Large-scale Data Mining Method based on Clustering Algorithm Combined with MAPREDUCE

Yulun Zhang [1, *], Chenxu Zhang [1], Lei Yang [1], Hongyang Li [2]

[1] Minsk, School of Business, Belarusian State University, Belarus

[2] Department of Applied Mathematics and Informatics, Belarusian State University Minsk, Belarus

* Corresponding Author Email: zhangyulun0507@gmail.com

Abstract. With the continuous deepening and development of information technology, the diversity and amount of information in data continue to grow. Effectively mining these text data to extract valuable content has become an urgent task in the field of data research. This study combines the MapReduce distributed system with the K-means clustering algorithm to meet the challenges of large-scale data mining. At the same time, the paper use a distributed caching mechanism to solve the problem of repeated application of resources for multiple MapReduce collaborative operations and improve data mining efficiency. The combination of MapReduce's distributed computing and the advantages of K-means clustering algorithm provides an efficient and scalable method for large-scale data mining. Experimental results combining internal and external indicators show that the advantage of combining K-means with MapReduce is to fully utilize the distributed and parallel computing characteristics of MapReduce, providing users with an efficient and scalable data mining tool. Through this research, the paper provide new methods and insights for large-scale data mining, improving the efficiency and accuracy of data mining.

Keywords: Data Mining; Clustering Algorithm; Apache; Mapreduce; K-means Algorithm.

1. Introduction

As a powerful distributed computing tool, the MapReduce framework has been widely used in large-scale data processing, so it has an ideal platform for processing big data [1]. At the same time, K-means clustering algorithm, as a common data mining technology, is widely used to discover patterns and clusters in data [2]. By combining MapReduce with K-means, the paper are able to efficiently analyze large-scale data and identify clustering patterns and clusters in the data to provide strong support for decision making and insight discovery [3-4].

This article will first introduce the MapReduce distributed system and K-means clustering algorithm, and then explain how to combine them to meet the needs of large-scale data mining. Subsequently, the paper will verify the effectiveness of the method in experiments, and finally, the paper will summarize the advantages of combining MapReduce and K-means and their importance in the field of data mining.

Through this combination, the paper aim to better address the challenges of large-scale data mining, improve data processing efficiency, and at the same time deeply mine the inherent patterns and structures of data to provide strong support for decision-making and insight discovery.

2. Large-scale Data Mining Method based on Clustering Algorithm Combined with MAPREDUCE

2.1. Distributed Environment Layout based on Mapreduce for Data Mining Purposes

Clustering algorithm is one of the main methods for data processing in data mining. The distributed cloud computing data mining method based on MapReduce can be combined with the clustering algorithm to discover clustering patterns in the data, identify similar data points, and Generate information about data distribution and improve data efficiency. MapReduce is a distributed computing framework that adopts a distributed computing model that allows large-scale data sets to be broken down into multiple data blocks and processed in parallel on multiple computers [5-6]. data block. This parallelism makes processing big data more efficient and faster. An efficient way to process large data sets by breaking the problem into parallel tasks and performing distributed computations [7]. Use this principle to build a large-scale data mining model based on clustering algorithm combined with MAPREDUCE, as shown in Fig 1.

