Improvement of FCM Neural Network Classifier using K-Medoids Clustering

Xiaoqian Zhang, Bo Yang*, Lin Wang and Zhifeng Liang
Shandong Provincial Key Laboratory of Network based Intelligent Computing, University of Jinan, Jinan, 250022, China
*Corresponding Author
E-mail: yangbo@ujn.edu.cn

Ajith Abraham
[1]IT4Innovations, VSB-Technical University of Ostrava, Ostrava, Czech Republic
[2]Machine Intelligence Research Labs (MIR Labs), WA, USA
E-mail: ajith.abraham@ieee.org

Abstract—**Floating Centroids Method (FCM) is a new method to improve the performance of neural network classifier. But the K-Means clustering algorithm used in FCM is sensitive to outliers. So this weakness will influence the performance of classifier to a certain extent. In this paper, K-Medoids clustering algorithm which can diminish the sensitivity to the outliers is used to partition the mapping points into some disjoint subsets to improve FCM's robustness and performance. Some data sets from UCI Machine Learning Repository are employed in our experiments. The results show a better performance for the FCM using our improved method.**

Keywords-neural network; classification; clustering; K-means; K-medoids; Floating Centroids Method

I. INTRODUCTION

Classification is an important form of data analysis in data mining. Classification tasks are often divided into two types: supervised and unsupervised classification. The model of classifier can be created in the training process. There are many classification techniques including decision trees[1], Bayesian network[2][3], neural network[4], support vector machines[5][6], etc. As we know, decision tree is simple and easy to be comprehended. Bayesian method based on Bayes theorem is a statistical classification. SVM is a powerful machine learning method developed on the Statistical Learning Theory. The wonderful classification performance has been proved in many classification tasks. To a certain extent, SVM can overcome dimension disaster and over-fitting problem. Neural network is also a popular method in solving classification tasks.

Many methods have been proposed to finish the tasks in neural network classification. One-Per-Class is a method to solve multi-classification problems, in which each output of neural network is regarded as probability belonging to corresponding class. Since classification is an exclusive task, Bridle[7]used SoftMax (softmax activation function) to ensure that the total value of output probabilities equals to 1. Dietterich and Bakiri proposed error-correcting output code (ECOC)[8]based on statistical robustness properties to improve the classification performance. The number and position of centroids, the corresponding relationship between centroids and classes are set manually in above methods. Further division of partition space (FDPS)[9]divides the partition space into some shape-fixed regions. FDPS combined with FNT (Flexible Neural Tree)[10][11][12] are used to solve classification tasks and have been proved to have a good classification results and performance. Lin Wang and Bo yang propose a Floating Centroids Method (FCM)[13] to solve the limitation of shape-fixed partitions. The neural network classifier using FCM has been used to solve some problems of cement and other tasks.

The rest of this paper is organized as follows: Section II reviews two new neural network classifier methods and several novel types of neural network. Section III introduces the neural network classifier using our improved method. Some experimental results are presented in Section IV to show the performance. Finally, Section V concludes the paper.

II. RELATED WORKS

A. FDPS

Unlike traditional partition space, the partition space in FDPS[9] is divided into fixed shape partitions. The dimension of the partition space and number of partitions in each dimension are set by users. If dimension is set to 2, the partition number in each axe is 4. The partition space can be divided into 16 partitions showed in Fig. 1.

Training samples are mapped into the partition space by neural networks. Every partition in the partition space will be labeled by a class according to the distribution of mapped samples. If mapped samples of one category are majority, this class will label the partition. Particle Swarm Optimization (PSO) is used to optimize neural network. Then, we get an optimized partition space corresponding to the optimized neural network.

