An Integration of Extreme Learning Machine for Classification of Big Data

Guanwu Zhou[1,a], Yulong Zhao[2,b] and Wenju Xu[3,c]

[1]State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, 710049, China

[2]State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, 710049, China

[3]State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, 710049, China

[a]alainzhou@stu.xjtu.edu.cn, [b]zhaoyulong@mail.xjtu.edu.cn, [c]xuwenju@stu.xjtu.edu.cn

Keywords: data mining, extreme learning machine, back propagation neural networks, support vector machine, decision tree, big data

Abstract: Classification is an important task in data mining field. As the time of big data is coming, the traditional methods of classification cannot satisfy the requirements of real-time processing and storage for big data. This study firstly applies a machine learning technique called extreme learning machine (ELM) to classify for big data. Performances of ELM for big data are evaluated by using big data. The experimental results show that classification method based on ELM outperforms other methods based on BP neural networks and support vector machine (SVM). Secondly, the possibility of paralleling ELM based on MapReduce is analyzed and a paralleling ELM is designed.

Introduction

Classification techniques are supervised learning which learn from an existing categorized dataset what inputs of a specific category look like. As a result, the classification model constructed in the classification process is able to classify the unlabelled input dataset to the correct category. Therefore, classification is an important task in data mining; and is widely used in many areas such as image, finance, human capital, stock, marketing, manufacturing, health care, and etc. Many classification techniques have been proposed by researchers in machine learning, pattern recognition and statistics; and mainly focus on improving the performances of algorithms but not the data processing capability. There are several classical algorithms for classification such as decision tree (DT), Bayesian methods, Bayesian network, rule-based algorithms, neural network, support vector machine (SVM), association rule mining, k-nearest-neighbor, case-based reasoning, genetic algorithms, rough sets, and fuzzy logic [1]. However, as the time for big data is coming, some classification techniques are unsuitable to this condition or need to be redesigned with distributed parallel computing method, such as DT, NaiveBayes (NB), SVM, etc [2, 3]. It is important to know performances of classification techniques for big data, in this paper, four main classification techniques i.e. decision tree, BP neural network, and NB and SVM were used for experiments. From the learning speed and operability view, this study endeavored to explore ELM for big data compared with other algorithms. And now the machine learning based on Hadoop MapReduce is the research focus for big data, so this study simply analyzed ELM based on MapReduce.

This paper is organized as follows. The second section introduces the fundamental principle of proposed method based on ELM. The third section discusses experiment setup and results. Then, the forth section analyzes the possibility of paralleling ELM based on MapReduce and designs. Finally, the paper ends with conclusions and some related future works.

Methodology

In this section, the classification model for big data using extreme learning machine (ELM) is presented. Taking into account the randomness for the selection of weights and biases in hidden layer of ELM [4], an integration of ELM is used to determine the number of hidden nodes, and then the

average of related results of ELM is used to evaluate the performances of classification. For this method, firstly the useful dataset were extracted from the raw data by preprocessing dirty data. Then the most significant attributes are selected to be the inputs of the ELM and the output of the ELM is the category. Subsequently, before being divided into training and predicting datasets, the data composed of input/output pairs are transformed into numerical or string format so that the inputs fall into a specific range and the outputs satisfy the requirement of algorithms. Based on the network structure and configuration parameters obtained by training set, the predicted category for the predicting set can be calculated quickly by the established ELM. In experiments section, a figure about how each step is implemented is presented. In the following, a concise review of ELM [5] is given.

Extreme learning machine

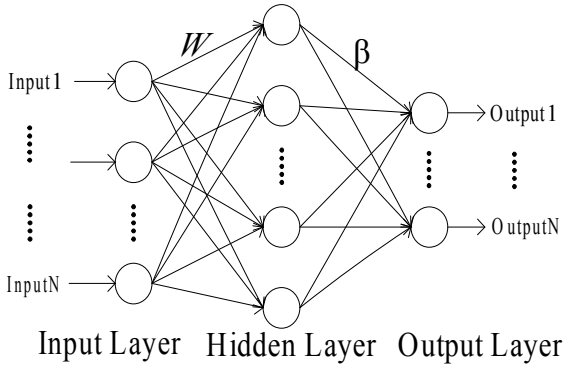


Fig 1.The structure of ELM algorithm

The structure of ELM is a single hidden-layer feedforward neural network, as shown in Fig. 1. The input weight matrix is randomly generated and uniformly distributed on the open interval (0, 1), and the output weight matrix is calculated with matrix operations.