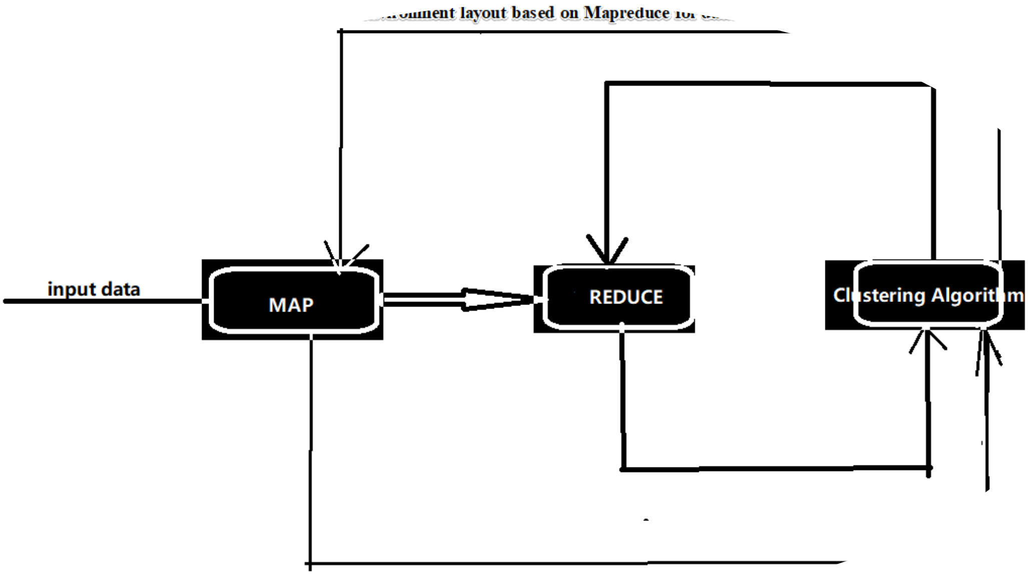

Fig 1. Operation diagram of MR and aggregation algorithm

In the MapReduce distributed large-scale data mining model, Map is responsible for data preprocessing and feature processing. The purpose is to prepare the original data into the form of a clustering algorithm. The Reduce stage is used to generate the final clustering results. Since there are many types of data mining models in the APACHE HADOOP cluster environment, and there are many types of clustering algorithms, this article selects Kmeans as the algorithm for this Mapreduce model[8]. Its advantages are its efficiency, scalability and performance in data clustering tasks. Computational efficiency. In large-scale text data mining, the K-means method can divide text data into different clusters so that similar data points belong to the same cluster, thus helping to mine patterns and structures in the data.

Combining the K-means clustering algorithm with the MapReduce framework is a powerful strategy especially suitable for performing cluster analysis on large-scale data sets. This combination takes full advantage of the many advantages of the MapReduce framework. First, it is scalable and can handle large-scale data efficiently because data can be distributed across multiple nodes without being loaded into memory at once. In addition, MapReduce's fault tolerance enables data mining tasks to continue executing without interruption even if failures occur in large-scale clusters. The data locality principle reduces data transmission overhead and improves computing efficiency. Task parallelism allows multiple Mapper nodes to execute K-means tasks at the same time, speeding up the calculation of the clustering algorithm[9]. The system's scalability makes it easy to add processing capabilities to meet growing data processing needs. In addition, MapReduce is often used in conjunction with distributed file systems (such as Hadoop's HDFS), which makes efficient storage and management of data possible. Finally, the Reduce stage of MapReduce can merge the clustering results generated by different Mapper nodes to generate the final clustering result, which facilitates the output sorting of the K-means algorithm.

This combination is suitable for scenarios where cluster analysis needs to be performed in a big data environment, such as text mining, image processing, and bioinformatics. It not only improves the efficiency of the K-means algorithm, but also makes it suitable for a wider range of data sizes and application fields. In summary, the advantage of combining K-means with MapReduce is that it takes full advantage of the distributed and parallel computing characteristics of the MapReduce framework and provides users with an efficient and scalable data mining tool.

2.2. Construction of Clustering Algorithm in the Context of Data Mining

K-means is a common clustering algorithm. Its goal is to divide data points into K different clusters so that data points in the same cluster are similar to each other and data points between different clusters are as dissimilar as possible. This algorithm achieves clustering by minimizing the sum of squares of the average distances of data points within a cluster (i.e., the within-cluster variance). Kmeans can cluster large-scale data in a short time [10]. By selecting different k values, controlling the number of clusters, and exploring different clustering structures to find the number of clusters that are most suitable for the data, this article plans to use Different two-dimensional data sets are randomly created as training sets. The process of Kmeans construction is as follows:

The K-means algorithm uses cluster centers to represent each cluster. For the i-th cluster, its cluster center can be expressed as:

$$C_i = \frac{1}{N_i}, \Sigma_{j=1}^{N} x_j \tag{1}$$

First, randomly set K points in the feature space as the initial clustering center, and find the Euclidean distance.

$$d = \sqrt{(x_n - x_1)^2 + (y_n - y_1)^2} \tag{2}$$

where n is the dimension of the data point, xi and y_i are the coordinates of the data points x and y on the i-th dimension,Then for each other point, the distance to K centers is calculated. For unknown points, the nearest cluster center point is selected as the label category,Furthermore, after facing the marked cluster center, recalculate the new center point (average value) of each cluster,Finally, if the calculated new center point is the same as the original center point (the center of mass no longer moves), then it is over, otherwise the second step is repeated[11].

