2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Data Intensive Parallel Feature Selection Method Study

Zhanquan Sun
Shandong Provincial Key Laboratory of Computer Network
Shandong Computer Science Center
Jinan, Shandong, 250014
sunzhq@sdas.org

Zhao Li
School of Software Engineering
Beijing Jiaotong University
Beijing, 100044, China
liz@sdas.org

***Abstract*—Feature selection is an important research topic in machine learning and pattern recognition. It is effective in reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility. With the development of computer science, data deluge occurs in many application fields. Classical feature selection method is out of work in processing large-scale dataset because of expensive computational cost. This paper mainly concentrates on the study of data intensive parallel feature selection method. The parallel feature selection method is based on MapReduce program model. In each map node, a novel method is used to calculate the mutual information and combinatory contribution degree is used to determine the number of selected features. In each epoch, selected features of all map nodes are collected to a reduce node and from which a feature is selected through synthesiation. The parallel feature selection method is scalable. The efficiency of the method is illustrated through an example analysis.**

Keywords—Feature selection; MapReduce; mutual information; contribution degree

I. INTRODUCTION

In recent years, data has become increasingly larger in both number of instances and number of features in many applications such as genome projects, text categorization, image retrieval and customer relationship management and so on [1-2]. It may cause serious problems to many machine learning algorithms with respect to scalability and learning performance. How to select the most informative variable combination is a crucial problem. Feature selection is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion [3]. It becomes very necessary for machine learning tasks when facing high dimensional data nowadays.

Lots of work has been done on feature selection and many efficient feature selection methods have been developed. Those methods can be divided into different categories according to different search strategies or evaluation criteria. Feature selection can be divided into three categories according to search strategies, i.e. global optimization, random search and heuristic search [4]. Some evolutionary algorithms are used to improve the searching speed, such as GA, ant colony algorithm and so on [5-6]. The computation cost of these feature selection methods is expensive. It is not suitable to deal with large-scale dataset. Many feature selection methods based on evaluation criteria have been proposed and applied to many practical areas. Correlation and distance measures are the basis of feature selection. Different measures have been adopted, such as correlation coefficient, logistic regression, kernel distance, PCA, mutual information and so on [7-10]. Mutual information is a significant measure of feature selection because that it can measure arbitrary statistical correlations between variables. Currently, most feature selection algorithms are designed and implemented for a centralized computing architecture. With the development of electronic and computer technology, the quantity of electronic data increases in exponential growth [11]. The usability of classic feature selection methods will be decreased when the data size is very large.

Feature selection method based on parallel computation will be the mainly choice for dealing with large-scale data. Parallel computing is implemented using different parallelization techniques such as threads, MPI, MapReduce, and mash-up or workflow technologies [12]. MapReduce is taken as the most efficient programming architecture to deal with data intensive problems. Some MapReduce models were developed. Google is taken as the first one to apply MapReduce to large-scale information retrieval. Hadoop is the most popular open source MapReduce software. For improving the computation efficiency, iterative MapReduce model, Twister, was proposed by Indiana University professor Fox [13-14]. Some parallel feature selection methods based on MapReduce have been studied. Singh and Kubica proposed a parallel logistic regression method based on MapReduce model [15]. Feature selection based on mutual information is parallelized based on MapReduce model in reference [16], but the computation cost of the parallel method is expensive and the number of selected feature variables can't be determined objectively. Based on previous work, this paper proposes a novel combinatory mutual information computation method. It can improve the computation speed markedly in case that the number of feature and feature values is very big. Combinatory contribution degree is proposed to determine the number of selected feature variables. The selected features are input to a SVM model to verify the efficiency through comparing classification correct rate.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

The following of the paper is organized as follows. Basic knowledge of mutual information is introduced briefly in section 2. A novel mutual information calculation method are presented in section 3. The iterative MapReduce model, Twister, is introduced in part 4. Parallel feature selection method based on MapReduce is proposed in section 5. Two practical examples are analyzed with the proposed model in section 6. At last some conclusions are summarized.

