were calculated and assigned to the nearest cluster center and update the K cluster centers can be paralleled. All points be computed and assigned to the nearest cluster center can use many map tasks to achieve, update the K cluster centers can be implemented using multiple reduce tasks. Program loop to perform multiple MapReduce programs, each MapReduce corresponding to one iteration of serial spectral clustering. First, the program randomly generates the k cluster centers and stored in HDFS, then it operated multiple MapReduce tasks, each operation is performed first map MapReduce task to send the output of the map data to different Reduce, the final implementation of judgment function, the return value judgment function decides whether to perform the next operation according to MapReduce.

4. Experiment analysis

4.1 Comparison between parallel and serial experimental

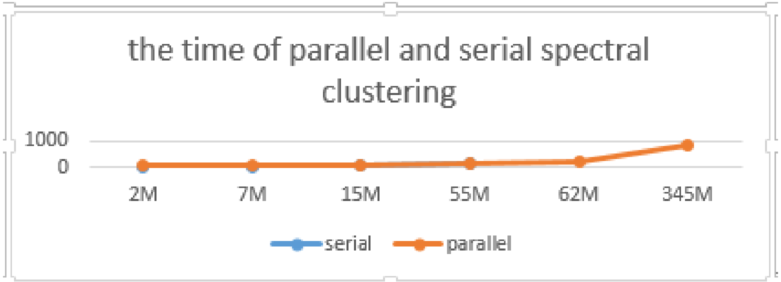

Fig. 1 the time of parallel and serial spectral clustering

4.2 The effects of different numbers nodes on the parallel spectral clustering

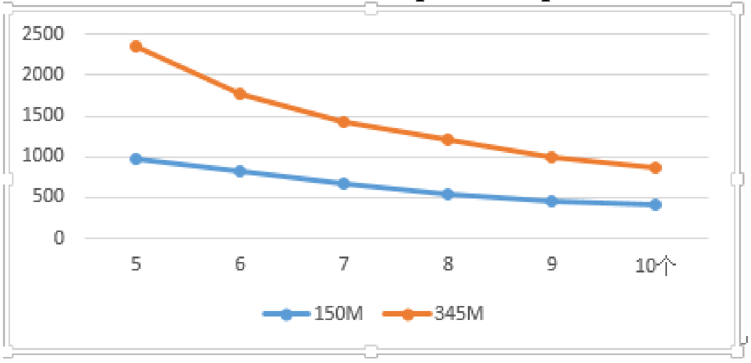

Fig. 2 the time of different nodes

5. Summary

Spectral clustering algorithm based on Hadoop cloud platform parallelism which is an effective solution to run on a single machine, that is too slow even when the results can't be affected. Through the above experiments we can conclude that, when a small amount of data, spectral clustering run slower on parallel than on a single machine, but after reaching a certain amount of data, parallel faster than serial , with an additional amount of data serial appears insufficient memory. And the higher the number of nodes in parallel, spectral clustering algorithm faster. Therefore, we effective solute the lack of spectral clustering on single machine.

References

[1] Naisbitt J. Megatrendss: Ten new directions tansforming our lives[M].New York:Warner Books.1982.16-17.

[2] Han J W. Kamber M. Data mining:concepts and echniques[M]. San Francisco. US:Morgan Kaufmann.2001.

[3] Dean J,Ghemawat S. MapReduce: simplified data processing on large clusters[J].Communications of the ACM.2005,51(1):107-113..

[4] Boutsinas B, Gnardellis T, On Distributing the Clustering Procss[J]. Patter Recognitio Letters,2002.23(4):999-1008.

[5] Chu C T. Kim S K,Lin Y A.MapReduce for machine learning on multicore[C] //Proceedings of Neural Information Processing Systems Conference (NIPS).Boston:MIT.2006:281-288.

[6] Apache Hadoop. Hadoop[EB/OL].[2011-02-15].htp://hadoop.apache.org.

[7]Erdogmus H.Cloud computing:does nirvanahide behind the nebula[J].IEEE Software .2009.26(2):4-6.