The algorithm ELM can be summarized as:

Step1: Assume that there are a training setactivation function, and hidden node number .

Step 2: Randomly generate input weight and bias . There also exists parameters , such that

(1)

Where, is the weight vector connecting the *i*th hidden neuron and the output neurons, is the weight vector connecting the *i*th hidden neuron and the input neurons, is bias of ith hidden neuron.

Step 3: Transform the Eq. 1 as follows, and then calculate the hidden layer output matrix *H*.

(2)

Where, is the hidden-layer output matrix, , denotes the output of the *i*th hidden neuron with respect to

Step 4: Calculate the output weight through matrix operations.

(3)

Where, is the Moore-Penrose generalized inverse of the matrix *H* and .

Experimental steps of classification using ELM for big data

Likes other most classification algorithms, the classification process of ELM has two main phases: the first phase is the training process, at this process, the weight vector will be calculated through analyzing the training set based on the ELM algorithm. The second phase is classification process where the predicting set is used to estimate the performance of the classification model. If the performance is considered acceptable, the classification model can be applied to the classification of new data. And the specific steps of experiments are shown in Fig. 2.

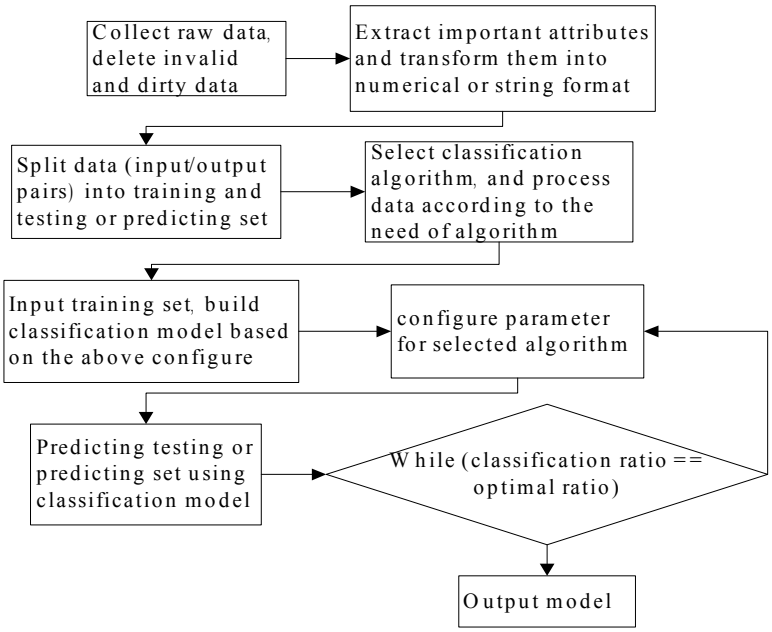


Fig 2.The flowchart of experiments for big data using ELM algorithms

Experiments

In order to test the validity and performances of ELM algorithm for big data, in this section some experimental results are presented. The batch steepest descent back propagation algorithm with an adaptive learning rate (BPGDA), and the gradient descent momentum and adaptive learning ratio (BPGDX), and Levenberg Marquardt (BPLM) are three typical back propagation algorithms [6,7,8]. DT, NB and SVM also are typical classification algorithms. As a comparison, the experimental results obtained by the DT, NB, SVM, BPGDA, BPGDX and BPLM algorithms on the same dataset were given. All experiments have been carried out in MATLAB 7.0 and MATLAB R2011b running in Windows XP with a AMD Athlon II X2 240 CPU (2.81 GHZ) and 2G memory.

