ISSN:0975-9646

Classification of Real Value and Complex Value Data using Machine Learning Concepts

S.Padma [a], B.Kanchana Devi [a]

[a] *Asst Prof, Dept of Computer Science,*
K.S.R College of Arts and Science, Tiruchengode – 637215, India

Abstract- Learning is acquiring knowledge which makes man to study new things. The familiarity of the new concept is attained by the machine by giving repeated training on the same concept. In this way machines can also learn by repeated training on the same set of data. Data mining includes the concept of classification, which can be done by machine learning algorithms. Data that are to be classified can also be complex values. The performance varies if both the real and imaginary part are considered for classification. There are different machine algorithms with different features. Some of the machine learning algorithms such as Support Vector Machines (SVM), Extreme Learning Machines (ELM), Self-Adaptive Resource Allocation Network (SRAN) and Phase Encoded Complex-Valued Extreme Learning Machine (PE-CELM)are considered. This paper gives a comparison of these algorithms with its working nature and discusses the simulated results performed by these algorithms on balanced and imbalanced dataset for complex values and real values.

Keywords: **(Machine Learning, Support Vector Machines, Extreme Learning Machines, Self-Adaptive Resource Allocation Network, Phase Encoded Complex-Valued Extreme Learning Machine.)**

1. INTRODUCTION

Data mining (Knowledge Discovery) is the process of analyzing data from different perspectives and summarizing it into useful information. Using data mining the data can be classified, clustered, associated and finding patterns. Learning in data mining can be classified as supervised learning and unsupervised learning. Supervised learning is learning contents on some fixed rules whereas unsupervised learning is learning using some measures such as distance etc. Classification of data plays a vital role in research which can be used for prediction and comes under the category of supervised learning. There are various methods of classification such as decision tree, Naïve Bayes, neural networks etc.

Neural networks are framed based on the neuron structure of the human brain. While learning the new concept the brain may not be aware what it is. If training is given on the new dataset repeatedly then learning is made quicker. This concept is used for classifying data using machine learning. The dataset will be divided into training and testing. The training dataset contains the features and the class it belongs. The machine learning algorithms are used to train the neurons based on the values of the training dataset. If the neurons are repeatedly trained with the dataset then it can predict the dataset which is given as test data.

The network for machine learning can be either feed-forward or back propagation. Feed forward networks executes in a unidirectional way where it may be either single layer or multi layer depending on the application. Back propagation network executes in a loop form where the output is again sent as input to the network which is used to reduce the error. The feed forward network includes sequential learning and batch learning.

Data that are used for classification contains only the real part. But the original value of a data will be a complex number which contains both real and imaginary part. Using imaginary part may be difficult for classifiers. But the performance of complex values shows improvement when compared to others.

This paper focuses on three machine learning algorithms SVM, ELM, SRAN, PE-CELM. The working nature of the algorithms is discussed in the section 2. Section 3 describes the simulated results for the algorithms and comparison on the features of the algorithm. Section 4 concludes the study.

2. METHODS

2.1. Support Vector Machines

SVM (Support Vector Machines) are a useful technique for data classification. Al-though SVM is considered easier to use than Neural Networks, users not familiar with it often get unsatisfactory results at first. A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one target value" (i.e. the class labels) and several attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.[1]

2.2. *Extreme Learning Machine*

ELM was originally proposed (Huang) for standard single hidden layer feed forward neural networks (with random hidden nodes (random features)).[2][3] ELM provides a unified learning platform with widespread type of feature mappings and can be applied in regression and multi-class classification applications directly. [4][5][6] From the optimization method point of view ELM has milder optimization constraints compared to SVM, LS-SVM and PSVM. [7][8][9][10]

The ELM algorithm is based on the following two principles.

1. When the number of training samples equals the number of hidden nodes, i.e., $N = \tilde{N}$, one can randomly assign the parameters of hidden nodes (the input weights and biases for additive hidden nodes or the centers and impact factors for RBF) and based on this analytically calculate the output weights by simply inverting H and realize zero training error. Calculation

of the output weights is done in a single step here. There is no need for any lengthy training procedure where the network parameters are adjusted interactively with appropriately chosen control parameters (learning rate and learning epochs, etc.).

