Single-Pass Clustering Algorithm Based on Storm

LI Fang, DAI Longlong, JIANG Zhiying and LI Shunzi

Beijing University of Chemical Technology, Beijing, 100029

lifang@mail.buct.edu.cn

Abstract. The dramatically increasing volume of data makes the computational complexity of traditional clustering algorithm rise rapidly accordingly, which leads to the longer time. So as to improve the efficiency of the stream data clustering, a distributed real-time clustering algorithm (S-Single-Pass) based on the classic Single-Pass [1] algorithm and Storm [2] computation framework was designed in this paper. By employing this kind of method in the Topic Detection and Tracking (TDT) [3], the real-time performance of topic detection arises effectively. The proposed method splits the clustering process into two parts: one part is to form clusters for the multi-thread parallel clustering, the other part is to merge the generated clusters in the previous process and update the global clusters. Through the experimental results, the conclusion can be drawn that the proposed method have the nearly same clustering accuracy as the traditional Single-Pass algorithm and the clustering accuracy remains steady, computing rate increases linearly when increasing the number of cluster machines and nodes (processing threads).

1. Introduction

With the development of the Internet, accessing information from the network has become a part of daily life. Therefore, the research of Topic Detection and Tracking (TDT) has attracted more and more attentions. TDT is a processing technology what is aiming at unknown information identification and known information tracking. TDT makes the topic occurred in special time and place extend to extension of topic with more relevance, which becomes an important technology for the decision makers to obtain necessary information.

It is well known that this is the era of big data[4], which is characterized by a large amount of data, complex structure, faster generated data, and more data types. Therefore, the traditional Single-Pass algorithm couldn’t handle such a large amount of data. Clustering and detecting of large amounts of data become increasingly difficult and timeliness is getting lower and lower. So in this paper, a Single-Pass clustering algorithm based on Storm framework is used to identify and track text data. Storm is based on Hadoop [5] distributed, fault-tolerant, real-time computing systems, and makes up a
Hadoop batch not meet real-time requirements. Mark and Li Lingjuan achieve the CluStream algorithm in Storm[6], improving the speed of the cluster, and the accuracy has not changed. Wang Mingkun implemented DBSCAN algorithm in Storm [7] and increased throughput. All of them are illustrated by Storm framework can be used in clustering. Therefore, this paper trying to achieve combine Single-Pass algorithm with the Storm framework and improved the algorithm. As Single-Pass algorithm for clustering data can be affected by the order of the results, this paper proposed a two-tier clustering algorithm, ensuring the accuracy of the algorithm. Proved by testing new algorithms to ensure the accuracy of cluster at the same time, which greatly improves the speed and improve timeliness.

2. Operating Mechanism

2.1 Development of Storm

With the development of big data, Hadoop, a software framework which can use on big data to reliable, efficient, and can telescopic of way for distributed processing, came into being. The Mapreduce[8] framework can improve the algorithm parallel degrees and operation speed. However, Hadoop can only process the offline data, and real time cannot be guaranteed. From 2011 the Storm open source, its high throughput, high efficiency greatly improved the processing of big data.

2.2 Mechanism of Storm

Storm cluster consists of a master node and many worker nodes. Master node runs the "Nimbus" daemon, also known as Nimbus nodes for distributing code, assigning tasks, and fault detection. Each node runs the "Supervisor" daemon which has become a Supervisor node for listening, starting and terminating the worker process. Between master nodes and worker nodes in the Storm, there is Zookeeper cluster. Supervisor node contains a lot of Worker nodes, specific tasks are performed by the Worker nodes. Storm framework is shown in the following example, 1:

![Figure 1. the framework of storm](image)

Topology is the logical units of Storm. Will be packaged into the calculation task after the topology is published, task topology is composed of many of the spout node and bolt node. Spout is the node of reading data, through spout node can read data from external data source (MySQL, and HBase) and to metal group of form sent to corresponding of bolt node, calculation and processing by bolt node, bolt node also can sent the results of their operation to next layer of bolt node, such on put task into multiple part, each part is parallel processing, speed up the whole task. Task topology after submission will run unless end by others, in line with the characteristics of stream data processing.
3. Distributed Real Time Clustering Algorithm For Stream Data

3.1 TF-IDF Algorithm

This paper uses data from Guizhou province Research Center of public opinion text data, stored in a cluster of HBase. Text based on vector space model (VSM) [9], expressed in the form, can have a text representation

\[ D = \{(t_1, w_1), (t_2, w_2), (t_3, w_3), \ldots, (t_n, w_n)\} \]

The \( t_1, t_2, t_3, \ldots t_n \) is feature for text content, And the \( w_1, w_2, w_3, w_n \) is corresponding feature weights, Then the weight calculated by TF-IDF algorithm.