Test data set is adult data set selected from the UCI [9]. Adult data set has 14 attributes, 48,842 samples. The samples were transformed into numerical matrix 45222×15 through deleting invalid input/output pairs. The data set is randomly divided into 10%~90% for the training set in the experiment and 90%~10% for the predicting set. The performances such as accuracy, runtime of the algorithms are analyzed based on the different number of training set and predicting set. In order to obtain an optimal structure of ELM for higher prediction accuracy and shorter computing time, as shown in Fig. 3, an integration method is proposed in this paper as selection criteria of the number of hidden nodes for ELM.

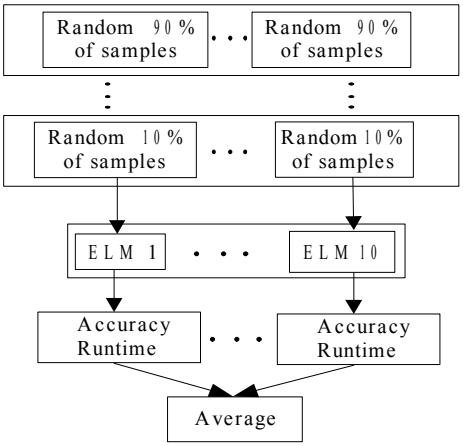
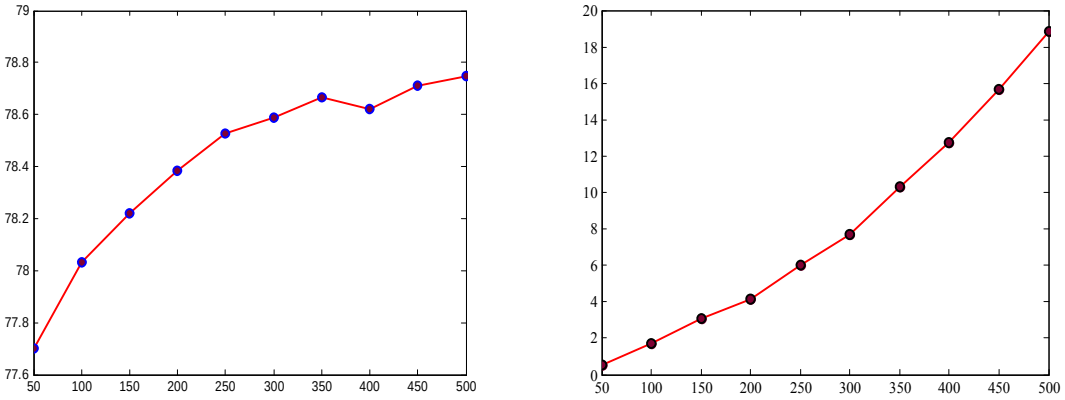


Fig 3. The selected scheme of the number of hidden nodes for ELM

As shown in Fig. 4 and Tabe 1, the number of hidden nodes is determined as 50 for ELM according to the average of predicting accuracy in predicting and runtime in training. The figures were rounded off to two after the decimal point in Table 1.

Table 1
The predicting accuracy and training CPU time of ELM

Hidden nodes	Predicting accuracy (%)	CPU time (s)
50	77.70%	0.53
100	78.03%	1.72
150	78.22%	3.05
200	78.38%	4.12
250	78.53%	6.00
300	78.59%	7.71
350	78.66%	10.34
400	78.62%	12.78
450	78.71%	15.64
500	78.75%	18.83



