On-line motion pattern recognition algorithm for moving objects in Spark environment

Xinbing Fang, Nanping Mao, Yan Su, Hansheng Zhang, Qiang Wang, and Xiaodong Sun, Xudong Zhang, Lanhui Zeng

Marine Department of Satellite Tracing and Metering Jiangyin, China

Keywords: Spark; Stay points; classification

Abstract: Apache Spark is an emerging engine for big data processing, developed by AMPLab at the University of California, Berkeley. Its main feature is to provide a clustered Distributed memory abstract RDD (Resilient Distributed Dataset), an immutable set of records with partitions, and a programming model in Spark. RDD in Spark has two types of operations, transformations and actions. Transformation refers to an operation that results in a new RDD, action refers to getting a numerical result through operation. Based on Spark programming framework, an online motion pattern recognition algorithm for moving objects is proposed, which takes the characteristics of stopping points into consideration. The experimental results show that the algorithm improves the recognition accuracy with the increase of training trajectory, and improves the execution speed of the algorithm in the big data environment.

1. Introduction

The architecture of Apache Spark is shown in Figure 1, the main feature is to provide a clustered Distributed memory abstract RDD (Resilient Distributed Dataset), an immutable set of records with partitions, and a programming model in Spark. RDD in Spark has two types of operations, transformations and actions. Transformation refers to an operation that results in a new RDD, including map, flatMap, filter, etc. Action refers to getting a numerical result through operation, including collect, reduce, etc. Spark and Hadoop are both important projects in distributed computing architecture. Hadoop solves the problem of reliable storage and processing of big data, it has two important parts. The first is HDFS, a cluster composed of several computers, is used for distributed storage of files, while multiple copies are stored on different machines to ensure reliable storage of data. The second is MapReduce, Hadoop provide abstraction of Mapper and Reducer. On an unreliable cluster composed of multiple computers, large data sets are processed concurrently and distributed, and complex steps in the process of specific Map and reduce are encapsulated. Users only need to call the interface, which greatly facilitates users.

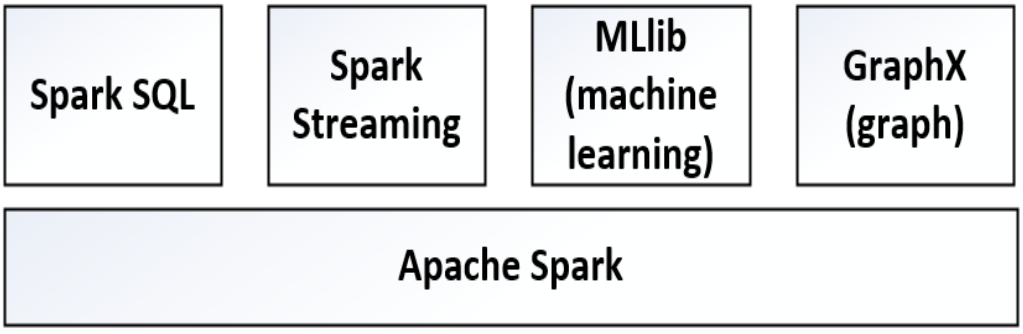

Figure 1. Spark ecosystem.

However, MapReduce has some disadvantages that make it more complicated to use. For example, it only provides two operations, Map and Reduce, which lacks expression power. The processing logic is hidden in the details of the code and there is no overall logic. Intermediate results are also

  DOI: 10.25236/cseem.2020.177

placed in the HDFS file system, and the reduceTask needs to wait for all maptasks to be completed before it can start. Furthermore, the high delay time, only Batch data processing, for interactive data processing, real-time data processing support is not enough.

But Spark solves some of the problems of MapReduce. For example, RDD abstract model is used to make the code more concise, provide many transformations and actions, such as join, groupBy, etc., which make the operation easier. The intermediate results are stored in memory first, and only when the memory is insufficient can they be stored in the local disk, so that the data can be read faster. Data is cached in memory to improve the speed of iterative calculation. Therefore, this chapter selects Spark as the programming framework for parallel computing. The parallel algorithm is mainly divided into two stages, namely the training stage and the test stage. There are four parallel parts in the training stage, including motion feature extraction, stop point cluster rule mining and path rule mining, stop point feature construction and classifier learning. The parallelism in the test stage is divided into three parts, which are the parallel extraction of motion features, the parallel construction of stopping point features and the parallel prediction of classifier.