2. When the number of training samples is greater than the number of hidden nodes, i.e., $N > \tilde{N}$,one can still randomly assign the parameters of hidden nodes and calculate the output weights by using a pseudo inverse of to give a small nonzero training error $\in > 0$. Here also the output weights' calculation is done in a single step and does not need lengthy training procedure.

The standard SLFNs with $\tilde{N}$ hidden nodes with activation function $g(x)$ can approximate these N samples with zero error means by $\sum_{j=1}^{\tilde{N}} \| o_j - t_j \| = 0,$ i.e., there exist β_i, wi and bi such that

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(w_i.x_j + b_i) = o_j, \quad j = 1, 2, ..., N, \tag{1}$$

The above equations can be rewritten compactly as

$$H\beta = T, \tag{2}$$

Where $H(w_1, ..., w_{\tilde{N}}, b_1, ..., b_{\tilde{N}}, x_1, ..., x_N)$

$$= \begin{bmatrix} g(w_1.x_1 + b_1) & ... & g(w_{\tilde{N}}.x_1 + b_{\tilde{N}}) \\ . & & . \\ g(w_1.x_N + b_1) & ... & g(w_{\tilde{N}}.x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ . \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \text{and} \quad T = \begin{bmatrix} t_1^T \\ . \\ t_N^T \end{bmatrix}_{N \times m} \tag{4}$$

2.3.Self Adaptive Resource Allocation Network

In the setting of standard online/sequential learning, the training sample arrives one at a time and the network adapts its parameters based on the difference in knowledge between the network and the current sample. When new sample (x^t) is arrived to the network, based on the sample error (e), the sample is either used for network training (growing/learning) immediately, pushed to the rear end of the stack for learning in future, or deleted from the data set. The detailed description of SRAN is explained in [11].

The control parameters in the sequential algorithm are self-regulated, so, they are fixed, and are mostly independent of the problem considered. The control parameters alter the sequence in which the SRAN classifier approximates the decision function, based on the difference between the information contained in each sample and the knowledge acquired by the network. The higher the difference, the earlier a sample participates in learning. A few samples with lesser differences are pushed to the rear end of the sample data stack. These samples are later used to fine-tune the network parameters. Also, a few samples with redundant information are discarded from the training data set, thus avoiding over-training. Thus, the finally realized network is compact and provides better generalization performance.[12]

The sequence of the training sample is controlled internally using self- regulated control parameters as explained below. The self-regulated control parameters (η_a, η_1) identify the sample 't' with maximum information. The learning process of SRAN involves allocation of new hidden neurons, as well as adjusting network parameters. If the current sample 't' does not satisfy the learning criteria, then the current sample is stacked at the rear end of the sequence, for future use. These samples do not take part in training at this stage. Without loss of generality, let us assume that the network has K hidden neurons from t-1 training samples. The deletion of samples, growing/learning, and sequence altering are the principles of self-regulation system.

The output of the SRAN classifier $(\hat{y} = [\hat{y}_1, ..., \hat{y}_n]^T)$ with K hidden neurons has the following form:

$$\hat{y}_i = \sum_{j=1}^{K} \alpha_{ij} y_h^j, \quad i = 1, 2, .., n \tag{5}$$

$$y_h^j = \exp\left(-\frac{\| x - \mu_j^I \|^2}{(\sigma_j^I)^2}\right) \tag{6}$$

where μ_j^I is the jth neuron center corresponding to the ith class, σ_j^I is the width of the jth neuron and α_{ij} is the weight connecting the *i*th output neuron and *j*th Gaussian neuron.

The predicted class label *cc* for the new training sample is given by

$$\hat{c} = \arg \max_{i \in 1,2,..,n} \hat{y}_i \tag{7}$$

2.4 Phase Encoded Complex-Valued Extreme Learning Machine

To classify the complex values PE-CELM can be used. The working nature of this is same as ELM which has been discussed above.The complex-valued activation function used for the hidden neurons in this method is hyperbolic secant function.

The neurons at the input layer transform the real – valued input features to the complex domain. Phase encoding is used to transform the input layer. The parameters of the hidden neurons are chosen randomly and parameters of the output neurons of the network are estimated analytically.

The PE-CELM algorithm is as follows :

1. For a given training set (X,Y), select the appropriate number of hidden neurons K.
2. Choose the scaling factor and the neuron centers randomly.
3. Calculate the output weights analytically.