B. Floating centroids method

In FDPS, the partition space is some shape-fixed partitions. However, Shape-fixed partition cannot distinguish the boundary clearly. This weakness in FDPS will lead to

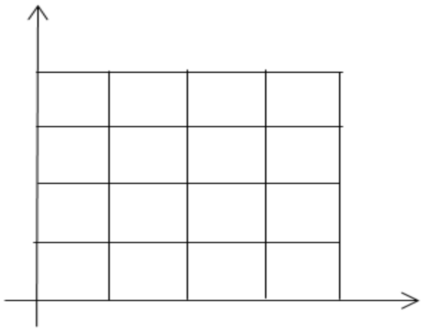

Figure 1 partition space having 16 partitions

poor generalization ability. There are some floating centroids in FCM[13]. The centroids are the important points in partition space to finish the classification tasks. The regions controlled by these centroids form partitions in partition space. There are natural boundaries between these partitions. The main idea is "first clustering then classify". First, clustering algorithm is performed to get k disjoint clusters. Then, we can get the centroids in partition space by some rules in clustering algorithm. When new samples are classified, we will calculate the Euclidian distance between the mapping points in partition space and current centroids and choose the class as the new sample's label which is the minimum distance value.

C. Local coupled feed-forward neural network

The connection structure of the local coupled feed-forward neural network (LCFNN)[14][15]is same as that of Multilayer Perceptron with one hidden layer. In the local coupled feed-forward neural network, each hidden node is assigned an address in an input space, and each input activates only the hidden nodes near it. For each input, only the activated hidden nodes take part in forward and backward propagation processes.

D. MIMO CMAC neural network classifier

This study presents a cerebellar model articulation controller neural network (CMAC NN)[16]classifier. To improve the accuracies of training and generalization, the CMAC NN classifier is designed with multiple-input and multiple-output (MIMO) network topology.

E. Generalized classifier neural network

There are five layers in the generalized classifier neural network[17].They are input, pattern, summation, normalization and output layers. A smoothing parameter which is updated to converge squared error of winner neuron to global minimum is assigned for each neuron of pattern layers. GCNN uses target values for each pattern layer neuron and provides regression based effective classification.

III. METHODOLOGY

A. Mapping relationship

The training data set is changed from the original data space to the partition space by using the mapping relationship *F*. The mapping relationship *F* is showed in formula (1)

$$\vec{O} = F(\vec{I}) \tag{1}$$

where $\vec{I}$ is input vector. The output value $\vec{O}$ of *F* can form the color points in the partition space.

As we know, the feed-forward neural networks have the ability to approximate any nonlinear functions. If the number of neurons in input layer is set to the dimension of $\vec{I}$ and the number of neurons in the output layer equals the dimension of $\vec{O}$, the feed-forward neural network can play the role of *F*. A simple feed-forward neural network with *n* input neurons and *m* output neurons is showed in Fig. 2.

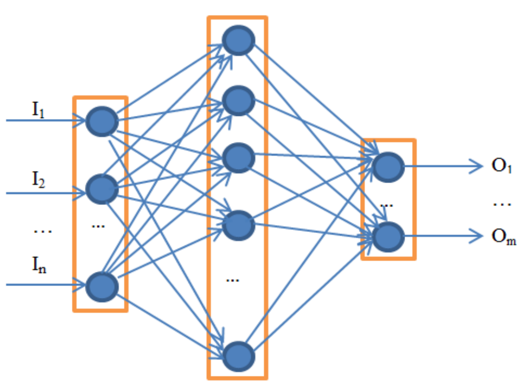

Figure 2 A feed-forward neural network having *n* input neurons and *m* output neurons.

B. Partition space

Partition space *p* is used to categorize samples by the output value of neural network *F*. There are many color points formed by the output value of *F* in the partition space. Then we can deal with those color points in one of the clustering algorithms to get some centroids. Those centroids in partition space which are the key points in the classification process are used to classify the mapping points. These centroids can divide the partition space into many irregular regions.

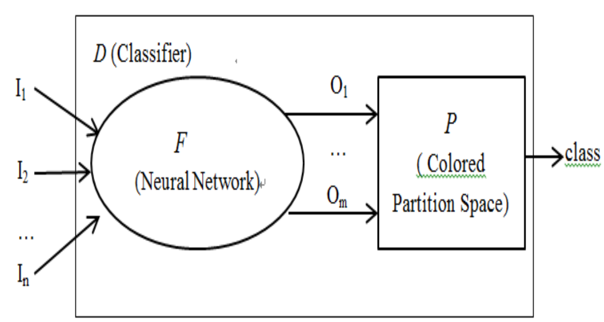

Figure 3 Model of classifier

C. The model of classifier

After the learning process, we can get the optimized neural network and the colored partition space. The neural network classifier D is composed of F and P, where P *and* F are the colored partition space and mapping relationship, respectively. Fig. 3 shows the model of classifier D.

