A Parallel K-means Algorithm for High Dimensional Text Data

Xiaolei Shan,Yanming Shen,Yuxin Wang
Dalian University of Technology

***Abstract*--This paper proposed a Parallel K-means Algorithm for High Dimensional sparse Text data (PKHT). By using GPU （Graphic Processing Unit） and MPI （Message-passing Interface）, the proposed algorithm achieves a up to 11x lower running time .**

I. INTRODUCTION

High dimensional data clustering is the key point of clustering analysis. Text clustering algorithm can be divided roughly into two categories. One is based on hierarchical clustering and the other is based on partition clustering. In this paper, the popular k-means clustering algorithm[1] is adopted. According to the different characteristics of the text data, the data is divided into different classes. For high-dimensional data, using Minkowski Distance to construct a cluster is not appropriate, so the combination of TF-IDF[2] and cosine is used instead. For the initial seed selection of clustering algorithm, the proposed algorithm is based on the idea of density and k-means ++[3]. For k-means clustering algorithm, the main bottleneck is matrix computing. Therefore, we can achieve parallel matrix computing by using GPU. As the amount of data increases, a single GPU can not meet the requirements. Therefore, MPI[4] technology is applied to realize multi-GPU computation. This paper uses the universal parallel parallel computing framework CUDA to implement GPU[5]. Experimental results show that the proposed has a higher accuracy , and a small time overhead on large data sets.

II. THE ALGORITHM

The algorithm proposed in this paper involves the design of three parts, which are the design of clustering algorithm suitable for high-dimensional sparse data, the GPU implementation of parallelizable part of algorithm, the data distribution of MPI part and the inter-process signal transfer design. Figure 1 shows the overall framework of the algorithm.

A. Clustering Algorithm Design

First, for the text data, after the document vectorized representation, the data is preprocessed by the idea of dimensionality reduction. Dimensionality reduction methods can be divided into feature selection and feature transformation[6]. In the feature selection method, we attempt to pick out the feature items from source data[6]. In this paper, we take advantage of this idea and choose the typical method TF-IDF method to realize feature selection.

Second, the k-means algorithm was chosen as the clustering algorithm. In the initial seed determination step, we mainly use the idea of k-means ++ algorithm, except for the method of randomly selecting the first initial point, we use the maximum density point as the first initial seed. This avoids the random selection of anomalous points. But in the process of the whole clustering, for distance measurement, we adopt the cosine similarity function to measure the distance between text objects.

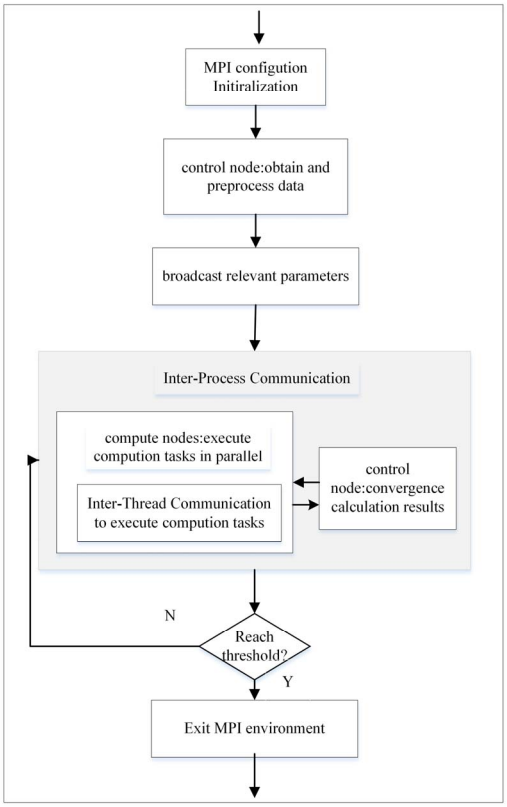

Fig 1 the overall framework of the algorithm

B. GPU Part Design

This paper uses CUDA computing platform to realize the GPU.The inherent time complexity of the k-means algorithm is $O(nkdl)$, where n is the number of documents, d is the number of words, k is the number of clusters and l is the number of iterations.This part of the design focuses on the measure of similarity between text objects, that is, the matrix calculation, the original calculation complexity is $O(nd)$ .In parallel design, it is designed to start n threads, which runs in parallel. Then the complexity becomes $O(d)$, which greatly reduces the algorithm running time.

C. MPI Part Implement

This section divides nodes into two types, one control node and several compute nodes respectively. Among them, the

control node is responsible for receiving external data and publishing the data to a remote storage window, and sending the control information, the data set, the initialized seed and other data to the computing node. The compute nodes perform corresponding calculation through the data set read from the window and the control information received from the control node and sends the calculation result to the control node. The control node receives the data and performs corresponding judgment, and performs other operations. Through the transfer of information between the two nodes, we can achieve parallel computing.