II. MUTUAL INFORMATION BASED ON SHANNON ENTROPY

Mutual information can measure any kind of statistical dependence between variables. It has been widely applied to pattern recognition area. Probability is the basis of entropy. Many kinds of definition of entropy had been proposed. The commonly used one is Shannon entropy. Mutual information based on Shannon entropy is introduced.

Feature variables are denoted by vector $\boldsymbol{X}=(X_1,X_2,\cdots,X_i,\cdots,X_m)^T$, where $X_i=(x_{ij})$, $i=1,2,\cdots,m, j=1,2,\cdots,q$ denotes the ith feature variable with q difference values. Class variable is denoted by Y, $Y=(y_i), i=1,2,\cdots,k$. It means that all features are projected to k different classes. In this paper, feature variables and class variable are supposed to be discrete. p_{X_i} denotes the probability distribution of feature variable X_i, p_Y denotes the probability distribution of class variable Y, and p_{X_iY} denotes the joint probability distribution of X_i and Y. All probability distributions are calculated through sample statistics. The Shannon entropy H of feature variable X_i can be described as

$$H(X_i)=-\sum_{j=1}^{q} p_{x_{ij}} \log p_{x_{ij}} \tag{1}$$

Shannon entropy of class variable Y can be described as

$$H(Y)=-\sum_{i=1}^{k} p_{y_i} \log p_{y_i} \tag{2}$$

Joint entropy between feature variables and class variable is

$$H(X_i,Y)=-\sum_{j=1}^{q}\sum_{l=1}^{k} p_{x_{ij}y_l} \log p_{x_{ij}y_l} \tag{3}$$

where X_i can be substituted by subset of feature vector S, i.e. the joint entropy can be generalized to p variables.

Mutual information between feature variable and class variable based on Shannon entropy is defined as

$$\begin{aligned} I(X_i,Y)&=H(X_i)+H(Y)-H(X_i,Y)\\ &=\sum_{j=1}^{q}\sum_{l=1}^{k} p_{x_{ij},y_l} \log \frac{p_{x_{ij}y_l}}{p_{x_{ij}} q_{y_l}} \end{aligned} \tag{4}$$

III. COMBINATORY MUTUAL INFORMATION CALCULATION AND CONTRIBUTION DEGREE

A. Calculation of Total Combinatory Mutual Information

Let $\boldsymbol{S}=(\boldsymbol{X},Y)$ denote n-sample set, where $\boldsymbol{X}=\{X_1,X_2,\cdots,X_n\}$ denote feature sample and each feature sample is a m-variable vector i.e. $X_i=(X_{i1},X_{i2},\cdots,X_{im})$, $i=1,2,\cdots,n$. $Y=\{Y_1,Y_2,\cdots,Y_n\}$ denotes class variable set. $B=\{B_1,B_2,\cdots,B_n\}$ denotes the frequency set with the same feature vector values. $\boldsymbol{D}=\{D_{ij}\}$, $i=1,2,\cdots,n; j=1,2,\cdots,k$ denotes the frequency set with the same feature vector and class values. $E=\{E_1,E_2,\cdots,E_k\}$ denotes the frequency set with same class value. The algorithm can be realized as follows.

1) Let sample set S be known. Set matrix B all to 1 and set matrix D and E all to 0.

2) The frequency of each data set is calculated as follows.

for $i=1,2,\cdots,n-1$
  $D_{il}=D_{il}+1$ and $E_l=E_l+1$, $l=1,2,\cdots,k$, If $Y_i=y_l$
end
for $i=1,2,\cdots,n-1$
  for $j=i+1,i+2,\cdots,n$
    $B_i=B_i+1$, $B_j=0$ If $X_i=X_j$ and $B_i\neq 0$
    $D_{il}=D_{il}+1$, $D_{jl}=0$ $l=1,2,\cdots,k$ If $X_i=X_j$, $Y_i=Y_j=y_l$ and $D_{il}\neq 0$
  end
end

3) Calculate the combinatory mutual information

$$I=\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{D_{ij}}{n}\log\left(\frac{D_{ij}/n}{(B_i/n)(E_j/n)}\right) \tag{5}$$

B. Contribution Degree

Similar to the definition of contribution degree in Principle Component Analysis method, contribution degree based on mutual information is defined as follows.