D. K-Medoids clustering algorithm

The K-Means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. The K-Means algorithm is sensitive to outliers. When those outliers are assigned to a cluster, they can dramatically distort the mean value of the cluster. This inadvertently affects the assignment of other objects to clusters. If there are some points far from the majority of the data points, the performance of K-Means will be influenced by the outliers.

How can we modify the K-Means algorithm to diminish such sensitivity to outliers? K-Medoids clustering algorithm[18][19] is another most well-known and commonly used partitioning method except K-Means. Instead of taking the mean value of the objects in a cluster as a reference point, actual objects are picked in K-Medoids to represent the clusters. Each of clusters has an actual representative object. Each remaining object is assigned to the cluster of which the representative object is the most similar. The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object p and its corresponding representative object O. The error criterion is given by

$$E = \sum_{i=1}^{k} \sum_{p \in C_j} dis(p, O_i) \tag{2}$$

, where E is the sum of error for all subjects p in data set, k denotes the number of clusters and O_i is the representative object of cluster C_i .

A typical K-Medoids partitioning algorithm is like Algorithm 3.1.

Algorithm 3.1: K-Medoids
Input:

- k: the number of clusters.
- D: a data set containing n objects.

Output: a set of k clusters.

Method:
Step 1: arbitrarily choose k objects in D as the initial representative objects or seeds;
repeat
Step 2: assign each remaining object to the cluster with the nearest representative object;
Step 3: randomly select a non-representative object O_{random};
Step 4: compute the total cost S of swapping representative object O_j with O_{random};
Step 5: if S<0
then
swap O_j with O_{random} to form the new set of k representative objects;
until unchanged;

E. Centroids generation

A colored partition space P is a part of classifier which can be got by performing a centroids generation process. How can we get a colored space? First, the mapping relationship F will map training samples into partition space. Those mapping points in the partition space are called color points. Then, color points in partition space are divided into a fixed number of k disjoint clusters by means of K-Medoids. The center of each cluster is given by an actual representative object which has a minimum total cost S.

After getting centroids of k clusters, each of them should be labeled by a class. This is a coloring process. The coloring principle is that if color points of one class are majority among all the points belonging to a centroid, this class will color the centroid.

Algorithm 3.2: centroids generation.
Input:

- k: number of centroids
- F: A neural network
- *Data*: Training data set

Output: the colored partition space P corresponding to the given neural network F.

Method:
Step 1: map training data set *Data* to partition space using the given neural network F and form color points in partition space;
Step 2: use K-Medoids clustering algorithm to divide color points in partition space into k disjoint clusters;
Step 3: arbitrarily choose k representative objects of clusters as centroids and set current centroid as the first centroid;
Step 5: while *exist centroids which have not been colored*
do
Step 6: perform centroid coloring process in coloring principle to get the color of majority in the current centroid;
Step 7: current centroid is colored by majority's color;
Step 8: get the next centroid;
Step 9: end while
Step 10: return the colored partition space;

F. Learning

Particle swarm optimization (PSO) which is used as an optimizer in the learning process is a population based stochastic optimization technique. The main goal of learning process is to look for a best neural network and its corresponding colored partition space. The learning process is showed in Algorithm 3.3.

Algorithm 3.3: learning process.
Input:

- *m*: dimension of partition space
- *num*: number of centroids in partition space
- data: training data set

Output: the best neural network *F* and its corresponding colored partition space *P*.

Method:
Step 1: code the neural network as an individual;
Step 2: while *hasn't reached the maximum generation* do
Step 3: for (every individual);
Step 4: choose one individual and decode it to a neural network;
Step 5: perform centroids generation process to get a colored partition space corresponding to the given neural network;
Step 6: use current neural network and its corresponding colored partition space to compute the value of optimization target function as the fitness value of current individual;
Step 7: end for
Step 8: update every individual by their fitness value;
Step 9: end while
Step10: return the best neural network and its corresponding colored partition space;

G. Categorize

After getting the best neural network and its corresponding colored partition space, we can classify the new samples in testing data sets. Firstly, we map the new sample into partition space. Then, compute the Euclidean distance between the mapping points and current centroids. Finally, choose the class of the centroids which has the minimum distance value between them as the predicted class label of this new sample.

