2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

Comparing Performance of Interval Neutrosophic Sets and Neural Networks with Support Vector Machines for Binary Classification Problems

Pawalai Kraipeerapun and Chun Che Fung
School of Information Technology, Murdoch University, Perth, Australia
Email: {p.kraipeerapun | l.fung}@murdoch.edu.au

Abstract— **In this paper, the classification results obtained from several kinds of support vector machines (SVM) and neural networks (NN) are compared with our proposed classifier. Our approach is based on neural networks and interval neutrosophic sets which are used to classify the input patterns into one of the two binary class outputs. The comparison is based on several classical benchmark problems from UCI machine learning repository. We have found that the performance of our approaches are comparable to the existing classifiers. However, our approach has taken into account of the uncertainty in the classification process.**

Index Terms— **neural network, interval neutrosophic sets, support vector machine, binary classification, uncertainty**

I. Introduction

In this paper, we aim to compare the accuracy of our results obtained from our previous work [1] with the results obtained from several kinds of support vector machines and other existing classifiers. It is understood that it may be difficult to compare the results from different types of classifiers without the same testing environments. However, we intend to compare the results in general while we recognize that different classifiers may be suitable for different problems.

A. Support Vector Machines

A Support Vector Machine (SVM) is an algorithm that can learn to recognize objects by examining training samples. SVM can be described in four basic concepts: the separating hyperplane, the maximal margin hyperplane, the soft margin, and the kernel function [2]. There are several techniques used to improve the performance of SVM. For example, when the training set is large, SVM can be improved by using incremental learning in which the subsets of training data are considered one at a time, and then all results are combined [3]. In [3], Syed et al. proposed SV-incremental learning algorithm, in which each new batch of data, together with the support vectors obtained from the previous learning step, were trained. However, a drawback of SV-incremental learning is that the old support vectors obtained from the previous learning step can be considered as the outliers if the new batch of data is distributed differently from the old support vectors. In order to address this problem, Rüping [4] proposed SV-L-incremental algorithm based on SV-incremental algorithm. In his experiment, he added more weight on an error obtained from the previous support vectors than an error from a new batch data.

A Least Squares SVM (LS-SVM) is another modified SVM, which was introduced by Suykens [5]. A LS-SVM involves in a linear equation instead of a quadratic programming problem that is involved in the traditional SVM. Hence, it is found to provide a low computational complexity cost [6]. However, a drawback of LS-SVM is that its sparseness is lost. Instead of training with the only support vectors, the whole input data is trained. Valyon [7] improved LS-SVM by introducing a Least Squares version of the Least Squares Support Vector Machine (LS^2-SVM). His algorithm covers a sparse solution by reducing the number of columns in a kernel matrix, but the approach still preserves the quality solution. Consequently, the number of training input data is decreased.

In Proximal Support Vector Machine Classification (PSVM) [8], instead of using a separating plane in the classification, two parallel planes are used. Both planes are generated as far away as from each other, whereas each plane is closest as possible to the points belonging to one of the two classes. PSVM may be considered as a regularized least squares SVM. It was found to provide comparable results to the traditional SVM, however PSVM performed considerably faster.

A Fuzzy Proximal Support Vector Machine (FPSVM) [9] is an extension of PSVM. Fuzzy membership values are assigned to data points before these points are assigned to the two parallel planes. The point with a high membership value is more important than the one with a lower membership value. FPSVM was found to provide better performances than PSVM and SVM. It was also found that the FPSVM was significantly faster than the traditional SVM.

In recent years, a Generalized Eigenvalue Proximal Support Vector Machine (GEPSVM) was proposed by Mangasarian and Wild [10]. This modified SVM applies two non-parallel planes instead of parallel planes used in PSVM. Objects or points belonging to each class are proximal to each plane. Two generalized eigenvalue problems are generated. The smallest eigenvalue of each generalized eigenvalue problem is correspondent to the eigenvector that forms each non-parallel plane. The generalized eigenvalue problem is a simple problem that can be solved easier and faster than the optimization algorithm used in SVM-Light, which is an implementation of the traditional SVM [10].

