A Scheme of Implementing PCA Alogrithm on Storm Platform

Shan Yan[1, a], Li Lingjuan[1, b *], Ji Yimu[1, c]

[1]School of Computer, Nanjing University of Posts and Telecommunications, Nanjing 21003, China

[a]13041103@njupt.edu.cn, [b]lilj@njupt.edu.cn, [c]jiym@njupt.edu.cn

Keywords: Data Stream; Dimensional Reduction; Storm; PCA; Distribution and Parallelization

Abstract. In order to sovle the problem of dimension disaster when mining the high-dimensional data in the data stream and the problem of poor real-time response and insufficient system throughput of dimensional reduction algorithms, a scheme of implementing PCA algorithm on Storm platform is designed. This scheme programs each branch of PCA algorithm by using Storm's own components, and each component forms the task entity through data flow communication. The scheme realizes the alogrithm distribution and parallelization by setting the threads number and the process number of task entity. Experimental results of running PCA algorithm on Storm and computer cluster according to the scheme show that the PCA algorithm on Storm platform can meet the requirement of real-time dimensional reduction of data stream.

Introduction

Data Stream is a continuous, unpredictable, sudden, rapid and time-varying stream [1]. One key issues of data flow management is how to effectively do the dimensional reduction and compression of the data stream with a limited storage resource, according to the characteristics of the data stream and using effective memory scanning method, and express data flow information in a compressed form [2]. As a classical linear dimension reduction algorithm, PCA is simple and has not parameter limits, and it has been widely used in data compression and feature extraction. However, doing dimensional reduction on data stream under the current stand-alone environment by PCA can not meet the real-time dimensional reduction requirement for data stream because the throughput will be small and the complexity will be high. Therefore we solve such problems by means of distributed computing model and the computer cluster.

Storm is an open source framework for distributed real-time computing [3], and it can efficiently handle large data streams.

In this paper, we design the scheme of implementing PCA algorithm on Storm platform, and configure a high-performance cluster environment to implement PCA algorithm on Storm and computer cluster according to the scheme. The results verify that the scheme is feasible and it has the ability for doing real-time dimensional reduction on data streams.

PCA Algorithm

The main dimension reduction algorithm is divided into linear dimension reduction algorithm and nonlinear dimensional reduction algorithm, linear dimension reduction algorithm mainly including PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), etc. PCA algorithm has perfect theory, simple concept, convenient calculation, and optimal linear reconstruction error; in addition it has no parameter limits and is widely used in data compression and feature extraction. The principle of PCA is converting the original component related random vector to the new component unrelated random vector by orthogonal transformation [4].

Collecting p-dimensional random vector $x=(x_1,x_2,\ldots,x_p)^T$ n samples $x_i=(x_{i1},x_{i2},\ldots,x_{ip})^T$ (i=1, 2,…, n, n>p) to construct the sample matrix X:

$$X=\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1p}\\ x_{21} & x_{22} & \cdots & x_{2p}\\ \vdots & \vdots & \ddots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{np}\end{bmatrix}$$

The steps of PCA are as follows:

(1) Standardized calculation

Standardize sample matrix, firstly obtain its averages and variance, and then obtain a standardized matrix through the averages and standard deviation.

$$z_{ij}=\frac{x_{ij}-\bar{x}_j}{s_j},\quad i=1,2,\cdots,n;\quad j=1,2,\cdots,p; \tag{1}$$

We might calculate averages and variance by: $\bar{x}_j=\frac{\sum_{i=1}^{n}x_{ij}}{n}$, $s_j^2=\frac{\sum_{i=1}^{n}(x_{ij}-\bar{x}_j)^2}{n-1}$.

(2) The correlation matrix calculation

To obtain the correlating coefficient matrix of standardized matrix Z

$$R=\begin{bmatrix}r_{11} & r_{12} & \cdots & r_{1p}\\ r_{21} & r_{22} & \cdots & r_{2p}\\ \vdots & \vdots & \ddots & \vdots\\ r_{n1} & r_{n2} & \cdots & r_{np}\end{bmatrix}$$

The correlation coefficient can be caculated as:

$$r_{ij}=\frac{\sum z_{ki}z_{kj}}{n-1},\quad i,j=1,2,\cdots,p \tag{2}$$

(3) The eigenvalues and eigenvectors calculation

To obtain the p non-negative eigenvalues of Characteristic equation ($|R-\lambda I_p|=0$) of R correlating coefficient, $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_p\geq 0$, the corresponding eigenvectors are $C^{(i)}=(C_1^{(i)},C_2^{(i)},\cdots,C_p^{(i)})$, i=1,2, ···,p, where

$$C^{(i)}C^{(j)}=\sum C_k^{(i)}\cdot C_k^{(j)}=\begin{cases}1 & (i=j)\\ 0 & (i\neq j)\end{cases} \tag{3}$$

