Affinity Propagation Clustering Algorithm based on Spark Platform

Lijia Zhang[a], Lianglun Cheng

School of Computer Science and Technology，Guangdong University of Technology，Guangzhou 510006，China.

[a]781176915@qq.com

Keywords: Affinity propagation, Resilient Distributed Datasets, Spark, Large scale dataset.

Abstract. With the explosive growing of data, there are challenges to deal with the large scale complex data. Many clustering algorithms have been proposed. Such as Affinity Propagation (AP) clustering Algorithm, AP takes similarity between pairs of data point as input measures. AP is a fast and efficient clustering algorithm for large dataset compared with the existing clustering algorithm. As the scale of data grows more explosively, the time efficiency of AP algorithm cannot be satisfied. Therefore, AP clustering algorithm based on Spark platform (Spark-AP) is proposed in this paper. Firstly, a dataset is partitioned into several Resilient Distributed Datasets (RDD) on a strategy and select the exemplars of each RDD. Then exemplars are merged and are used to next AP clustering algorithm, which forms a set of high-quality exemplars after convergence. Experiments show that Spark-AP performs better both in processing scale and processing time.

1. Introduction

Clustering algorithm has been widely applied to data mining, pattern classification and many other fields. With the rapid development of information and network technology, data size is increasing and data type is being complex, and the requirements of clustering efficiency and effect are also getting higher. For improving the efficiency and effectiveness of large-scale complex data clustering, distributed clustering algorithm[1] has been the focus of recent research.

The method of selecting the representatives of traditional clustering algorithm is that representatives are randomly selected firstly, and then representatives are being adjusted by calculating iteratively until representatives are no longer obviously changed or the iteration is completed. The selection of the initial representatives is connected with clustering results. AP clustering algorithm[2] take all data points of a dataset as possible representatives and then calculate Responsibility and Availability iteratively based on the similarity matrix to adjust representatives. Moreover, AP clustering algorithm is a fast and efficient clustering algorithm for large dataset.

RDD[3] of Spark[4] allows developers to perform large-scale calculations on the cluster based on memory, and has the capability of fault tolerance. In order to realize the clustering of massive data and improve the processing efficiency, many scholars do a lot of research for clustering algorithm based on Spark to improve the time efficiency of AP algorithm[5-7].

In this paper, AP clustering algorithm based on Spark platform (Spark-AP) is proposed in this paper. The clustering efficiency is improved by parallel clustering on Spark platform. Experimental results show that the efficiency of Spark-AP has significantly improved compared with the original clustering algorithm.

2. Affinity Propagation Clustering Algorithm

Frey and Dueck first proposed AP clustering algorithm in Science. AP clustering algorithm takes similarity between pairs of data point as input measures. AP clustering algorithm is a fast and efficient clustering algorithm for large dataset compared with the existing clustering algorithm. AP clustering algorithm take all data points of a dataset as possible representatives and then calculate Responsibility and Availability iteratively based on the similarity matrix to adjust representatives. AP clustering algorithm has 3 indicators, namely Similarity, Responsibility and Availability, defined as follows:

Similarity: Assuming that $X = \{ x_1, x_2, ..., x_n \}$ is a dataset with n samples, the similarity of the dataset is:

$$s(i, j) = -\left\| x_i - x_j \right\|^2, i \neq j \tag{1}$$

In equation 1, $s(i,j) = 0$ represents that x_i and x_j has a greatest similarity, and $s(i,j) = -\infty$ represents that x_i and x_j belong to different categories. $s(i,i)$ represents preference parameter, $s(i,i)$ is greater, the probability that x_i is selected as an exemplar is higher. Preference parameter also indicates the number of exemplars, and the larger is the parameter, the more is exemplars. Another two indicators of the AP algorithm are:

Responsibility: $r(i,k)$ sent from x_i to candidate exemplar x_k, reflects the accumulated evidence for how well-suited x_k is to serve as the exemplar for x_i, taking into account other potential exemplars for x_i.

Availability: $a(i,k)$ send from candidate exemplar x_k to x_i, reflects the accumulated evidence for how appropriate it would be for x_i to choose point k as its exemplar, taking into account the support from other points that x_k should be an exemplar.