2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

Khemchandani and Chandra [11] proposed Twin Support Vector Machine (TWSVM), which is also a non-parallel plane classifier but different formulation from the GEPSVM. In TWSVM, two smaller size of quadratic programming problems are solved instead of the large one used in the traditional SVM. The constraints of each quadratic programming problem are determined by patterns belonging to each class. TWSVM was found to perform four times faster than the traditional SVM.

The previous paragraphs explain only some examples of modified SVM that are comparable or better than the traditional SVM. There is still a lot more research on the modifying SVM, in which we cannot explain all of them in this paper. Only some of them that test the performance of their algorithms based on the classical benchmark UCI data sets are shown in this paper. In section II , some classification accuracy results obtained from those modified SVM are represented and compared.

B. Some other classifiers

In this section, some other types of classification algorithms that have been tested for their performance based on UCI data sets are described.

In [12], Yang and Honavar applied a genetic algorithm to select a subset of features to represent the patterns to be classified based on neural networks constructed by DistAI, which is a constructive neural network that adds hidden neurons one at a time.

In [13], Draghici created the constraint based decomposition (CBD) technique, which is a constructive neural network technique guaranteed the convergence and can deal with both binary and multiclass problems. This technique was found to be able to solve complicated problems fast and provide reliable solutions.

Schetinin et al. [14] compared the results obtained from the randomized decision tree (DT) ensemble technique to the Bayesian decision tree with restarting strategy technique. They found that the Bayesian decision tree technique provided superior results and performed two or three times faster than the other technique.

Frank and Pfahringer [15] proposed a method named input smearing. Bagging was modified using their method. In this method, after bootstrap resampling was used to select attribute values for each bag, each attribute value was modified. An attribute value was transformed into a smeared value by adding Gaussian noise to the original attribute value. The noise threshold value was set using cross-validation. It was found that their method can improve the performance compared to a single trees and bagging.

Sridharan et al. [16] proposed a competitive finite mixture of neurons, a mixture of experts model with competitive penalties between the experts. An Expectation Maximization (EM) algorithm was used for learning the weights of each neuron. It was found that their method provided superior performances over neural networks and two types of SVMs.

In [17], Chen et al. proposed a simulated annealing (SA) approach to select a subset of features used in the classification. The optimal parameters are also found by applying Hide-and-Seek SA, used to solve the optimization problem with continuous decision variables. They claimed that their method can provide the best architecture and parameter setting for BPNN.

Yeung et al. [18] proposed a generalization error model based on the localized generalization error using the stochastic sensitivity measure. An architecture selection method named MC^2SG is also proposed based on their generalization error model. Their proposed method can be applied to any classification problems with different numbers of samples, features, and classes. In their experiment, MC^2SG is used to find the number of hidden neurons for a radial basis function neural network.

There is still a lot more research on how to determine the best approach to solve the problem of classification. However, we select only some of them to be compared in this paper and to compare the performance of our proposed approach. The results obtained from the above examples are summarized in Table III.

C. Proposed Technique Using Neural Networks and Interval Neutrosophic Sets

In our previous paper [1], instead of using only a single neural network for the classification, our approach was applying a pair of opposite neural networks used to predict degree of truth and false membership values. The difference between the truth and false membership values was used to represent uncertainty in the classification. This uncertainty value was considered as the indeterminacy membership value. The predicted truth and false membership values were compared in order to give us the binary classification results, whereas the indeterminacy membership value gave us the confidence level of the classification. We have presented the three outputs: truth membership, indeterminacy membership, and false membership in the form of an interval neutrosophic set [19]. In our research, we follow the definition of interval neutrosophic sets defined in [19].

