Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 6, June 2018, pp. 1360-1364
DOI: 10.23940/ijpe.18.06.p28.13601364

A Data Mining Algorithm based on Relevant Vector Machine of Cloud Simulation

Wuqi Gao[a,*],Gang Li[b], and Hui Liu[c]

[a]*School of Computer Science and Technology, Xi'an Technological University, Xi'an, 710021, China*
[b]*School of Economics and Management, Xi'an Technological University, Xi'an, 710021, China*
[c]*School of Electronics and Information Engineering, Xi'an Technological University, Xi'an, 710021, China*

Abstract

Regarding the problems of long time running and memory overflowing caused from the analysis of data mining algorithms for tactical communication network simulation data, using relevance vector machine (RVM), a data mining algorithm that is mainly used on the small sample of data mining with a good effect but a large amount of calculation that is based on an open source distributed storage and computing platform Hadoop, the author designs a kind of relevance vector machine data mining algorithm based on cloud computing. Based on the sum of the distribution of small sample data mining law in sequence, in some cases, the algorithm reflects the law of large sample data mining. Then, it carries on programming and empirical research, which supports the analysis of massive cloud simulation data.

Keywords: cloud computing; tactical communication network; data mining algorithm

(Submitted on March 29, 2018; Revised on April 12, 2018; Accepted on May 23, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

1. Introduction

With the enlargement of service resources for cloud simulation platform and huge information in the simulation process of tactical communication network, huge amount of data storage and analysis is one of the key problems to be solved [9]. Usually, the solution is to use distributed computing, parallel computing, cloud computing and other technologies. The regular data mining algorithm is RVM with good features. But, its problems are easy to produce memory overflow and long time-consuming [8]. Take the application of RVM to Matlab for example; when the data volume increased to more than 5000 in the process of data mining, the Matlab will warn error: "Out of memory" due to the large storage of memory [12,14].

The solution to data mining requires data to be identical distribution. When dealing with non-identical distribution, Mr. Li Gang provides a block algorithm based on the heuristic algorithm according to the principle of partition [3,11]. The data collection divides into data subsets, the answers of data subsets are calculated with the using of basic algorithm method and result vector are associated into the next data subset. This will compose a "block", then optimize "block" results of iteration starting point according to the principle of basic algorithm, and finally get the results of data collection for multiple recycling of above processing [7]. The same distribution is ideal; it is difficult to achieve in reality.

2. The designation for data mining algorithm based on the relevant vector machine of cloud simulation

This paper proposes a parallel distributed algorithm of relevance vector machine (SVM) using the cloud computing technology. The principle of this algorithm is based on the regular that the sum of data mining for small sample in sequence distribution will reflect the regular of large sample data mining in some cases. The combination of fitting results in every interval can reflect the whole interval's fitting results such as the curve fitting [2]. In tactical communication network, the network history data will firstly be divided according to different factors such as bandwidth, delay, throughput and so on. If divided by bandwidth, the sample data could be divided into three intervals: small bandwidth sample data, normal bandwidth sample data

* Corresponding author.
E-mail address: langfei@hrbust.edu.cn

and large bandwidth sample data. So, the results with respectively using data mining algorithm for the above divided sample data can reflect the regular of bandwidth in various circumstances.

The designation for this kind of algorithm is shown as follows:

- implementation of RVM data mining algorithm
- releasing RVM Matlab algorithm into the Java package
- establishing the Hadoop environment;
- incoming the sample data and sorting them in sequence
- setting up Map Reduce13 client platform, dividing sample data into sample pieces by the key value, using Reduce of data mining algorithm for information block to get the data mining5results of sample pieces
- collecting the data mining results of every sample piece and concluding total sample's data mining result

3. The research of relevant vector machine technology

Relevance vector machine (RVM) is a kind of sparse Bayesian model, which was proposed by Michael E. Tipping in 2000; it was built and trained under the Bayesian framework. The basic RVM model is built on regression problems and uses sigmoid function to solve classification problems under the same above framework. Compared with SVM, RVM has the following advantages6:

- can get the probability forecast
- can avoid the subjective error of parameters C and ε
- number of relevant vectors in the training is less than in SVM
- basis function is not needed to satisfy the Mercer condition [4,10], which has a greater range of options

For independent characteristics or target pairing data set $\{x_n, t_n\}_{n=1}^{N}$, in which x_n is the characteristic value (vector), $t_n$is the target value (a scalar). For regression problems, $t_n \in R$, assume RVM satisfies following formula:

$$t_n = y(x_n) + \varepsilon_n \tag{1}$$

$\varepsilon_n$is a zero mean gaussian noise; many random factors will make up ε_n: $(0, \sigma^2)$. This can be proved under the central limit theorem1:

$$p(t_n|x_n) = N(t_n|y(x_n), \sigma^2) \tag{2}$$

Which means that $t_n$is a kind of normal distribution with $y(x_n)$ for the mean and $\sigma^2$for the variance. In formula (1):

$$y(x) = \sum_{m=1}^{M} w_m \varphi_m(x) \tag{3}$$

In the above, $\varphi_m(x) = \varphi_m(x, x_m)$ is the basis function, and $w_m$is the weight value for $\varphi_m(x)$.
The topology prototype of RVM model can be shown as Fig. 1:

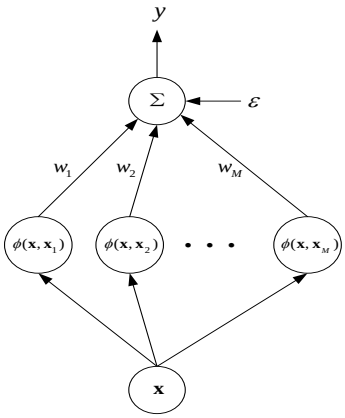

Figure 1.Topology prototype of RVM

Remark:

$$\begin{aligned} y &= (y(x_1) \ldots\ldots y(x_n))^T, \\ \varepsilon &= (\varepsilon_1 \ldots\ldots \varepsilon_N)^T, \\ w &= (w_1 \ldots\ldots w_M)^T, \\ \varphi_{nm} &= \varphi_m(x_n), \end{aligned} \tag{4}$$

then：

$$\begin{aligned} t &= y + \varepsilon \\ &= \varphi w + \varepsilon \end{aligned} \tag{5}$$

4. The Map Reduce process analysis for cloud computing

The operating model of MapReduce is shown as Fig. 2. One Map operation is a specified operation on a part of the original data, and each Map operation is operated against the different original data. So, every Map operation is independent from each other, which makes the operations fully parallelized. One Reduce operation is used to merge the partial results of the intermediate operations' results, i.e. Map operation' results, and each Reduce operation's results are not crossed from each other. All final results from Reduce operations will form a complete result set after a simple connection; thus, Reduce operations can also be conducted in a parallel environment.

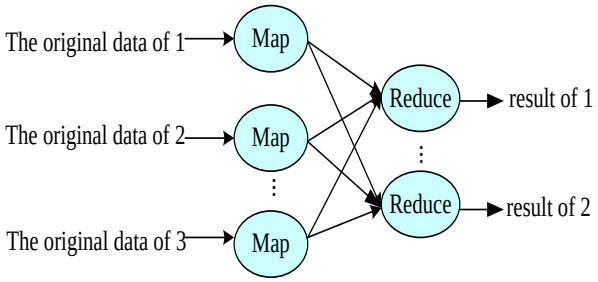

Figure 2. The logical model of MapReduce operation

5. The empirical for Cloud simulation data mining algorithm

In this section, based on the example of RVM relevance vector and data model, the idea of cloud computing for MapReduce to complete curve fitting is discussed and the corresponding results are analyzed. Detailed process is shown as the following: Generating the data of Y = x/sinx: Sinc： Equidistantly choosing100 samples for x in the scope of -10 to 10 under the formula t=noise+sin(x)/x, x in the scope of -10 to 10 , then the results will be with the characteristic of Uniform distribution in the scope of -10 to 10.

```
1.  Clear
2.  Clc
3.
4.  Rand('state',0);% Generate random seed
5.  m=-10;n=-8;
6.
7.      data(:,1)=sortrows(Unifrnd(m,n,5000,1)); % Generate 5000 samples of orderly
8.      data(:,2)=sin(data(:,1))./ (data(:,1)+eps);
9.      DATA(:,:,i)=data;
10.     save DATA % Save the sample data
11. end
```

6. Establishing the Map Reduce client:

The Reduce function:

```
1.  Public static class Reduce extends
2.  MapReduceBase implements Reducer<IntWritable, IntWritable ,Text, IntWritable >{

3.     Public void reduce (IntWritable Key,Interator<IntPair> values, OutputCollector<Text, IntWritable >) output
4.     Report reporter) throw IOException{
5.     myRVM=new RVM();//Instantiate the modle of RVM Jar
6.     RV_m=myRVM.operation(values); //Block the matching of curve
7.     Output.collect(Key, RV_M)  //Fit the output of parameter
8.     }
9.  }
```

The Map function:

```
1.  Public static class Map Extends
2.        Mapper <longWritable,Text,IntPair,IntWritable>{
3.        Private final IntPair intkey=new IntWritable(); //The variable values of key assignment
4.        Private final IntWritable intvalue=new IntPair();//The array of sample data
5.        Public void map(LongWritable key,Text value ,Context context)
6.        throws IOException, interruptedException {
7.        String line=value.toString();
8.        stringTokenizer tokenizer=new StringTokenizer(line);
9.        int left=0;
10.       int right=0;
11.       if(tokenizer.hasMoreTokes()){
12.             left=Integer.parseInt(tokenizer.nextToken());
13.             //divide into ten of partition
14.             if(-10<=left<-8)  intkey=1;
15.             if(-8<=left<-6)   intkey=2;
16.             if(-6<=left<-4)   intkey=3;
17.             if(-4<=left<-2)   intkey=4;
18.             if(-2<=left<0)   intkey=5;
19.             if(0<=left<2)    intkey=6;
20.             if(2<=left<4)    intkey=7;
21.             if(4<=left<6)    intkey=8;
22.             if(6<=left<8)    intkey=9;
23.             if(8<=left<10)   intkey=10;
24.             if (tokenizer.hasMoreTokenizer.nextToken());
25.             right=Integer.parseInt(tokenizer.nextToken());
26.             intvalue.set(left, right); //assign the value of the sample data and array
27.             context.write(intkey, intvalue); //Export the MAP
28.           }
29.        }
30.    }
```

7. Analysis on operation results

Order $h(x, c) = \exp(-\frac{\|x-c\|^2}{\sigma^2})$,i,e. h（.） is the basis function of Gauss, naming $\sigma = 0.05$, then there are results of associated vector, $c_1, c_2, \ldots, c_{11}$ with the calculation of relevance vector machine and weights, $w_1, w_2, \ldots, w_{11}$. The errors and running time are shown in Table 1.