Definition: Let $I(X_i;Y)$ denote mutual information between each feature variable and class variable and $I(\boldsymbol{X};Y)$ be total mutual information. Contribution degree based on mutual information of each feature variable is

$$r_i=I(X_i;Y)/I(\boldsymbol{X};Y) \tag{6}$$

After feature selection, cumulative contribution degree between selected feature subset $\boldsymbol{S}$ and class variable is

$$r_{\boldsymbol{s}}=I(\boldsymbol{S},Y)/I(\boldsymbol{X},Y) \tag{7}$$

Based on the cumulative contribution degree value, we can determine the number of selected features objectively.

IV. FEATURE SELECTION BASED ON MUTUAL INFORMATION

Feature selection is usually used to select the most informative feature combination with least information loss for

classification problems. Here we generalized it to regression problems. As we all known, it is difficult to build classification model when feature variables are too much. If we can select the most informative variables for classification, it will save lots of computation cost and reduce the effect of noise. The information between class variable and feature variables is measured with mutual information. Feature selection of classification is to select the feature variable combination who has the largest mutual information value with class variable. The feature selection of classification based on mutual information metric can be formulized as follows:

Step 1: Set $\boldsymbol{X} \leftarrow$ "initial set of n independent variables;" $\boldsymbol{S} \leftarrow$ "empty set". Y is the class variable. Prescribe the number r of independent variables to be selected.

Step 2: Compute the mutual information $I(X_i;Y)$ between each feature variable $X_i \in \boldsymbol{X}$ $i=1,2,\cdots,n$ and class variable Y.

Step 3: Find the feature variable X_i that maximizes $I(X_i;Y)$; set $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{X_i\}$; set $\boldsymbol{S} \leftarrow \{X_i\}$.

Step 4: For all couples of variables $(X_i, \boldsymbol{S})$ with $X_i \in \boldsymbol{X}$, compute $I(X_i, \boldsymbol{S}; Y)$, choose feature variable X_i as the one that maximizes $I(X_i, \boldsymbol{S}; Y)$; set $\boldsymbol{X} \leftarrow \boldsymbol{X} \setminus \{X_i\}$; set $\boldsymbol{S} \leftarrow \boldsymbol{S} \bigcup \{X_i\}$. Repeat the process until r feature variables are selected;

Step 5: Output the set $\boldsymbol{S}$ containing the selected feature variables.

V. MapReduce Model Based on Twister

Many data mining algorithms are simple iterative structures. Most of them can be found in the domains such as data clustering, dimension reduction, link analysis, machine learning, and computer vision. These algorithms can be implemented with iterative MapReduce computation. Professor Fox developed the first iterative MapReduce computation model Twister. Twister's programming model can be described as in figure 1.

MapReduce jobs are controlled by the client program. During configuration, the client assigns MapReduce methods to the job, prepares KeyValue pairs and prepares static data for MapReduce tasks through the partition file if required. Between iterations, the client receives results collected by the Combination method, and, when the job is done, exits gracefully.

Map daemons operate on computational nodes, loading the Map classes and starting them as Map workers. During initialization, Map workers load static data from the local disk according to records in the partition file and cache the data into memory. Most computation tasks defined by the users are executed in the Map workers. Twister uses static scheduling for workers in order to take advantage of the local data cache.

Reduce daemons operate on computational nodes. The number of reducers is prescribed in client configuration step. The reduce jobs depend on the computation results of Map jobs. The communication between daemons is through messages.