2. Algorithm idea and description

The input data stream passes through the Spark Streaming receiver, the data is shred into DStream (the logical abstraction of the Streaming data in Spark Streaming), and the DStream is then processed in parallel by Spark Core's offline computing engine. All new tracks during online training are placed in a Dstream. The stopping points are extracted in parallel, and then the original cluster of stopping points is broadcast to each node. When Dstream itself joins the cluster of the stopping point after the update, the stopping point cluster pair is obtained. The original trajectory network graph and the mapping relationship between the old and new clusters are broadcast to each node, and the trajectory network graph is updated in parallel. The trajectory network diagram, the current rule set and the mapping relationship between the old and new clusters are broadcast to each node, and the stopping point cluster rules and path rules are updated in parallel. Finally, the updated rules are used to extract the motion feature and stopping point feature for each trajectory in parallel. After the trajectory feature is obtained, the Bagged decision tree classifier is updated in parallel.

3. Experiment and result analysis

In order to verify the effectiveness and operation efficiency of the online recognition algorithm of moving object motion mode considering the characteristics of stopping points and the parallel algorithm under Spark environment, the following experiments are carried out:

3.1 Algorithm effectiveness experiment

In order to verify the online learning ability of the online recognition method, the online recognition method was compared with the following offline methods: Offline algorithm: no online algorithm is used to update the cluster rules and path rules of stopping points, as well as the classifier. The test trajectory is directly built up with the rules obtained by the off-line algorithm, and the original classifier is used to predict the trajectory motion pattern. In addition, by changing the increment ratio, the accuracy of on-line recognition method and off-line recognition method is detected under different increment ratio. The accuracy of each method was evaluated using a 50% fold cross test. The identification accuracy of online and offline methods with different incremental proportions is shown in table 1.

Table 1. Comparison of recognition accuracy between offline algorithm and online algorithm with different increment ratio.

method / Incremental proportion	The offline algorithm	The online algorithm
30%	0.764	**0.835**
40%	0.759	**0.839**
50%	0.760	**0.840**
60%	0.752	**0.835**
70%	0.739	**0.838**

It can be found that the online algorithm is better than the offline algorithm (no longer updating the stopping point cluster rule, path rule and classifier) in the case of different increment ratio when the classification model is constantly updated with the increase of training track set, so as to achieve higher recognition accuracy. At the same time, the online algorithm's recognition accuracy changes stably under different increment ratio, while the offline algorithm fluctuates greatly.

The off-line algorithm is used to mine the number of training tracks of stopping point cluster rule and path rule under different increment ratio. The more important rules are mined, the easier it is to improve the recognition accuracy. In general, online moving object motion pattern recognition algorithm can obtain better recognition accuracy than offline algorithm in the case of increasing training track.

3.2 Operating efficiency of the algorithm

In order to verify the efficiency of the online recognition algorithm, this algorithm is compared with the simple online recognition algorithm. The simple online recognition algorithm means that when the new training track comes, the original training track is merged with the new training track, the stopping point cluster rule and the path rule are mined again, and the classifier is constructed. Experiments were carried out under different incremental proportions to test the time cost of online recognition algorithm and simple online recognition algorithm respectively. The final experimental results are shown in Figure 2.

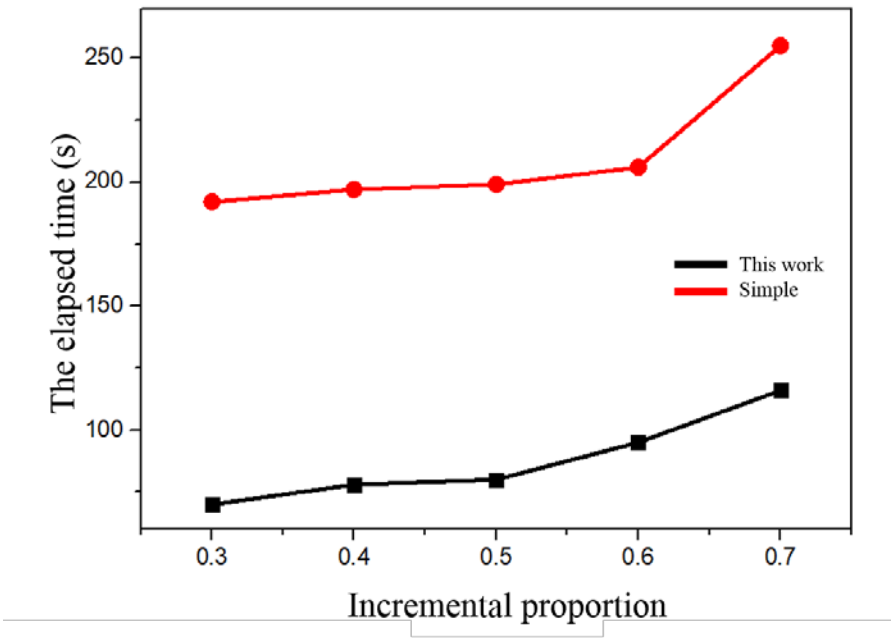

Figure 2. Comparison of running time under different increment ratio.