IV. Experiments and Results

In this section, some data sets are chosen to test the effectiveness of our method. In order to show the performance of this algorithm, three evaluation criterions are defined.

The first criterion is training accuracy (*TA*). If a method is good, it has a better *TA* to classify the training data into the correct class. $Num_{traintotal}$ denotes the number of total training samples. If $Num_{traincorr}$ represents the number of correctly classified samples in training data set, the formula is given by

$$TA = \frac{Num_{traincorr}}{Num_{traintotal}} \times 100 \tag{3}$$

In testing process, testing samples are used to check the performance of the neural network classifier. Then generalization accuracy (*GA*) is defined as a standard to evaluate generalization capability. Let $Num_{testtotal}$ denotes the number of total samples in test data set. Assuming that $Num_{testcorr}$ is the number of correctly classified samples in test data set, *GA* can be evaluated by the following criterion:

$$GA = \frac{Num_{testcorr}}{Num_{testtotal}} \times 100 \tag{4}$$

The third criterion is Average F-Measure (*Avg. FM*) which is a popular measure of a testing accuracy to show the performance of classification. The precision and recall of the test are two important parameters to compute the score. *TP* and *FP* denote true-positive and false-positive, respectively. Then precision *P* is the number of correct results divided by the number of all returned results. The formula is given by

$$P = \frac{TP}{TP + FP} \tag{5}$$

When *FN* represents false-negative, recall is the number of correct results divided by the number of results that should have been returned. The value of R can be calculated by the formula

$$R = \frac{TP}{TP + FN} \tag{6}$$

Now, the *Avg. FM* is defined as follows

$$vg.FM = \frac{\sum_{i=1}^{N}(2 \times \frac{P_i \times R_i}{P_i + R_i})}{N} \times 100 \tag{7}$$

The best value of *Avg.FM* is 1while the worst is 0.

In this section, four benchmark's data sets from the UCI repository are used to further demonstrate the classification performance for the FCM using our improved method. These data sets we choose in this experiment are WBCD (Wisconsin Breast Cancer Diagnostic Data Set), IRIS (Iris Data Set), VEHICLE (Statlog Vehicle Silhouettes Data Set) and CMC (Contraceptive Method Choice Data Set).Main

TABLE I CHARACTERISTICS OF DATA SETS

Data set name	No. of samples	No. of features	No. of classes
WBCD	569	30	2
IRIS	150	4	3
VEHICLE	846	18	4
CMC	1473	9	3

TABLE II Training accuracy results on these data sets

	TRADITIONAL	SOFTMAX	ECOC	FDPS	FCM	THE IMPROVED FCM
WBCD	97.48	N/A	N/A	96.88	98.54	98.73
IRIS	88.59	99.33	89.48	99.11	99.33	98.23
VEHICLE	49.13	80.00	76.95	72.04	82.80	83.68
CMC	46.50	60.35	46.47	55.48	61.17	63.35

characteristics of these data sets are depicted in Table I.

All features in these data sets are normalized by

$$f_{nor} = \frac{f_{value} - f_{min}}{f_{max} - f_{min}} \tag{8}$$

Where f_{max} is the maximum value of this feature and f_{min} is the minimum feature's value. The original value of this feature of a sample is f_{value} and f_{nor} is the normalized value.

To compare with previous FCM, we set the experimental conditions the same as its experiment. The mapping relationship F is a three-layered feed-forward neural network. The number of neurons in hidden layer is set to 15.

TABLE III Generalization accuracy results on these data sets

	TRADITIONAL	SOFTMAX	ECOC	FDPS	FCM	THE IMPROVED FCM
WBCD	93.50	N/A	N/A	92.96	94.71	95.85
IRIS	82.67	95.33	86.67	96.00	96.67	97.56
VEHICLE	49.07	71.74	73.60	67.21	78.37	79.93
CMC	46.71	52.01	45.50	48.39	54.50	55.43

TABLE IV Average F-Measure results on data sets

	TRADITIONAL	SOFTMAX	ECOC	FDPS	FCM	THE IMPROVED FCM
WBCD	92.93	N/A	N/A	92.50	94.36	94.87
IRIS	78.01	95.24	83.27	95.93	96.60	97.00
VEHICLE	37.44	68.41	69.63	66.01	77.57	78.58
CMC	30.64	43.99	29.50	44.48	52.37	52.92

Some parameters in PSO algorithm are given by the following settings: max generation=10000, population size=20, inertia weight W=1.0, learning factor C_1=1.8, C_2=1.8, the speed limitation V_{max}=3. Ten-fold cross-validation is used to verify the new method we proposed.