(4) The principal component determination

Using contribution rate $\eta=\frac{\sum_{j=1}^{k}\lambda_j}{\sum_{j=1}^{p}\lambda_j}\geq 0.85$ to determine the value of k, $\eta\geqslant 85\%$ means the information utilization rate is above 85%, solving equations Rb=$\lambda_j b$ for each λ_j (j=1, 2, ···, k) to obtain unit feature vector $b_j^0=\frac{b_j}{\|b_j\|}$。

(5) Projection

Transforming Indicator variables into principal components U_{ij}:

$$U_{ij}=z_i^T b_j^0,\quad i=1,2,\cdots,n,\quad j=1,2,\cdots,k \tag{4}$$

U_i is called the ith principal component.

Storm Platform

Storm is a distributed real-time computing system which focuses on the data stream processing [5], it can integrate the various existing technologies, such as Kestrel, Kafka, Memcached, Hbase, Redis and so on. In Storm framework, the task was performed under the logic of topology object. Topology consists of Spout, Bolt and Streams. Both of Spout and Bolt are multi-threaded running on a cluster, message delivery is completed by StreamGrouping, as shown in Fig.1. Spout is Topology message producer, usually read data from an external data source (message queues, data files, network transmissions, etc.), and then send them to the Stream in the form of Tuple. Bolt encapsulats the processing logic, can perform filtering, aggregation, calculation, function operation after receiving the Tuple. The relationship between Spout and Bolt is a subscription, and can flexibly achieve orientation and shunt of data.

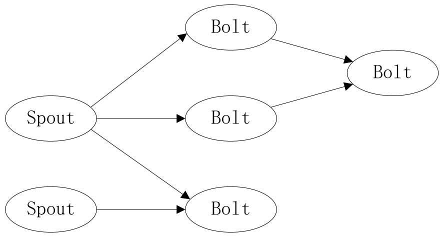

Figure 1. Topology Model of Storm

The Scheme of Implementing PCA Algorithm on Storm Platform

In this paper, we use the fully distributed environment of Storm and the characteristics of PCA algorithm to design a scheme of implementing PCA algorithm on Storm platform. The core of the scheme is to assign computing tasks to multiple nodes, and then summarize results. That is to achieve performance improvement through the task allocation and scheduling. The topology of implementing PCA on Storm platform is shown in Fig. 2.

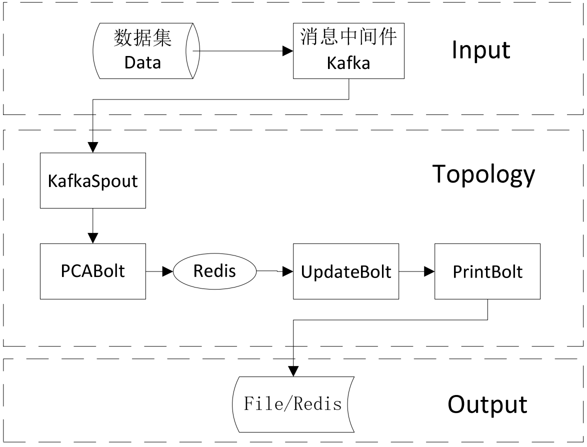

Figure 2. Topology of PCA on Storm Platform

The implementation of the algorithm can be divided into three modules: data access module, data processing module and data memory module.

Data Access Module.

Storm does not specify the source of the data, so the format of the data is varied, may be log file, database, message queue, or direct connection socket, etc., as long as the Spout can read those data through the appropriate interface. The data herein is message queue which was read from external data source through message middleware Kafka, and Storm as consumer of message queue takes the initiative to deal with data pulling from the message queue.

Data Processing Module.

Data processing module uses Storm's API and corresponding interfaces to implement the various functions of the algorithm. The main idea is to read the Kafka message queue through KafkaSpout, and then implement specific business logic of the algorithm through bolt. There are methods open(), nextTuple(), close() in Spout, and prepare(), execute(), cleanup() in Bolt. The open() and prepare() are used to declare execution object and output objects as well as access relevant resources; nextTuple() and execute() are used to perform computing tasks, and will be periodically called by framework; close() and cleanup() are used to release and clean-up resources to ensure the smooth implementation of follow-up task.

There are four components in Figure 2: KafkaSpout, PCABolt, UpdateBolt and PrintBolt.

KafkaSpout reads stream data from Kafka and sent its output to PCABolt.