III. EXPERIMENT

In order to evaluate our proposed algorithm, we use the real data set obtained from BBC news, KEEL and data mining classic dataset classic4 as shown in Table I. For the experiments we used Intel(R)Core(TM) i3-7100 CPU @3.90GHz and GeForce GTX 1050, running the ubuntu 16.04.4.

TABLE 1
DATA SETS USED IN EXPERIMENTAL EVALUATION

data set	N	D	K
opt	5620	64	10
bbcs(sports)	737	4613	5
docb	7094	5896	4
bbcb(business)	2225	9638	5

This paper uses the classical k-means algorithm[6] , k-means++ algorithm and the proposed algorithm to test the selected test set respectively. Of the three algorithms, only the classic algorithm uses the Euclidean distance to measure the similarity, and the rest are cosines to measure the similarity.We use the Normalized Mutual Information(NMI) as the measurement standard of accuracy, NMI is an accurate measurement of the mismatch between the number of reference classes and the number of clusterings. The formula is shown in (1) :

$$I(X;Y)=\sum_{y\in Y}\sum_{x\in X}p(x,y)\log(\frac{p(x,y)}{p(x)p(y)})$$
$$U(X,Y)=2R=2\frac{I(X;Y)}{H(X)+H(Y)} \quad (1)$$
$$H(X)=\sum_{i=1}^{n}p(x_i)\log_b\frac{1}{p(x_i)}=-\sum_{i=1}^{n}p(x_i)\log_b p(x_i)$$

Where X and Y represent the operation results and the corresponding labels respectively, n is the number of documents, $p(x,y)$ is the joint distribution probability of x and y . The results shown in Table II are the average values obtained from 20 experiments. We can find that the proposed algorithm improves the accuracy of high-dimensional text data up to 3x.

In addition, to evaluate the running time of the proposed algorithm, we run the algorithm on CPU, GPU, and MPI + GPU platforms, respectively. As shown in Figure 2, the results obtained by averaging 20 times on each platform are shown where "Runtime" represents the clustering algorithm iterative part of the operation time. By contrast, it is clear that the

TABLE II
RECORD FOUR DATA SETS FOR THE NMI FOR DIFFERENT ALGORITHM

Algorithm	optdigit	bbcs	docb	bbcb
Classic Kmeans	**73.59%**	25.32%	26.96%	28.24%
K-means++	72.99%	67.43%	55.84%	**76.54%**
K-means++ L-&TF-IDF	49.46%	77.94%	53.21%	73.93%
PKHT	50.02%	**81.08%**	**60.70%**	75.04%

algorithm runs significantly faster with GPU acceleration. For the opt dataset, it is nearly 11x and for the other datasets it is at least 4x. The use of MPI and GPU, not only can improve the algorithm speed, but also can be easily ported to the GPU cluster.

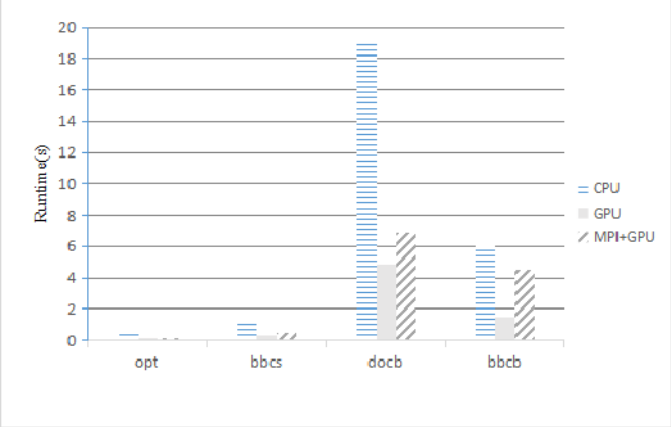

Fig 2 the algorithm running time

CONCLUSION

The experimental results show that the algorithm proposed in this paper not only has a good accuracy for clustering high-dimensional sparse text data, but also significantly improves the running speed through the use of CUDA framework and MPI.

REFERENCE

[1] Hartigan, J. A., and M. A. Wong. "Algorithm AS 136: A K-Means Clustering Algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, 1979, pp. 100 – 108. JSTOR.

[2] Salton, G. & Buckley, C. (1988). Term-weighing approache sin automatic text retrieval. In Information Processing & Management, 24(5): 513-523

[3] Arthur, David and Vassilvitskii, Sergei. k-means++: The advantages of careful seeding. In Symposium on Discrete Algorithms (SODA), pp. 1027 – 1035. Society for Industrial and Applied Mathematics,2007.

[4] Using MPI: Portable parallel programming with the message-passing interface[J]. Computers and Mathematics with Applications,2000,40(2).

[5] CUDA toolkit documentation, http://docs.nvidia.com/cuda/cuda-math-api/ , accessed November 27, 2017.

[6] Aggarwal, Charu C., and ChengXiang Zhai, eds. Mining text data. Springer Science & Business Media, 2012.