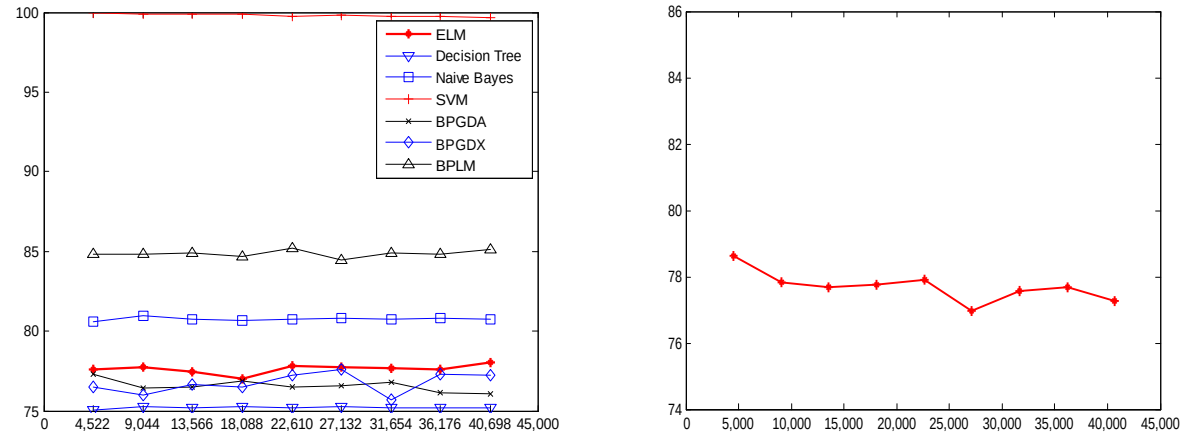
(a)The average of predicting accuracy (b)The average of training time

Fig. 4. The performances of ELM with different hidden nodes for big data

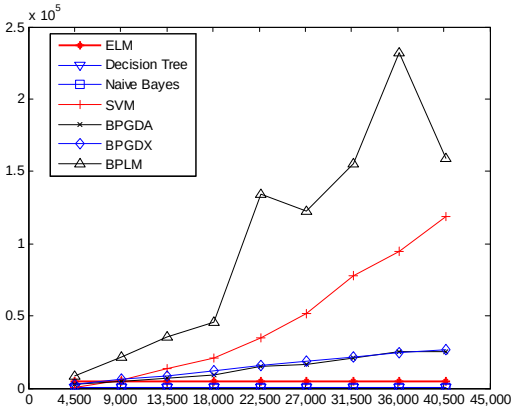
The experimental comparisons for BPGDA, BPGDX, BPLM, ELM, DT, NB and SVM are shown in Fig. 5. And the configured parameters for algorithms are shown in Table 2.

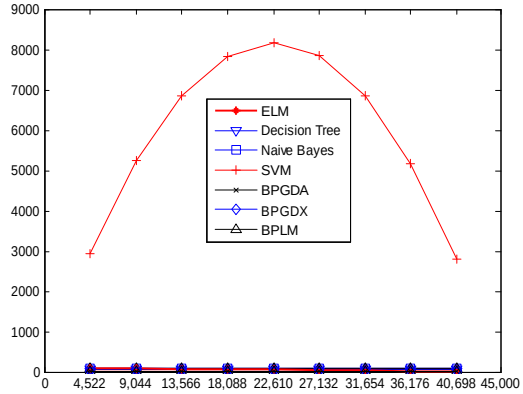
Table 2
The configured parameters

Algorithms	Configured parameters
ELM	nodes : 50; activation func : sig
SVM	'-s 0 -t 2 -c 5.0 -g 60'
BPNN	nodes : 10; activation func : tansig
DT/NB	Default



(a)The comparison of training accuracy (b)The comparison of predicting accuracy





(c) The comparison of CPU time in training (d) The comparison of CPU time in predicting

Fig. 5. The performances of different algorithms for big data

The performances of training time and predicting accuracy are the most important factors for evaluating algorithms. The averages of training time and predicting accuracy with different proportions of training and predicting set for algorithms were shown in Table 3.