Term frequency-inverse document frequency (TF-IDF) is one of the most fundamental techniques in IR which is a product of term frequency (TF) and inverse document frequency (IDF) where TF represents the importance of the term in a document and IDF represents the importance of term in entire corpus. Equation (1) (2) is as follows:

\[ tf_{i,j} = \frac{n_{i,j}}{\sum n_{i,j}} \quad (1) \]

\[ idf_j = \log \left( \frac{|d|}{|\{j : t_j \in d\}|} \right) \quad (2) \]

The title and content is usually divided into two parts, the title usually as the content of the summary expression, more important than content, and the weights for title should be increased, so this paper gives the weights a coefficient (coefficient > 1).

3.2 Computes the Degree of Similarity between Two Documents

After vectorization of text, we usually compute the similarity of the text and compare the distance with the threshold size for text clustering. Text vector will be sent Shuffle grouping from this bolt node to the next node. On the second layer of bolt node, text space distance between the texts which usually is to calculate the cosine of the angle between two text vectors. Equation (3) is as follows:

\[ \cos \theta = \frac{\sum_{i=1}^{n} (A_i \times B_i)}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3) \]

When the cosine values closer to 1, show the closer of the two texts, that is, the more likely of the two texts belong to the same cluster. Due to calculation text each other similar degrees of calculation volume too complex, leading to consumption of time too long, so in formed of each cluster in the select one and other text average similar degrees highest of text as a center text, after that the non-classified text calculate the similar degrees with various of center text, according to the calculation results to determine the non-classified text belongs to which cluster. If the distance between the non-classified text and all center text is larger than threshold T, the non-classified text will form a new cluster. On the contrary, the non-classified text will be merged into the minimum distance cluster, and the cluster will recalculate the Centre text.
A cluster formed on the second layer bolt node, and the cluster will be sent to the third layer bolt through Globlegrouping. When sending the cluster of second layer to the third layer bolt, the bolt node will conduct the Single-Pass cluster for the second time. Topology diagram as shown in Figure 2:

![Topology Diagram](image)

**Figure 2.** The topology of storm

### 3.3 Pseudocode

BEGIN

readData(input)  // Read data from the Hbase
sendToFirstBolt(input)  // Sent from the spout to the first bolt node
svm := nlpData(input)  // Data to quantify
sendToSecondBolt(svm)  // Quantitative data will be sent to the second bolt node
C := nearest miroCluster  // Find the nearest micro-clusters
maxSimilarityValue := similarityValue(p, C)
if maxSimilarityValue > threshold then
    insertMiroCluster(p)  // Similar to that of the most micro-clusters
else
    create new miroCluster C  // New micro-clusters
end if

sendToThirdBolt(C)  // The formation of micro-clusters is sent on to the third floor bolt node

T := nearest topic
maxSimilarityValue := similarityValue(C, T)
if C is new then  // Micro-cluster is first formed
    if maxSimilarityValue > threshold then
        insertTopic(C)
    else
        create new Topic T
    end if
else
    if maxSimilarityValue > threshold then
        if oldTopic(T) then  // If is the subject of one of the old topics
            updateTopic(T)
        else
            splitOldTopic(T)
            insertNewTopic(T)
        end if
    else
        splitOldTopic(T)
        create new Topic T
    end if
end if

END

### 4. Experiments

Testing the accuracy of text clustering, the paper uses NIST establishing evaluation criteria for TDT [10,11], a comprehensive performance evaluation, the formula is as follows:
\[ C_{Det} = C_{Miss} \times P_{Miss} \times P_{target} + C_{FA} \times P_{FA} \times P_{non-target} \]  

(4)

\( C_{Miss} \) as the coefficient of miss rate, \( C_{FA} \) as the coefficient of mistake rate, TDT evaluation values were 1.0 and 0.1 respectively; \( P_{Miss} \) as the miss rate; \( P_{FA} \) as the mistake rate; \( P_{target} \) and \( P_{non-target} \) as prior objective probabilities, \( P_{target} = 0.02, P_{non-target} = 0.98 \). In order to further evaluate the TDT system performance, it is often used the \( C_{Det} \) generalized, the normalized error recognition price \( (C_{Det})_{Norm} \), defined as follows:

\[ (C_{Det})_{Norm} = \frac{C_{Det}}{\min(C_{Miss} \times P_{target}, C_{FA} \times P_{non-target})} \]  

(5)

The data set used in public opinion by the Guizhou Research Center, it contains 40 million data. This paper selects 10,000 test data from the date set. The selected data is related to hundreds of events. Text clustering similar threshold \( T \) is set to 0.05.