As can be seen from the figure, the running time required by a simple online recognition algorithm in different incremental proportions is greater than that of the online recognition algorithm, so the online recognition algorithm can update the stopping point cluster rules, path rules and classifiers used in the motion pattern recognition more quickly.

3.3 Parallel algorithm operation efficiency experiment

In order to verify the execution efficiency of online recognition algorithm of moving object motion pattern considering stopping point feature in Spark environment, this section only conducts experiments on the time required for online updating classifier stage. Experiments were carried out on a cluster of 18 working nodes using trajectory data sets of different specifications.

As can be seen from the figure, when the data set is small, the running speed of the algorithm on 18 nodes is not much different from that on a single node. This is because the parallel algorithm is used for data communication between nodes most of the time when the data volume is small. When the data is large, the algorithm runs much faster on 18 nodes than on a single node. Because most of the time is spent performing computational tasks, parallel algorithms are performed by multiple machines in parallel. In order to analyze the acceleration of parallel algorithm in cluster environment, 2, 6, 10, 14 and 18 computing nodes were used for experiments.

It is found from the figure that with the increase of the number of nodes, the running time of the algorithm decreases gradually, and the reduction is smaller and smaller. This is because the increase of the number of nodes leads to the increase of the resources involved in the calculation and the increase of the communication overhead between nodes.

4. Conclusion

When the training track with label comes continuously, the cluster rule and path rule of stopping point can be updated in time, as well as the classification model, so as to improve the scalability of the algorithm. Based on Spark programming framework, an online recognition algorithm of moving object motion pattern based on Spark is proposed. Experimental results show that the algorithm improves the recognition accuracy and the execution speed with the increase of training track.

Reference

[1] Brinkhoff T. A Framework for Generating Network-Based Moving Objects [J]. Geoinformatica, 2002, 6(2):153-180.

[2] Hemminki S, Nurmi P, Tarkoma S. Accelerometer-based Transportation Mode Detection on Smartphones[C]. ACM Conference on Embedded Networked Sensor Systems, Rome: Italy, 2013:1-14.

[3] Endo Y, Toda H, Nishida K, et al. Deep Feature Extraction from Trajectories for Transportation Mode Estimation[C]. Pacific-Asia Conference on Knowledge Discovery and Data Mining, Auckland: New Zealand, 2016: 54-66.

[4] Visvalingam M, Whyatt J D. The Douglas-Peucker Algorithm for Line Simplification: Re-evaluation through Visualization [J]. Computer Graphics Forum. 1990, 9(3): 213-228.

[5] Reddy S, Mun M, Burke J, et al. Using Mobile Phones to Determine Transportation Modes[J]. ACM Transactions on Sensor Networks, 2010, 6(2):662-701.

[6] Stenneth L, Wolfson O, Yu P S, et al. Transportation Mode Detection Using Mobile Phones and GIS Information. [C]. ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, Chicago: USA, 2011:54-63.

[7] Bolbol A, Cheng T, Tsapakis I, et al. Inferring Hybrid Transportation Modes from Sparse GPS Data Using a Moving Window SVM Classification[J]. Computers Environment & Urban Systems, 2012, 36(6):526-537.

[8] Mun M Y, Seo Y W. Everyday Mobility Context Classification Using Radio Beacons[C]. IEEE Consumer Communications and networking Conference. Las Vegas: USA, 2013:112-117.

[9] Zhu Y, Zheng Y, Zhang L, et al. Inferring Taxi Status Using GPS Trajectories [J]. ArXiv Preprint ArXiv: 2012:1205. 4378.

[10] Jahangiri A, Rakha H A. Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data [J]. IEEE Transactions on Intelligent Transportation Systems, 2015, 16(5): 2406-2417

[11] Sun Z, Ban X. Vehicle Classification Using GPS Data [J]. Transportation Research Part C Emerging Technologies, 2013, 37(3):102-117.

[12] Shafique M A, Hato E. A Comparison among Various Classification Algorithms for Travel Mode Detection Using Sensors' Data Collected by Smartphones[C]. International Conference on Computers in Urban Planning and Urban Management, Massachusetts: USA.2015:175-192.