3. Experiment and Analysis

When conducting large-scale data mining experiments with clustering algorithms combined with MAPREDUCE, our research methods and experimental designs follow the following logical steps:

First, the paper plan to generate multiple different 2D datasets to simulate different data distributions and cluster structures. The generation of these data sets will include different data set types such as random distribution and Gaussian distribution to reflect different data mining scenarios. Second, the paper will implement the K-means clustering algorithm ourselves. The implementation of this algorithm will include the initialization of cluster centers, allocation of data points, update of cluster centers, and iterative process to gain an in-depth understanding of the internal mechanism and working principle of K-means.

To evaluate the performance of K-means, the paper will use internal and external metrics. Internal metrics, such as within-cluster variance, can help us measure the quality of clustering. External indicators, such as silhouette coefficient, will also be used to evaluate the clustering effect. This will give us a quantitative measure of the clustering quality of K-means on different datasets.

During the experiment, the paper may try different K values and different parameter configurations to find the best clustering results [12]. Additionally, the paper will utilize visualization tools to plot the clustering results to better understand how K-means performs clustering on different data sets, as shown in Fig 2,3,4 and 5.

Import code blocks in jupyter based on Kmeans method

```
Import matplotlib.pyplot as plt
From sklearn.datasets.samples_generator import make_blobs
From sklearn.cluster import KMeans
From sklearn.metrics import calinski_harabaz_score
```

Create a data set with X as the sample feature and Y as the sample cluster category, with a total of 1,000 samples, Each sample has 2 features, a total of 4 clusters. Note that the cluster centers are [-1,-1], [0,0], [1,1], [2,2], and the cluster differences are [0.4, 0.2, 0.2, 0.2]

```
X,y=make_blobs(n_samples=1000,n_features=2,centers=[[-1,-1],[0,0],[1,1],[2,2]],
cluster_std=[0.4,0.2,0.2,0.2],
random_state=9)
plt.scatter(X[:,0],X[:,1], marker='o')
plt.show()
```

Try n_cluses=2\3\4 respectively, and then check the clustering effect

```
y_pred=KMeans(n_clusters=2,random_state=9).fit_predict(X)
plt.scatter(X[:,0],X[:,1],c=y_pred)
plt.show()
#use Calinski-Harabasz Index-
print(calinski_harabaz_score(X,y_pred))
```