Moreover, we also extended our proposed single pair of neural networks to bagging neural networks. In [1], an ensemble of pairs of neural networks was created. Each pair applied the same bag for training, in which each bag of data was created using bootstrap resampling. For each input pattern, the truth membership values obtained from all components are dynamically weighted average. Also, all false membership values are dynamically weighted average. The weight is computed and normalized from the complement of uncertainty occurred in the classification for each pattern. After that, the weighted average truth membership and the weighted average false membership values are compared in order to classify the input pattern into a binary class. The classification accuracy results obtained from both single pair and ensemble of pairs of neural networks are shown in the next section.

II. Classification Accuracy Comparison

The previous section describes only some examples of modified SVM and several other types of classifiers. For each research, the environments are controlled and set to make it suitable for data sets and algorithms used in the research.

2008 Second IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2008)

TABLE I
DATA SETS USED IN THE COMPARISON.

Name	No. of Features	Size of Samples
heart-statlog	13	270
ionosphere	34	351
bupa liver	6	345
mushrooms	22	8124
pima indians	8	768
sonar	60	208
tic-tac-toe	9	958

However, we cannot exactly claim that which is the most suitable algorithm used for the unknown data sets. The purpose of this paper is to compare the classification accuracy of the results obtained from some kinds of SVM and several other types of classifiers based on a number of benchmarking UCI data sets. We compare results obtained from different research reported with different types of algorithms and parameters setting based on the same benchmark data sets. Those researches provide techniques with different pros and cons. However, most of them provide similar accuracy results. The question is that which technique should be selected to solve the problem with the unknown data set. The question is still an ongoing challenge as it requires detailed information on the characteristics of the unknown data.

In this paper, we aim to present the comparison among some of the existing techniques and compare them to our proposed techniques in order to show that our proposed techniques are capable to provide similar results and accuracy comparable with the other techniques.

The results obtained from several existing classification techniques are shown and compared based on seven UCI data sets [20]. The characteristic of these data sets are shown in Table I. Table II shows the comparison among nine different techniques of SVM classification, whereas Table III presents the comparison among some other classifiers including our proposed techniques.

In Table II and III, the first row shows the name of the classifiers. The second row shows the reference papers that provide results shown in each column. In Table II, all results shwon in column SVM, SV-Inc., SV-L-Inc., PSVM, GEPSVM, and TWSVM were obtained from 10-fold cross-validation. In column LS-SVM, each result was computed from 50 randomization runs. In column LS^2-SVM, there is no explanation about how the results were obtained. In column FPSVM, all results were obtained using 5-fold cross-validation.

In Table III, each data set shown in column GA-Selected Subset was partitioned ten times. Each time, the data set was separated into 90% training set and 10% test set. Each partition was used in five independent runs of the genetic algorithm. The average results were represented in the experiment. In column CBD, each data set was randomly split into 80% training set and 20% test set. For each data set, the average results were obtained over five trails on the test data. In column Bayesian DT, the results were evaluated on 5-fold cross-validation. In column Input Smearing, each data set was randomly partitioned into 90% training set and 10% test set. The size of an ensemble is ten. The results are averaged over 100 runs. In column Mix. of Neurons, each data set was randomly separated into 60% training set and 40% test set. The results were averaged over 20 trials. In column SA+BPNN, all results are obtained from 10-fold cross-validation. In column MC^2SG, each data set was divided into 50% training set and 50% test set. The classification accuracies on the test sets were averaged based on ten independent runs for each data set.

The results shown in the last four columns were obtained from our previous paper [1]. We applied feed-forward back-propagation neural networks to each pair of the networks used in the experiment. Each data set was split into a training set containing 80% of the data and a testing set containing 20% of the data. For each data set, the classification accuracies on the test sets were averaged based on twenty runs with twenty different randomized training sets. In column BPNN, only the truth membership values were considered in the classification. In column T>F, both truth and false membership values were compared to give us the binary classification results. In the last two columns, a bagging technique was applied to a single neural network and a pair of neural networks, respectively.