AP algorithm calculates the above two indicators iteratively based on the similarity matrix, as follows:

$$r(i,k) = \begin{cases} s(i,k) - \max\limits_{k' \neq k}\left[a(i,k') + s(i,k') \right], i \neq k \\ s(k,k) - \max\limits_{k' \neq k}\left[s(k',k) \right], i = k \end{cases} \tag{2}$$

$$a(i,k) = \begin{cases} \min\limits_{i \neq k}\left\{ 0, r(k,k) + \sum\limits_{i' \neq i, i' \neq k} \max\left[0, r(i',k) \right] \right\}, i \neq k \\ \sum\limits_{k' \neq k} \max\left[0, r(k',k) \right], i = k \end{cases} \tag{3}$$

In order to avoid the occurrence of shock, damping parameter $\lambda \in [0,1)$ is introduced. The updating of Responsibility and Availability are as follows:

$$r_{new}(i,k) = \lambda r_{old}(i,k) + (1-\lambda) r(i,k) \tag{4}$$

$$a_{new}(i,k) = \lambda a_{old}(i,k) + (1-\lambda) a(i,k) \tag{5}$$

In equation 4 and 5, the bigger is λ, the better is the effect of the elimination of shock, but the convergence rate of the algorithm will be slow.

3. Affinity Propagation Clustering Algorithm based on Spark Platform

Although AP clustering algorithm improves the clustering effect of complex data, but in large scale data, the algorithm still can not meet the requirements. Aiming at this problem, this paper proposes the Spark-AP clustering algorithm. The realization of algorithm is that a dataset is partitioned into several RDDs on a strategy and select the exemplars of each RDD. Then exemplars are merged and are used to next AP clustering algorithm, which forms a set of high-quality exemplars after convergence. For partitioning and merging strategies of data, this paper makes the following assumptions:

Assumption 1: The number of data points in the dataset is n, the times of iterative computing is T.

Assumption 2: The dataset is randomly partitioned into k RDDs, $k \in [2,n]$ and the size of each RDD data is nearly identical, namely, $m = n/k$.

Assumption 3: m data points are expected to get $\sqrt{\theta m}$ exemplars, $\theta \in [1/m, m]$. θ is associated with the preference parameter, when the preference parameter is large, θ is small, and the reverse is true.

The flow chart of Spark-AP clustering algorithm is shown in figure 1:

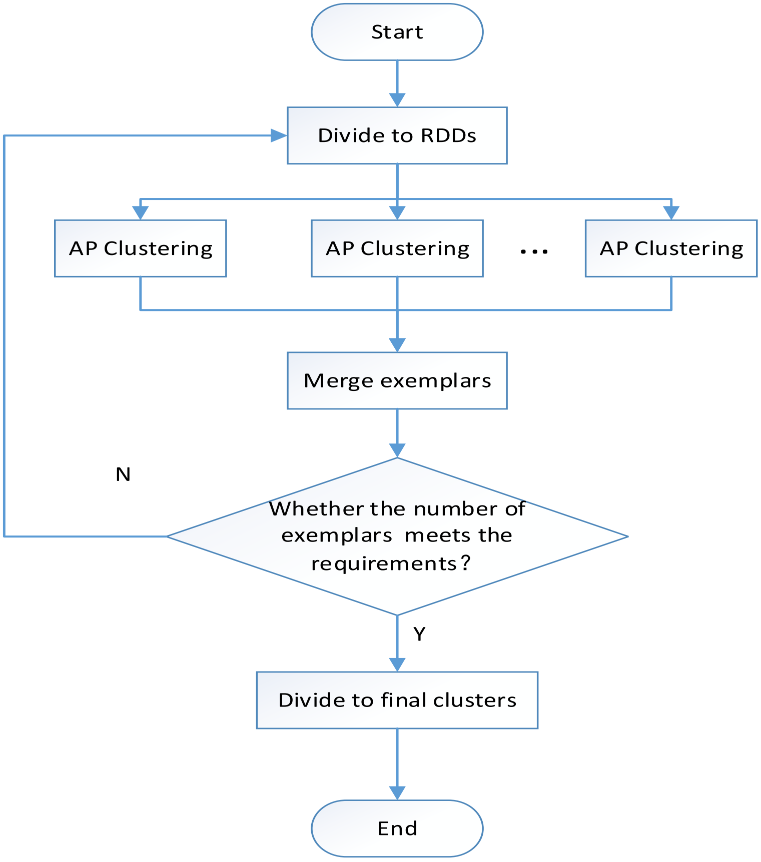

Fig. 1 The flow chart of Spark-AP clustering algorithm

4. Experimental Results and Analysis

The environment is composed of Spark cluster consists of four computers, 1 master host and 3 workers. The master host is mainly responsible for resource scheduling and work hosts are responsible for actual operation, each computer configuration for 2.4 GHz dual core processor, 2GB memory and 64 bit Ubuntu 12.04 operating system. And the experiment using manifold learning tools MANI[8] synthetic 3D Clusters to test the running efficiency of AP and Spark-AP algorithm, the AP algorithm preference parameter is the average value of similarity matrix, the damping parameter is 0.5, the number of iterations is 400, k is 8. Experimental results are shown in Figure 2.