In order to test the speed and precision of S-Single-Pass algorithm, the paper adopted conventional stand-alone Single-Pass cluster and S-Single-Pass algorithm. In the S-Single-Pass algorithm, the cluster contains a total of 16 nodes, including 1 spout node in the first layer of cluster, and the number of 1, 3, 5, 7 nodes in the second and third layer respectively, and 1 bolt node in the final layer. The threshold \( T_c \) of the number of clusters is set to 2000, the threshold \( T_t \) of the number of topic is set to 3000. When the number of topic is larger than the threshold \( T_t \), the cluster will remove the topic which is the least recently used. The value of Spout send queue is set to 10000. The result of calculating and the velocity contrast diagram shown in Figure 3:

![Speed comparison chart about two algorithm](image)

**Figure 3.** Speed comparison chart about two algorithm

In figure 3: When the number of nodes is 1, and the average of running speed is 1.66 times as much as the speed of stand-alone. On the whole, the efficiency increased by 66%. With the increase of the number of nodes in the second layer and the third layer, the program runs faster and faster. When the number of node in the second layer and third layer is up to 5 respectively, the total number of started node is up to 12, the program runs fastest, the efficiency increased by 151% overall. When the number of node in the first layer and second layer is up to 7 respectively, the total number of started node is up to 16, that is to say, all of the nodes have been started, the program runs slower and throughput is reduced. This is because when starting all of nodes, computer overload, information transfer between servers slow, then leading to the speed reduced.
Afterwards, compared to the accuracy of S-Single-Pass algorithm clustering and the accuracy of Single-Pass algorithm clustering, the paper conducted simulation operation. The algorithm different nodes mistake rate, miss rate, normalization error recognition costs as shown in table 1:

| Nodes     | $P_{miss}$ | $P_{FA}$ | $(C_{Det})_{Norm}$ |
|-----------|------------|----------|-------------------|
| 1         | 0.2491     | 0.0457   | 0.4730            |
| 3         | 0.2519     | 0.0491   | 0.4925            |
| 5         | 0.2489     | 0.0401   | 0.4454            |
| 7         | 0.2569     | 0.0414   | 0.4598            |
| stand-alone | 0.2464    | 0.0386   | 0.4358            |

As shown in table 1, the different node number of the S-Single-Pass algorithm and Single-Pass make a comparison through $P_{miss}$, $P_{FA}$, $(C_{Det})_{Norm}$ of three parts, it can be seen that different node number of the S-Single-Pass algorithm precision floating up and down is not large, basic flat, which can obtained based on Storm distributed framework of Single-Pass algorithm in guarantee precision of while, speed up the operation speed of algorithm.

5. Conclusion

Based on the poor efficiency of traditional Single-Pass algorithm employed in the massive data, this paper proposed a clustering algorithm based on distributed Single-Pass Storm to improve topic detection and tracking of timeliness. The structure of algorithm is use bilayer clustering, ensuring the accuracy of the clustering at the same time, and improving the speed of the cluster. Test results further validate the Single-Pass algorithm based on distributed framework for Strom to complete large-scale data processing in real time.

Acknowledgments

I would like to express my gratitude to the Beijing University of Chemical Technology for the financial support of this research. This study was supported by the Fundamental Research Funds for the Central Universities (JD1608) and the Fundamental Research Funds for the Central Universities (JD1502).

1. References

[1] Zhu, H., University, N., & Nanjing. (2011). Study on web topic online clustering approach based on Single-Pass algorithm. New Technology of Library & Information Service.

[2] Zhao, F., Lin, S., Gao, X., & Computers, S. O. (2016). The research and application of storm framework for large data. Microcomputer & Its Applications.

[3] Huang, S., Yang, Y., Li, H., & Sun, G. (2014, December). Topic detection from microblog based on text clustering and topic model analysis. In Services Computing Conference (APSCC), 2014 Asia-Pacific (pp. 88-92). IEEE.

[4] Cheng, X. Q., Jin, X. L., Wang, Y. Z., Guo, J. F., Zhang, T. Y., & Li, G. J. (2014). Survey on big data system and analytic technology. Journal of Software.

[5] Dong, X., Li, R., Zhou, W., Wang, C., Xue, Z., & Liao, D. (2013). Performance optimization and feature


enhancements of hadoop system. *Journal of Computer Research & Development.*

[6] Ke, M. A., & Lingjuan, L. I. (2016). Distributed real time stream data clustering algorithm and its implementation based on storm. *Journal of Nanjing University of Posts & Telecommunications.*

[7] WANG, M., YUAN, S., ZHU, Y., & WANG, D. (2014). Real-time clustering for massive data using Storm. *Journal of Computer Applications, 11*, 006.

[8] Li, J. J., Cui, J., Wang, D., Yan, L., & Huang, Y. S. (2011). Survey of mapreduce parallel programming model. *Tien Tzu Hsueh Pao/acta Electronica Sinica, 39*(11), 2635-2642.

[9] Guo, Q. (2008). The similarity computing of documents based on vsm. ,5186(11), 585-586.

[10] NIST. The 2002 Topic Detection and Tracking (TDT2002) Task Definition and Evaluation Plan [EB/OL]. (2002)[2008]. ftp: //jaguar. ncsl. nist. gov //tdt/td t2002/evalplans/ TDT02 . Eval. Plan. v1. 1. ps .

[11] Seo, Y. W., & Sycara, K. (2004). Text clustering for topic detection.