2.5 Comparison of the Methods

The table 1 gives the comparison of the three methods SVM, ELM and SRAN with different features. They differ in the type of learning but the other features such as the layer type, classification method and network used are same. But the learning method of an algorithm gives a vast change in the efficiency of the algorithm.

2.6 .Datasets

The datasets that are used for classification may be real, categorical etc. In the complete dataset a part is treated as training dataset and the remaining is testing dataset. As the classification is based on supervising learning the machines that are used for classification must be trained using the trained dataset. When the test dataset is given then the machine will easily predict the class it belongs as it was already trained.

Depending on the number of values in the dataset it may be said as balanced and imbalanced dataset. If a dataset contains data that is equal in number for all classes then it is termed as balanced dataset otherwise imbalanced dataset. The efficiency of the algorithm will have deviation with these dataset types.[13]

Table 1 – Comparison of Methods

State	SVM	ELM	SRAN	PE-CELM
Type of layer used	Both single and multilayer	Both single and multilayer	Both single and multilayer	Both single and multilayer
Type of Classification	Multiclass	Multiclass	Multiclass	Multiclass (Complex Values)
Feed Forward/ Back Propagation	Feed Forward	Feed Forward	Feed Forward	Feed Forward
Type of Learning	Batch	Batch, Sequential and Incremental	Sequential	Batch, Sequential and Incremental

Table 2 –Comparison of Datasets

Dataset		SVM	ELM	SRAN	PE-CELM
Balanced	Iris	90.62	96.19	96.19	96.5
	Image Segmentation	90.62	90.67	92.29	92.5
	Wine	98.04	98.09	97.19	98.2
Imbalanced	VC	67.99	77.59	76.86	80.4
	GI	60.01	68.46	80.95	88.0
	HEART	75.10	75.91	77.53	79.12
	LD	70.21	71.41	65.78	73.32
	PIMA	76.43	75.4	74.90	76.50
	BC	97.06	96.35	97.26	98.53
	ION	88.51	89.64	90.84	91.12

3. RESULTS AND DISCUSSION

The contents of the table 2 gives the average test efficiency of different datasets with four important machine learning concepts.[14]

The results in table 2 list the average test efficiency of different datasets. It shows the difference in the balanced dataset and imbalanced dataset. As an overall analysis on the balanced dataset it can be described that PE-CELM stands top followed by SRAN then ELM and then by SVM. When the comparison extends to imbalanced datasets PE-CELM works effectively compared to all the others. The remaining result is both SRAN and ELM stands at more or less equal level followed by SVM for some datasets and not for all. The following figures explain the analysis in detail.