III. CONCLUSION

There are several techniques used for binary classification. Each technique has different pros and cons. For example, some modified SVMs provide the results comparable to the results obtained from the traditional SVM. However, they can perform faster or provide less complexity than the traditional SVM. The integration of neural network and some other existing techniques can also improve the classification performance compared to the traditional neural network. Our proposed technique is also using the integration of the traditional back-propagation neural network (BPNN) and the existing theory called interval neutrosophic sets. From Table II and III, we found that the results obtained from our technique outperform the results obtained from the traditional BPNN, and they are also comparable to the results obtained from some classifiers. Most classifiers shown in this paper concentrate only on the truth membership values. However, our approach can represent three types of membership values: truth membership, indeterminacy membership, and false membership values. The indeterminacy membership is used to represent degree of uncertainty in the classification. The relationship among these three memberships can be used to support the confidence level in the classification. Hence, if the classification is involved in the uncertain information then our approach is another technique that can be chosen for the prediction.

REFERENCES

[1] P. Kraipeerapun, C. C. Fung, and K. W. Wong, "Ensemble Neural Networks Using Interval Neutrosophic Sets and Bagging," in *Proceedings of the Third International Conference on Natural Computation (ICNC'07)*, vol. 1, Haikou, China, August 2007, pp. 386–390.

[2] W. S. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, 2006.

[3] N. A. Syed, H. Liu, and K. K. Sung, "Incremental Learning with Support Vector Machines," in *Proceedings of the Workshop on Support Vector Machines at the International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, 1999.

TABLE II

CLASSIFICATION ACCURACY COMPARISON AMONG OUR PROPOSED TECHNIQUE (COLUMN T>F) AND SEVERAL EXISTING SVM CLASSIFICATION TECHNIQUES.

Name	SVM	SV-Inc.	SV-L-inc.	LS-SVM	LS^2-SVM	PSVM	FPSVM	GEPSVM	TWSVM
Reference	[3]	[3]	[4]	[6]	[7]	[8]	[9]	[10]	[11]
heart-statlog	84.45 (Gaussian)	79.69 (Gaussian)	83.33 (RBF)	84.6 (RBF)	70.00 -		83.89 (Linear)		84.44 (Linear)
ionosphere	94.57 (Gaussian)	93.14 (Gaussian)	95.45 (RBF)	95.0 (RBF)		95.2 (Non-Linear)			88.03 (Linear)
bupa liver	68.68 (Gaussian)	61.03 (Gaussian)	70.78 (RBF)	71.1 (RBF)	70.44 -	73.6 (Non-Linear)	64.18 (Linear)	63.8 (Gaussian)	67.83 (RBF)
mushrooms	100.00 (Linear)	99.88 (Linear)	100.00 (Linear)			88.0 (Non-Linear)		81.1 (Linear)	
pima indians				77.0 (RBF)	68.36 -	77.5 (Linear)		73.6 (Linear)	73.70 (Linear)
sonar	87.38 (Gaussian)	87.41 (Gaussian)	85.07 (RBF)	75.5 (RBF)					77.26 (Linear)
tic-tac-toe				98.9 (RBF)	94.37 -	98.4 (Non-Linear)			

TABLE III

CLASSIFICATION ACCURACY COMPARISON AMONG OUR PROPOSED TECHNIQUE (COLUMN T>F) AND SEVERAL EXISTING CLASSIFICATION TECHNIQUES.

Name	GA-Selected Subset	CBD	Bayesian DT	Input Smearing	Mix. of Neurons	SA+BPNN	MC^{2}SG	BPNN	T>F	BPNN bagging	T>F bagging
Reference	[12]	[13]	[14]	[15]	[16]	[17]	[18]	[1]	[1]	[1]	[1]
heart-statlog	93.9			80.8							
ionosphere	98.6	88.17	95.35	91.6	87.79	98.60	84.71	93.54	96.42	98.15	98.21
bupa liver	77.8	62.32		69.0	70.72			62.68	66.52	71.30	74.13
mushrooms							99.95				
pima indians	79.5	68.72	79.69	75.3	77.69	82.16	79.40	70.49	74.74	76.38	77.97
sonar	97.2		81.43	81.5			83.20				
tic-tac-toe		75.10									

[4] S. Rüping, "Incremental Learning with Support Vector Machines," in *Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM '01)*, 2001, pp. 641–642.