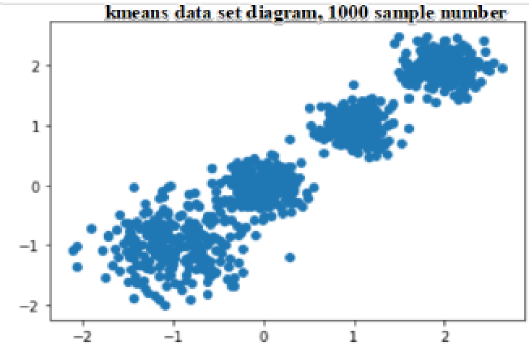

Fig 2. 1000 Sample Aggregation Grouping

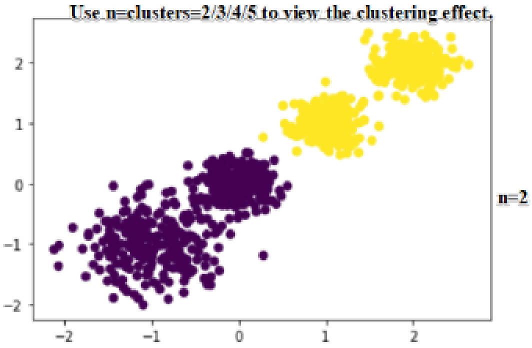

Fig 3. Kmeans adjustment n_ Cluster grouping diagram with clusters=2

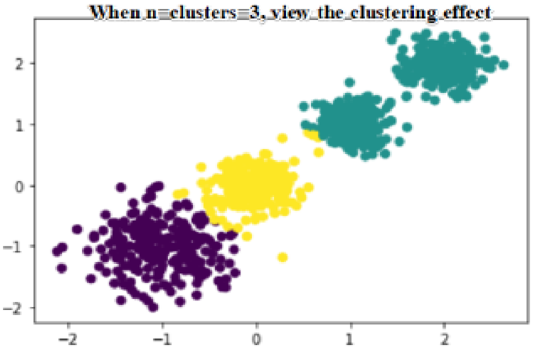

Fig 4. Kmeans adjustment n_ Cluster grouping diagram with clusters=3

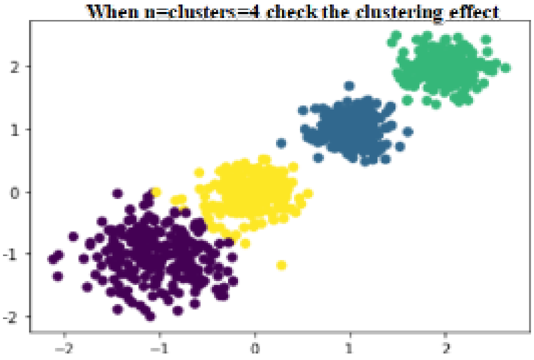

Fig 5. Kmeans adjustment n_ Cluster grouping diagram with clusters=4

In the context of data mining, the paper conducted a series of experiments aimed at exploring the performance and effect of the K-means clustering algorithm under different numbers of clusters (K values). By randomly generating different 2D datasets and trying clustering with K values of 2, 3, and 4, the paper emphasize the flexibility of K-means in data mining tasks and the criticality of K value selection.

the paper observe that different K values lead to significantly different clustering results, which highlights the importance of K value selection in the K-means algorithm. In practical data mining applications, choosing the correct K value is a key step to ensure meaningful clustering results. To quantify the clustering effect, the paper used the Calinski-Harabaz Index to evaluate the clustering quality under different cluster numbers. This metric helps us understand the performance of K-means under different K values and provides quantitative support for the selection of K values. in addition, through visual analysis, the paper clearly observed the clustering structure and data point distribution under different cluster numbers.

4. Conclusion

This paper implements a simple cluster analysis mining method by combining the data mining method based on MapReduce distributed system and K-means clustering algorithm. It mainly successfully draws on the MapReduce distributed system to provide ideal computing for large-scale data mining. platform. By sharding data and parallel computing, the paper can effectively deal with massive data, accelerate the computing process, and improve the efficiency of data mining. On the other hand, by dividing data points into different clusters, the paper can reveal the inherent patterns and structures between data, which is helpful for classification, recommendation, anomaly detection, etc., thus improving the efficiency of data mining.

References

[1] Qiao Yuanyuan, Liu Fang, Ling Yan, et al. Resource modeling and implementation of MapReduce in cloud computing environment Performance prediction [1] Journal of Beijing University of Posts and Telecommunications, 2014 (S1): 115-119.

[2] Li Zhenju, Li Xuejun, Yang Sheng, et al. MapReduce model based on multi-stage division [1] Computer Applications, 2015(12): 3374-3377 + 3382.

[3] Frenks B. Ukroshcheniye bol'shikh dannykh. Kak izvlekat' znaniya iz massivov informatsii s pomoshch'yu glubokoy analitiki [The taming of big data. How to extract knowledge from the massive amounts of information using deep Analytics]. Moscow, Mann, Ivanov i Ferber Publ., 2014. 352 p.

[4] M. V. Gladkiy, Belarusian State Technological University, DISTRIBUTED COMPUTING MODEL MAPREDUCE, BSTU 2016.NO_6,194-198.

[5] L. G. Valiant. A bridging model for parallel computation. Communications of the ACM, 33(8):103–111, 1997.

[6] Liao Bin, Zhang Tao, Yu Jiong, etc. Resource efficiency optimization of big data mining algorithms under the coordination of multiple MapReduce jobs [J]. Computer Application Research, 2020, 37(5): 1321-1325.

[7] Wang Bo, Wang Huaibin, Zhang Chao. Optimization of frequent pattern mining algorithm based on MapReduce[J]. Journal of Tianjin University of Science and Technology, 2018, 34(01): 6-11.

[8] Lu Guo, Xiao Ruixue, Bai Zhenrong, etc. Research on MapReduce parallel clustering optimization algorithm in big data mining [J]. Modern Electronic Technology, 2019, 42(11): 169-172.

[9] MapReduce model design based on heterogeneous computing[J]. Informatization Research, 2015(04): 40-43.

[10] Wan Cong, Wang Cuirong, Wang Cong, et al. Research on load balancing partitioning algorithm of reduce phase in MapReduce model [J]. Small Microcomputer Systems, 2015(02): 240-243.

[11] Liu Wei, Du Yongwen, Lu Xiaojian. Research on MapReduce model scheduling algorithm under Hadoop platform [J]. Journal of Guangxi University for Nationalities (Natural Science Edition), 2014(03): 72-74 + 85.

[12] Zhang Bin, Le Jiajin. MapReduce parallel connection algorithm based on column storage [J]. Computer Engineering, 2014 (08):70-75,85.