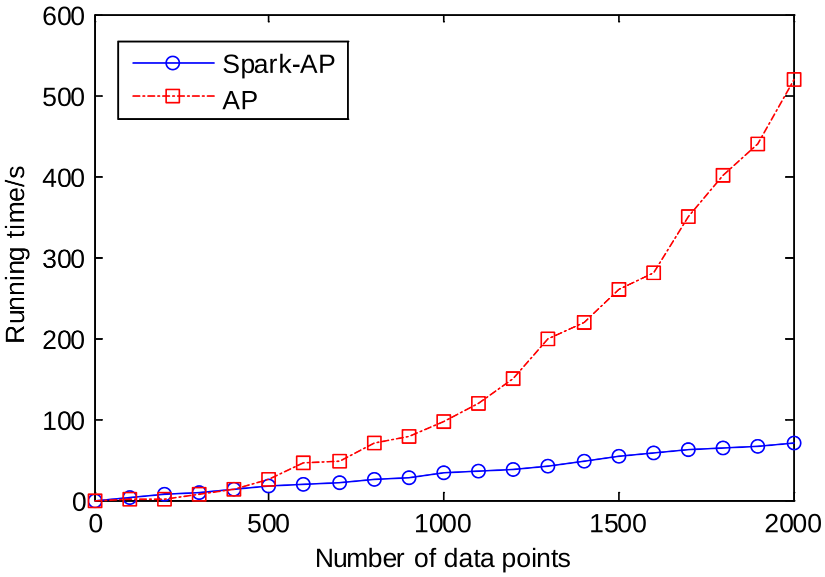

Fig. 2 The comparison of AP and Spark-AP running time

In the figure 2, when the data size is less than 500, the running time of Spark-SAP is slightly greater than AP, because the process of Spark-AP algorithm is more complex, such as the classification of datasets and the combination of exemplars, those processes also cost some time. When the data size increases, the running time of Spark-SAP is much less than AP. Because Spark-SAP divides a dataset into several RDDs and then parallelly cluster on Spark platform. On the whole, the efficiency of Spark-AP is better than other alignment algorithms.

5. Summary

In this paper, AP clustering algorithm is used in Spark, a dataset is divided into several RDDs and select the exemplars of each RDD. Then exemplars are merged and are used to next AP clustering algorithm, which forms a set of high-quality exemplars after convergence. Experimental results show that the efficiency of Spark-AP has significantly improved compared with AP clustering algorithm. Next, the clustering effect of AP and Spark-AP will be studied.

References

[1]. Lu W, Du C, Wei B, et al. Distributed Affinity Propagation Clustering Based on MapReduce[J]. Journal of Computer Research & Development, 2012, 49(8):1762-1772.

[2]. Frey B J, Delbert D. Clustering by passing messages between data points[J]. Science, 2007, 315(5814):972-6.

[3]. Zaharia M, Chowdhury M, Das T, et al. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing[C]// Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. USENIX Association, 2012:141-146.

[4]. Information on:http://spark.apache.org/ .

[5]. Sarazin T, Azzag H, Lebbah M. SOM Clustering Using Spark-MapReduce[C]// Parallel & Distributed Processing Symposium Workshops (IPDPSW), 2014 IEEE International. IEEE, 2015.

[6]. Ghesmoune M, Lebbah M, Azzag H. Micro-Batching Growing Neural Gas for Clustering Data Streams Using Spark Streaming [J]. Procedia Computer Science, 2015, 53(1):158-166.

[7]. Jin C, Liu R, Chen Z, et al. A Scalable Hierarchical Clustering Algorithm Using Spark[C]// IEEE International Conference on Big Data Computing Service and Applications. 2015:418-426.

[8]. Du B, Zhang L, Zhang D, et al. A manifold learning based feature extraction method for hyperspectral classification[C]// Information Science and Technology (ICIST), 2012 International Conference on. IEEE, 2012:491 - 494.