Table 1. Operation results of relevance vector machine (SVM) in the first and second intervals

Item	the first interval[-10，-8]	the second interval [-8，-6]
associated vector	[-9.9803;-8.3737]	[-7.3814;-6.5458;-6.0233]
weights	[-0.21669;0.24663]	[-0.42085;1.8696;-1.5187]
errors	0.00036073	0.00015533
running time	0.437	0.016

8. Conclusions

The piecewise fitting curve is shown as Fig. 3, and the average absolute error is 2.429625 e-004. We can conclude that the fitting effect is relatively close to the real curve, which means this proposed method is verified to be effective in the use.

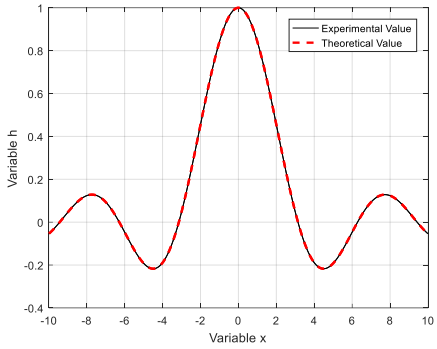

Figure 3. Piecewise fitting curve

Acknowledgements

This work was supported by the Shaanxi science and Technology Department's fund and Shaanxi Education Office's research unit fund. The authors would like to thank Pro. Gao for his proof-reading.

References

1. U. Aich, S.Banerjee, "Modeling of EDM responses by support vector machine regression with parameters selected by particle swarm optimization", *Applied Mathematical Modelling*, vol. 38, no(11-12), pp.2800-2818, 2014.
2. A. W. Blocker, F. V. Bonassi, S. L. Scott, et al., "Bayes and big data: The consensus Monte Carlo algorithm", *International Journal of Management Science and Engineering Management*, vol.11, no 2, pp. 78-88, 2016.
3. Y. Chaudhary, J. Joshi, S. Porwal, et al., "Data compression methodologies for lossless data and comparison between algorithms", *International Journal of Engineering Science and Innovative Technology (IJESIT)*, vol. 2, no 2, pp. 142-147, 2013.
4. S. Decherchi, P. Gastaldo, A. Leoncini, et al., "Efficient digital implementation of extreme learning machines for classification", *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no 8, pp. 496-500, 2013.
5. W. Ding, H. Lo, D. Wang, et al., "Crime hotspot mapping using the crime related factors—a spatial data mining approach", *Applied intelligence*, vol. 39, no. 4, pp.772-781, 2013.
6. B. L. Evans, J. Lin, M. Nassar, "Impulsive noise mitigation in powerline communications using sparse Bayesian learning", *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp.1172-1183, 2013.
7. M. Fazel, A. Jalali, S. Oymak, et al., "Simultaneously structured models with application to sparse and low-rank matrices", *IEEE Transactions on Information Theory*, vol.61, no.5, pp. 2886-2908, 2015.
8. L. Guo, C. Ruan, M. Wang, et al., "A cloud simulation based environment for multi-disciplinary collaborative simulation and optimization", in *International Conference on Computer Supported Cooperative Work in Design*, pp 445-450, 2017
9. H. Gupta, S. K. Ghosh, A. Vahid Dastjerdi, et al., "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments", *Software: Practice and Experience*, vol. 47, no.9, pp1275-1296, 2017.
10. J. Huang, J. W. Park, S. Shen, et al., "MATS: a Bayesian framework for flexible detection of differential alternative splicing from RNA-Seq data", *Nucleic Acids Research*, vol.40, no 8, pp. e61-e61, 2012.
11. F. Kang, J. Li, "Artificial bee colony algorithm optimized support vector regression for system reliability analysis of slopes", *Journal of Computing in Civil Engineering*, vol. 30, no. 3, 04015040, 2015
12. D. Liu, D. Y. Peng, Pan, X. Peng & J. Zhou, "Lithium-ion battery remaining useful life estimation with an optimized Relevance Vector Machine algorithm with incremental learning", *Measurement*, vol. 63, pp143-151, 2015
13. P. Wendell, R. S. Xin, M. Zaharia, et al. Apache spark: a unified engine for big data processing", Communications of the ACM, vol. 59, no.11, pp56-65, 2016
14. X. Wu, G. Q. Wu, X. Zhu, & W. Ding, "Data mining with big data", *IEEE transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97-107,2014.