Table 3
The average of training time and predicting accuracy for algorithms

Algorithm	Training time	Predicting accuracy
	Average (s)	Average (%)
ELM	0.53385	77.7%
DT	2.88941	75.2%
NB	0.12482	80.8%
SVM	464.296875	75.1%
BPGDA	140.79375	76.1%
BPGDX	152.06875	76.6%
BPLM	1015.49479	84.6%

From Table 2 and Fig. 5, NB was the best classification algorithm for adult data set according to the two important factors; its predicting accuracy and CPU time were 80.8% and 0.12482, respectively. Although BPLM had the highest predicting accuracy, it took more time (1015.49479s, the longest time in 7 algorithms) in training phase, and it was not suited to classification for big data if did not use paralleling algorithm. In training, the training accuracy of SVM was best, reached 99.81%, but its predicting accuracy just was 75.1% and the average of CPU time in predicting in Fig. 5. (d) was highest and the average of CPU time in training in Fig. 5. (c) was highest than other algorithms except BPLM. Compared with these results, ELM achieved the higher predicting accuracy in a short computation time than DT, SVM, BPGDX, BPGDA and BPLM.

Analyze and design paralleling ELM based on MapReduce

Hadoop MapReduce is a YARN-based system for parallel processing of large data sets [10]. DT, Bayesian, SVM, back propagation neural network and some other machine learning algorithms have been implemented based on Haddop MapReduce. The key point of implementing ELM is to calculate matrix inversion. And there are some methods including orthogonal projection, orthogonalization method, iterative method, and singular value decomposition (SVD) to calculate matrix inversion [11]. And SVD has been implemented based on Mahout [3]. So ELM also can be implemented on Hadoop MapReduce by above methods. The brief process of design is shown below.

1) Split training set into blocks.
2) Set structural parameter of ELM, and calculate the matrix H.
3) Calculate the $H^T H$ based on Hadoop MapReduce [12].
4) Calculate the inverse of $H^T H$ with SVD based on Hadoop MapReduce.
5) Calculate based on Hadoop MapReduce.
6) Calculate predicting output of predicting set with ELM.

Conclusions and future work

In this paper, an integration of ELM was proposed for classification of big data. Experimental results have shown that the integration of ELM was useful and had better predicting accuracy, simpler structure, and faster learning speed than some other classification algorithms such as DT, SVM, BPGDX and BPGDA on adult data set; and its predicting accuracy, the number of hidden nodes and CPU time were 77.7%, 50 and 0.53385, respectively . Compared with SVM, BPNN etc., ELM do not need complex configure, just set the number of hidden nodes. Operating simplicity is very important for testing algorithm for big data combining with faster learning speed. Therefore ELM is suitable to the classification of big data, and paralleling ELM based on Hadoop MapReduce will be more useful for big data.

For future work, the paralleling ELM is going to be implemented based on Hadoop MapReduce, and be tested by experiments on the Hadoop platform.

References

[1] H. Jantan, A.R. Hamdan, Z.A. Othman, Classification and Prediction of Academic Talent Using Data Mining Techniques, International Journal of Technology Diffusion, vol 1, pp. 29-41, 2010.

[2] Q. Lu, X.H. Cheng, "The Research of Decision Tree Mining Based on Hadoop," in 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery. Sichuan, China, 2012, pp. 798-801.

[3] http://mahout.apache.org/.

[4] J. Kittler, M. Hatef, P.W. Duin, J. Matas, On combining classifiers, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, pp. 226-239, Mar.1998.

[5]G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing, vol. 70, pp. 489-501, Dec. 2006.

[6] D.S. Huang, Systematic Theory of Neural Networks for Pattern Recognition, Publishing House of Electronic Industry of China, Beijing, 1996.

[7] K. Levenberg, A method for the solution of certain problems in least squares, Quarterly of Applied Mathematics, vol. 5, pp. 164-168, 1944.

[8] D. Marquardt, An algorithm for least-squares estimation of nonlinear parameters, Journal of the society for applied mathematics, vol. 11, pp. 431-441, 1963.

[9] http://archive.ics.uci.edu/ml/datasets/Adult.

[10] http://hadoop.apache.org/

[11] J.M. Ortega, Matrix Theory, Plenum Press, New York, 1987.

[12]S.Seo,E.J,Yoon,J.Kim,S.Jin,J.S.Kim,"HAMA: An Efficient Matrix Computation with the Map Reduce Framework," in Proceedings of the 2010 IEEE 2nd International Conference on Cloud Computing Technology and Science. Indianapolis, USA, 2010, pp. 721-726.