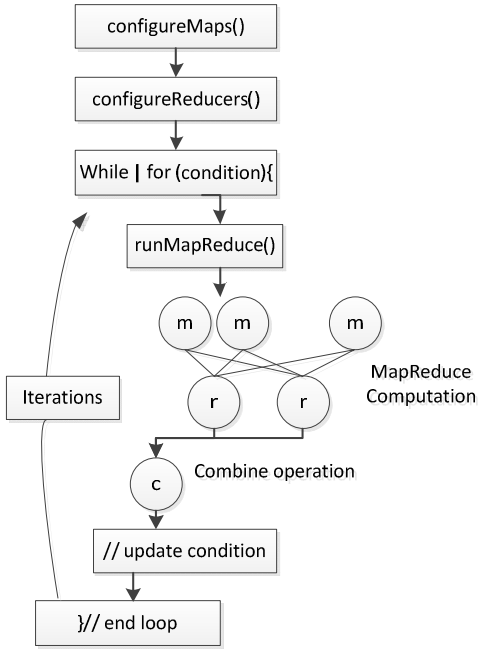

Fig. 1. Twister's programming model

Combine job is to collect MapReduce results. Twister uses scripts to operate on static input data and some output data on local disks in order to simulate some characteristics of distributed file systems. In these scripts, Twister parallel distributes static data to compute nodes and create partition file by invoking Java classes.

VI. Parallel Feature Selection Based on MapReduce

A. Parallel Feature Selection

Feature selection is usually used to select the most informative feature combination with least information loss for classification problems. If we can select the most informative variables for classification, it will save lots of computation cost and reduce the effect of noise. The information between class variable and feature variables is measured with mutual information. The parallel feature selection of classification based on mutual information metric can be formulized as follows.

Step 1: Initial dataset D is divided into N sections $D_1, D_2, \cdots, D_N$ and the sample numbers of each section are $n_1, n_2, \cdots, n_N$ respectively. Each sub dataset is deployed to each computational node. The threshold value of contribution degree λ and value α, ratio between the number of Map nodes that reached contribution degree threshold value and the number of total Map nodes, are prescribed.

Step2: Suppose S and V were two vectors and set $S=\Phi$ and $V=\{X_1,X_2,\cdots,X_m\}$, where m is the dimension value of feature variables and Φ is empty set. S denotes selected features and V denotes unselected features.

Step 3: On each computational node $i, i\in\{1,2,\cdots,m\}$, the total combinatory mutual information between all feature variables and class variable is calculated with the proposed method in section 3.

Step 4: On each computational node, the mutual information between $\{S,X_i\}, X_i\in V$ and Y is calculated. Let the number of selected features be denoted by r. The combinatory mutual information between selected feature variables and class variable is calculated according to the proposed method in section 3 if $p^k>(n_i-1)/2$, $i\in\{1,2,\cdots,N\}$, k is the value numbers of each feature variable, or else it is calculated according to (4). The contribution degree is calculated according to (7). The feature variable $X_j, j\in\{1,2,\cdots,N\}$ that maximizes $I(\{S,X_j\};Y)$ is selected. The serial number j of the selected feature, corresponding mutual information $I(\{S,X_j\};Y)$ and the flag value whether the contribution degree reaches the threshold value are collected to Reduce nodes.

Step 5: In Reduce node, the feature variable $X_j, j\in\{1,2,\cdots,N\}$ with maximum count is selected . If the counts of two feature variables are equal, the one with bigger mutual information value will be selected. Set $S\leftarrow\{S,X_j\}$ and $V\leftarrow V\setminus\{X_j\}$.

Step 6: The changed S and V are feedback to step 3. Iterate the process until α percent computational nodes' contribution degrees reach threshold value.

The selection process based on MapReduce is shown in figure 2.

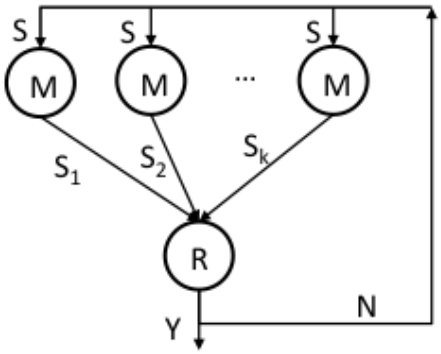

Fig. 2. Feature selection process based on MapReduce

B. Computing Complexity

Let D be a n-sample data set with m feature variables. Each feature variable has q values. r feature variables are selected. The computing complexity of different methods is analyzed as follows.