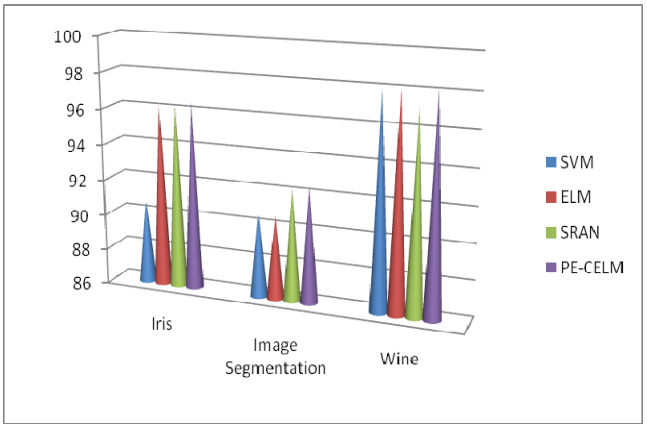

Fig. 1. Balanced Dataset Comparison

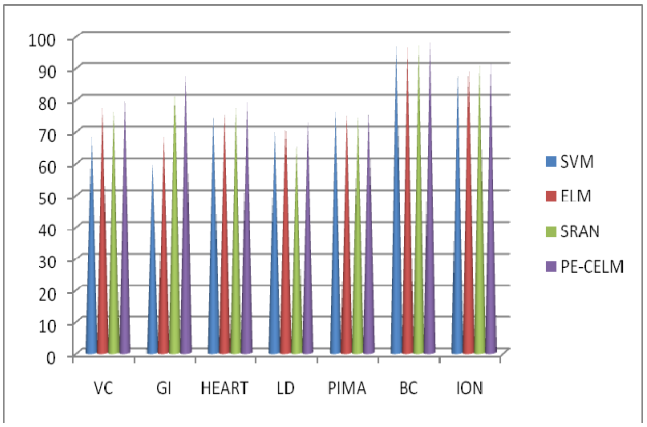

Fig. 2. Imbalanced Dataset Comparison

The “fig 1” and “fig 2” represents the variation of each dataset on three different approaches. Depending on the dataset the selection of the method is essential. Balanced dataset such as iris, image segmentation and wine are compared with machine learning algorithms such as SVM, ELM and SRAN and PE-CELM, The result shows that for all balanced datasets the performance of SVM is low compared to ELM, PE-ELM and SRAN. The performance of PE-CELM is high compared to ELM, SRAN and SVM.

Seven imbalanced dataset are taken for execution. The result shows that for all balanced datasets the performance of SVM is low compared to ELM, PE-ELM and SRAN. The performance of PE-CELM is high compared to ELM, SRAN and SVM.

4. CONCLUSION

This paper gives an overview of different machine learning techniques for classification. Depending on the type of learning such as batch, sequential and incremental the method of classification can also be selected. The working nature of the machines differs from each other which were discussed in this paper. The method of learning changes from one machine to the other. Due to this the efficiency of the experiments gets varied. The table gives the comparison of the efficiency of different algorithms on different dataset. Depending on the dataset either balanced or imbalanced any one of the above specified method can be used. These machine learning techniques can also be implemented on the forth coming researches with big data and others.

ACKNOWLEDGEMENT

The authors are thankful to Dr S. Suresh Nanyang Technological University, Singapore for the valuable help provided for doing this research work.

REFERENCES

1. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, A Practical guide to Support Vector Classification, April 2010.
2. Guang-Bin Huang*, Qin-Yu Zhu, Chee-Kheong Siew. Extreme learning machine: Theory and applications [J] 2006
3. http://www.ntu.edu.sg/home/egbhuang/index.htm
4. Vasiliy, S, S. Suresh, R. Savitha and H. J. Kim, "An ensemble of extreme learning machines for Steaganalysis based on risk sensitive hinge loss function," cognitive Computation, 2013 (Accepted)
5. S. Saraswathi, S. Suresh, N. Sundararajan, M. Zimmermann, and M. N., Hamilton, ICAG-PSO-ELM approach for Accurate Multi-Class Cancer Classification Resulting in Reduced Gene Sets in Which Genes Encoding Secreted Protein are Highly Represented, IEEE/ACM Trans. On Computational Biology and Bioinformatics, 8(2), pp. 452-463, 2011.
6. S. Suresh, S. Saraswathi and N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse data classification problems, Engineering Application of Artificial Intelligence, 23 (7), pp. 1149-1157, 2010.
7. S. Suresh, R. Venkatesh Babu and H. J. Kim, No-reference Image Quality Assessment using Modified Extreme Learning Machine Classifier, Applied Soft Computing, 9(2), 541-552, 2009.
8. G.B. Huang, Q.Y. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1-3) (2006) 489-501.
9. G.B. Huang, M. Li, L. Chen, C.K. Siew, Incremental extreme learning machine with fully complex hidden nodes, Neurocomputing 71 (4-6) (2008) 576-583.
10. G.B. Huang, X. Ding, H. Zhou, Optimization method based extreme learning machine for classification, Neurocomputing 74 (1-3) (2010) 155-163.
11. S. Suresh, Keming Dong, H.J. Kim A sequential learning algorithm for self-adaptive resource allocation network classifier [J] Elsevier Neurocomputing 2010.
12. S. Suresh, S. Saraswathi, N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse data classification problems, Eng. Appl. Artif. Intell. 23 (7) (2010) 1149-1157.
13. Padma, S., Kumar, S.S. , Manavalan, R., Performance analysis for classification in balanced and unbalanced data set, Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on (2011) 300-304.
14. G.Sateesh Babu, S.Suresh., Meta – Cognitive neural network for classification problems in a sequential learning framework [J] Elseiver Neurocomputing 2012.