[5] J. A. K. Suykens and J. Vandewalle, "Least Squares Support Vector Machine Classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[6] B. Baesens, S. Viaene, T. van Gestel, J. Suykens, G. Dedene, B. D. Moor, and J. Vanthienen, "An Empirical Assessment of Kernel Type Performance for Least Squares Support Vector Machine Classifiers," in *The Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, UK, August 2000, pp. 313–316.

[7] J. Valyon and G. Horváth, "A Sparse Least Squares Support Vector Machine Classifier," in *International Joint Conference on Neural Networks (IJCNN 2004)*, 2004, pp. 543–548.

[8] G. Fung and O. L. Mangasarian, "Proximal support vector machine classifiers," in *KDD*, 2001, pp. 77–86.

[9] Jayadeva, R. Khemchandani, and S. Chandra, "Fast and robust learning through fuzzy linear proximal support vector machines," *Neurocomputing*, vol. 61, pp. 401–411, 2004.

[10] O. L. Mangasarian and E. W. Wild, "Multisurface Proximal Support Vector Machine Classification via Generalized Eigenvalues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 69–74, 2006.

[11] J. R. Khemchandani and S. Chandra, "Twin Support Vector Machines for Pattern Classification," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

[12] J. Yang and V. Honavar, "Feature Subset Selection Using a Genetic Algorithm," *IEEE Intelligent Systems*, vol. 13, no. 2, pp. 44–49, 1998.

[13] S. Draghici, "The constraint based decomposition (CBD) training architecture," *Neural Networks*, vol. 14, no. 4–5, pp. 527–550, 2001.

[14] V. Schetinin, D. Partridge, W. J. Krzanowski, R. M. Everson, J. E. Fieldsend, T. C. Bailey, and A. Hernandez, "Experimental Comparison of Classification Uncertainty for Randomised and Bayesian Decision Tree Ensembles," in *IDEAL*, ser. Lecture Notes in Computer Science, Z. R. Yang, R. M. Everson, and H. Yin, Eds., vol. 3177. Springer, 2004, pp. 726–732.

[15] E. Frank and B. Pfahringer, "Improving on Bagging with Input Smearing," in *PAKDD*, ser. Lecture Notes in Computer Science, W. K. Ng, M. Kitsuregawa, J. Li, and K. Chang, Eds., vol. 3918. Springer, 2006, pp. 97–106.

[16] K. Sridharan, M. J. Beal, and V. Govindaraju, "Competitive Mixtures of Simple Neurons," in *The 18th International Conference on Pattern Recognition (ICPR'06)*, vol. 2, Hong Kong, 2006, pp. 494–497.

[17] S.-C. Chen, S.-W. Lin, T.-Y. Tseng, and H.-C. Lin, "Optimization of Back-Propagation Network Using Simulated Annealing Approach," in *IEEE International Conference on Systems, Man, and Cybernetics*, Taipei, Taiwan, October 2006, pp. 2819–2824.

[18] D. S. Yeung, W. W. Y. Ng, D. Wang, E. C. C. Tsang, and X.-Z. Wang, "Localized Generalization Error Model and Its Application to Architecture Selection for Radial Basis Function Neural Network," *To appear in IEEE Transaction on Neural Networks*, 2007.

[19] H. Wang, D. Madiraju, Y.-Q. Zhang, and R. Sunderraman, "Interval neutrosophic sets," *International Journal of Applied Mathematics and Statistics*, vol. 3, pp. 1–18, March 2005.

[20] A. Asuncion and D. Newman, "UCI Machine Learning Repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html