The computing complexity of feature selection method is divided into two parts, i.e. the computing cost of total combinatory mutual information c_{total} and the computation of pairwise mutual information between selected features and class variable c_m. When classic feature selection method is adopted, the computing cost is as follows.

$$c_{total}=q^N\times n\times k \tag{8}$$

$$c_m=q\times\frac{(1-q^r)}{(1-q)}\times n\times k \tag{9}$$

From above equation we can find that the computing cost based on classic feature selection method depends on the number of features and values of each feature. The computing cost will be very expensive when the number of feature variable is big.

When the proposed method in this paper is used to analyze the feature selection problem, the computing cost is cost as follows.

$$c_{total}=\begin{cases}q^N\times n\times k & if\ q^N\le n-1\\ (n-1)nk/2 & if\ q^N>n-1\end{cases} \tag{10}$$

$$c_m=\begin{cases}q\dfrac{(1-q^r)}{(1-q)}\times n\times k & if\ q^r\le(n-1)/2\\ q\dfrac{(1-q^t)}{(1-q)}\times n\times k+(n-1)nk(r-t)/2 & if\ q^t\le(n-1)/2,\ q^{t+1}>(n-1)/2\ and\ t<r\end{cases} \tag{11}$$

The computing cost mainly depends on the number of samples. It can save lots of computing cost when the number of feature is very big.

VII. PARALLEL SUPPORT VECTOR MACHINES

A. Support Vector Machines

SVM can be described as follows. It first maps the input points into a high-dimensional feature space with a nonlinear mapping function Φ and then carry through linear classification in the high-dimensional feature space. The linear classification in high-dimension feature space

corresponds to the nonlinear classification in low-dimensional input space.

Let l training samples be $T=\{(x_1,y_1),\cdots,(x_l,y_l)\}$, where $x_i \in \Omega_X = R^n$, $y_i \in \Omega_Y = R$, $i=1,\cdots,l$. Nonlinear mapping function is $k(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j)$. SVM can be implemented through solving the following equations.

$$\begin{aligned} &\min_{\alpha} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i,x_j) - \sum_{i=1}^{l} \alpha_i \\ &s.t. \quad y^T \alpha = 0 \\ &0 \le \alpha_i \le C, i=1,\cdots,l \end{aligned} \tag{12}$$

After obtaining optimum solution α^*, the following decision function is used to determine which class the sample belongs to.

$$f(x)=\mathrm{sgn}(\sum_{i=1}^{l} y_i \alpha_i^* k(x_i,x)+b^*) \tag{13}$$

It is very important to choose appropriate kernel function of SVM. The kernel function must satisfy the Mercer condition. At present, many kernel function model have been developed. The commonly used functions are polynomial function $k(x_i,x)=[(x\cdot x_i)+1]^q$, RBF function $k(x_i,x)=\exp\{-|x-x_i|^2/2\sigma^2\}$, and Sigmoid function $k(x_i,x)=\tanh(v(x\cdot x_i)+c)$ et al.

B. Parallel SVM

The parallel SVM is based on the cascade SVM model. The SVM training is realized through partial SVMs. Each subSVM is used as a filter. This makes it straightforward to drive partial solutions towards the global optimum, while alternative techniques may optimize criteria that are not directly relevant for finding the global solution. Through the parallel SVM model, large scale data optimization problems can be divided into independent, smaller optimizations. The support vectors of the former subSVM are used as the input of later subSVMs. The subSVM can be combined into one final SVM in hierarchical fashion. The parallel SVM training process can be described as in figure 2.

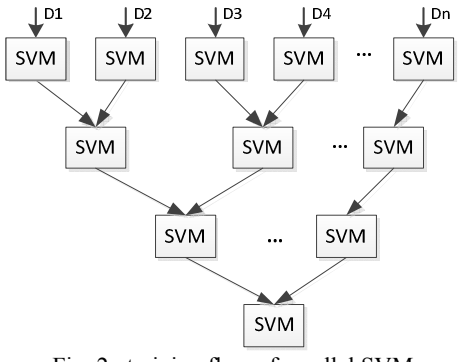

Fig. 2 training flow of parallel SVM

In the architecture, the support vectors of two SVMs are combined into one set and input to a new SVM. The process will stop when all subSVMs are combined into one SVM. In this architecture a single SVM never has to deal with the whole training set. If the filters in the first few layers are efficient in extracting the support vectors then the largest optimization, the one of the last layer, has to handle only a few more vectors than the number of actual support vectors. Therefore, the training sets of each sub-problems are much smaller than that of the whole problem when the support vectors are a small subset of the training vectors. In this paper, libSVM is adopted to train each subSVM.