PCABolt calculates principal component matrix of samples. The designed PCABolt in this paper does not immediately process every received data, rather than using 1 second as granularity, buffers the first received data within 1s; calculates the principal component matrix when cache data reached a certain amount, then writes the calculated principal component matrix to Redis to prevent the matrix loss when cluster is down.

UpdateBolt updates data stream. It reads principal component matrix from Redis, and updates each record according to principal component matrix, the updated data are sent to PrintBolt.

PrintBolt decides how to save dimensional reduction results in accordance with the actual demand. Tuples produced in each stage are passed to the next Bolt for processing according to user-defined.

Data Memory Module.

Storm has no storage mechanism. By PrintBolt component, the date results can be stored in the data file such as text file, or stored in a database such as Redis memory database. Data file is easy to replication and migration, and database is easy to query. In this paper, the data results are stored in Redis.

Experiments and Analysis

Experimental Environment and Data Set.

Hardware Environment: The cluster has 1 Nimbus node and 4 Supervisor nodes, and forms an internal network of 10.20.100.5 through high-speed switch.

Software Environment: JRE1.7.0_13, Zookeeper-3.4.6, Storm0.9.1, Kafka 2.8.1, redis-2.4.5, jedis-2.0.0.

Operating System: centos 6.4

Data set: Breast cancer data set [6] contains 102,294 instances; each instance has 117 attributes, beside a class attribute.

Experimental Results and Analysis.

In the experiments, we used the same data set do PCA dimensional reduction on a single machine and Storm clusters respectively, compared the dimensional reduction result of cluster with that of stand-alone in order to verify the advantages of the scheme.

We set the contribution rate η= 85%. Table 1 shows the distribution of the original data and the distribution after dimensional reduction, and m is the number of samples with the unit of ten thousands, n is the number of Dimensions. Fig. 3 shows the time consumption of PCA on a single machine and PCA on Storm clusters with data amount increasing.

Table 1 Dimensional Reduction Results

Original data set size (m×n)	1×117	2×117	3×117	4×117	5×117	6×117
Result data set size from a single machine	1×25	2×26	3×26	4×26	5×27	6×27
Result data set size from Storm cluster	1×24	2×24	3×24/25	4×24/25	5×24/25/23	6×24/25/23

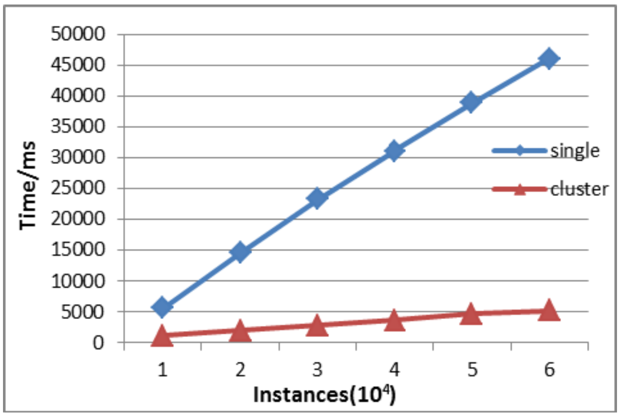

Figure 3. Time Consumption

It can be seen from Table 1 that each algorithm can effectively achieve data dimensional reduction, and the number of reduced dimension on Storm cluster is slightly less than that on a single machine. Figure 3 shows that doing dimensional reduction on Storm cluster has advantage in the aspect of time consumption, it needs less time. With the increasing of the data size, the advantage is more and more obvious.

Conclusions

In this paper, in order to meet the data stream processing needs, a scheme of implementing PCA algorithm on Storm platform is designed and implemented on the real-time distributed processing platform Storm. The experimental results of doing dimensional reduction on breast cancer data set show that Storm-based PCA has reduced dimensions and needs less time, that reflects the effectiveness of the scheme designed in this paper.

Acknowledgment

Results of this paper are supported by National Natural Science Foundation of China (Grant numbers 61302158,61571238) and ZTE Corporation Fund of University-industry Cooperation.

References

[1] Babcock B, Babu S, Datar M, et al. Models and issues in data stream systems//Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, 2002, pp. 1-16.

[2] Garofalakis M, Gehrke J, Rastogi R. Querying and mining data streams: you only get one look a tutorial//SIGMOD Conference. 2002, 2002, pp. 635.

Reference to a book:

[3] Anderson Q. Storm real-time processing cookbook. Packt Publishing Ltd, 2013.

[4] Hotelling H. Analysis of a complex of statistical variables into principal components. Journal of educational psychology, 1933, Vol.24, No.6.pp. 417.

Reference to a book:

[5] Leibiusky J, Eisbruch G, Simonassi D. Getting started with storm. O'Reilly Media, Inc., 2012.

[6] Information on http://www.sigkdd.org/kdd-cup-2008-breast-cancer