VIII. EXAMPLE

A. Adult Data Analysis

1) Data source

The source data are downloaded from NEC laboratory American Inc. website http://ml.nec-labs.com/download/data/milde/. In the adult database, 123 attributes are labeled 2 classes. Each attribute denoted by binary variable, i.e. 0 or 1. Labels are denoted by +1 or -1. It is a binary classification problem. The database includes two files. One is used for training and the other is used for testing. The training file includes 32562 samples. The testing file includes 16282 samples. In this example, 4 computational nodes are used. Training data are partitioned into sections randomly. Each section has roughly equal number data.

All examples are analyzed in India cluster node of FutureGrid. Twister0.9 software is deployed in each computational node. ActiveMQ is used as message broker. The configuration of each virtual machine is as follows. Each node is installed Ubuntu Linux OS. The processor is 3GHz Intel Xeon with 10GB RAM.

2) Feature Selection

Apply the proposed parallel feature selection method on the training samples. 32562 samples are partitioned into 4, 2 and 1 sections respectively and each section is deployed to different computational node. It is impossible to calculate the total combinatory mutual information between all feature variables and class variable with classic mutual information calculation method. The computation requires echoes. The threshold value of contribution degree is set 0.9. The selected results based on different partition schedule are listed in table 1. The selected features are taken as the input of parallel SVM introduced in reference [17]. The training samples are partitioned into 4 sections and deployed to 4 computational nodes respectively. After training, 16282 test samples are used to verify the classification correct rate of trained SVM. The classification results are listed in table 1.

TABLE 1
FEATURE SELECTION AND CLASSIFICATION RESULTS

Number of nodes	Number of selected features	Feature selection time(s)	Classification Correct rate
1	23	764.47	84.32
2	25	437.3	84.02
4	26	298.43	84.3

B. Forest Covertype Data Analysis

1) Data source

The source data are downloaded from http://ftp.ics.uci.edu/pub/machine-learning-

databases/covtype/. The data is used to classify forest cover type. The original data are collected by Remote Sensing and GIS Program, Department of Forest Sciences, College of Natural Resources, Colorado State University. Natural resource managers responsible for developing ecosystem management strategies require basic descriptive information including inventory data for forested lands to support their decision-making processes. The purpose is to predict the forest cover type according to cartographic variables' values. The square of each observed section is 30 x 30 meter cell. There are 54 columns in each data item. They denote 12 variables, i.e. Elevation, Aspect, Slope, Horizontal_distance_to_hydrology, Vertical_Distance_To_Hydrology, Horizontal_Distance_To_Roadways, Hillshade_9am, Hillshade_Noon, Hillshade_3pm, Horizontal_Distance_To_Fire_Points, Wilderness_Area, and Soil_Type, where Wilderness_Area is denoted by 4 binary columns and Soil_Type is denoted by 40 binary columns. They are labeled as 7 cover types, i.e. Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, and Krummholz. There are 100000 samples in total. In this example, 40000 samples are taken as training samples and the left are taken as test samples.

2) Feature selection

In this example, many feature variables are multi-value and class variable is multi value. The value ranges of feature variables are variance. It is difficult to calculate the combinatory mutual information with classic calculation method. The proposed parallel feature selection method is applied on the training samples. 40000 samples are partitioned into 4, 2 and 1 sections respectively. The threshold value of contribution degree is set 0.9. 16, 16 and 15 features are selected respectively according to the proposed feature selection method. The selected features are taken as the input of parallel SVM. 40000 samples are partitioned into 4 sections and deployed to 4 computational nodes respectively. After training, 60000 test samples are used to verify the classification correct rate of trained SVM. The classification results are listed in table 2.

TABLE Ⅱ
FEATURE SELECTION AND CLASSIFICATION RESULTS

Number of nodes	Number of selected features	Feature selection time(s)	Classification Correct rate
1	15	1185.47	73.24
2	16	659.3	72.98
4	16	354.23	72.33

C. Results analysis

From the analysis results of the examples, we can find that the computation speed of feature selection can be improved markedly through MapReduce programming. The accelerate ratio is approximate linear. The classification results show that classification correct rates of different partition plan are similar. It illustrates that the parallel feature election method is effective and efficiency. The computation cost only depends on dataset size. It is suitable to analyze multi value and high dimension problems.

IX. CONCLUSIONS

Feature selection is an important task of machine learning and pattern recognition. Feature selection based on mutual information is taken as one of the most efficient methods. For improving the computation speed, a novel parallel feature selection method based on MapReduce is proposed. In the method, the computation cost only depends on data partition's size. It has nothing to do with the feature dimension and variable's value range. It can accelerate the computation speed almost linearly. The example analysis results show that the proposed method is efficiency in reducing computational cost. The number of selected feature can be determined with an objective rule. The classification correct rate based on parallel feature selection method is similar to that of feature selection without partition. It is scalable and efficient in dealing with large scale and high dimension problems.

REFERENCES

[1] L Yu, H Liu， "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," *Twentieth International Conference on Machine Learning, Amer Assn for Artificial,* pp. 856-863, 2003.

[2] M Dash, H Liu, "Dimensionality Reduction," *Encyclopedia of Database Systems*j, pp. 843-846, 2009.

[3] H Liu, H Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, Boston, 1998.

[4] X Yao, X D Wang, Y X Zhang, W Quan, "Summary of feature selection algorithms," *Control and Decision*, vol. 27, no. 2, pp. 161-166, 2012.

[5] Z H Xia, X M Sun, J H Qin, C M Niu, "Feature selection for image steganalysis using hybrid genetic algorithm," *Information Technology Journal*, vol. 8, pp. 811-820, 2009.

[6] K R Robbins, W Zhang, J K Bertrand, R Rekaya, The ant colony algorithm for feature selection in high-dimension gene expression data for disease classification. *Mathematical medicine and biology : a journal of the IMA*, Vol. 24, no. 4, pp. 413-426(14), 2007.

[7] Z Y Cai, J G Yu, X P Li, et al. "Feature selection algorithm based on kernel distance measure," *Pattern Recognition and Artificial Intelligence*, vol. 23, no. 2, pp. 235-240, 2010.

[8] A K Jain, P W Robert, J C Mao, "Statistical pattern recognition: A review," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, 2000.

[9] L Yu, H Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, no. 1, pp. 1205-1224, 2004.

[10] R Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans on Neural Networks*, vol. 5, no. 4, pp. 537-550, 1994.

[11] J R Swedlow, G Zanetti, C Best, "Channeling the data deluge," *Nature Methods*, vol. 8, pp. 463-465, 2011.

[12] G C Fox, S H Bae, et al. "Parallel Data Mining from Multicore to Cloudy Grids," *High Performance Computing and Grids workshop*, IOS Press. pp. 311-340, 2008

[13] B J Zhang, Y Ruan et al. "Applying Twister to Scientific Applications," *Proceedings of CloudCom, IEEE CS Press*, pp. 25-32, 2010.

[14] J Ekanayake, H Li, et al. "Twister: A Runtime for iterative MapReduce," *The First International Workshop on MapReduce and its Applications of ACM HPDC*, ACM press, pp. 810-818, 2010.

[15] S Singh, J Kubica, S Larsen, D Sorokina, "Parallel Large Scale Feature Selection for Logistic Regression," *SIAM International Conference on Data Mining*, pp. 1171-1182, 2009

[16] Z Q Sun, “Parallel Feature Selection Based on MapReduce,” *The 3rd International Conference on Computer Engineering and Network*, accepted

[17] Z Q Sun, G C Fox, “Study on Parallel SVM Based on MapReduce,” I*nternational Conference on Parallel and Distributed Processing Techniques and Applications*, CSREA Press. pp, 495-501, 2012.