In addition to previous FCM, we also compare the improved FCM with other neural network classifiers, including Traditional method, SoftMax, ECOC, FDPS. Because both ECOC and SoftMax are multi-classifiers and WBCD is a binary classification data set. So we do not evaluate them on WBCD data set.

The training and testing classification results on those data sets are listed in Table II and III, respectively. Compared to traditional method and ECOC, FCM and the improved FCM have a higher training accuracy and generalization accuracy. For IRIS data sets, SoftMax, FDPS, FCM and the improved method have a close *TA* and *GA*. The performance is higher than FDPS except the *TA* on IRIS data set. The improved FCM improves lightly the classification performance.

The results of Avg.FM in Table IV further show the improved FCM is little superior to the previous method. These show that the improved FCM is capable of improving lightly the generalization ability of neural network classifier.

V. CONCLUSION

In this paper, we have proposed the idea of improving FCM using K-Medoids clustering algorithm. This method can diminish sensitivity to outliers in partition space. To show the performance of our new method, we choose four data sets from UCI repository. Through the experimental results, we can see that the performance can be improved slightly.

ACKNOWLEDGMENT

This work was supported by National Key Technology Research and Development Program of the Ministry of Science and Technology under Grant 2012BAF12B07-3. National Natural Science Foundation of China under Grant No. 61173078, No. 61203105, No. 61373054, No. 61 173079, No. 61070130, No. 81301298.Shandong Provincial Natural Science Foundation, China, under Grant No. ZR201 0FM047, No. ZR2012FQ016, No. ZR2012FM010, No. ZR2 011FZ001, No. ZR2010FQ028.Jinan Youth Science & Technology Star Project under Grant No. 2013012.

REFERENCES

[1] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.

[2] T. Wang and J. Yang, "A heuristic method for learning Bayesian networks using discrete particle swarm optimization," *Knowl. Inf. Syst.*, vol. 24, no. 2, pp. 269–281, 2010.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Comput. Biol.*, vol. 7, no. 3–4, pp. 601–620, 2000.

[4] M. W. Craven and J. W. Shavlik, “Using neural networks for data mining,” *Future Gener. Comput. Syst.*, vol. 13, no. 2, pp. 211–229, 1997.

[5] B. X. Wang and N. Japkowicz, “Boosting support vector machines for imbalanced data sets,” *Knowl. Inf. Syst.*, vol. 25, no. 1, pp. 1–20, 2010.

[6] S. Köknar-Tezel and L. J. Latecki, “Improving SVM classification on imbalanced time series data sets with ghost points,” *Knowl. Inf. Syst.*, vol. 28, no. 1, pp. 1–23, 2011.

[7] J. S. Bridle, “Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition,” in *Neurocomputing*, Springer, 1990, pp. 227–236.

[8] T. G. Dietterich and G. Bakiri, “Solving multiclass learning problems via error-correcting output codes,” *ArXiv Prepr. Cs9501101*, 1995.

[9] L. Wang, B. Yang, Z. Chen, A. Abraham, and L. Peng, “A novel improvement of neural network classification using further division of partition space,” in *Bio-inspired Modeling of Cognitive Tasks*, Springer, 2007, pp. 214–223.

[10] Y. Chen, A. Abraham, and B. Yang, “Feature selection and classification using flexible neural tree,” *Neurocomputing*, vol. 70, no. 1, pp. 305–313, 2006.

[11] B. Yang, L. Wang, Z. Chen, Y. Chen, and R. Sun, “A novel classification method using the combination of FDPS and flexible neural tree,” *Neurocomputing*, vol. 73, no. 4, pp. 690–699, 2010.

[12] L. Wang, B. Yang, Y. Chen, X. Zhao, J. Chang, and H. Wang, “Modeling early-age hydration kinetics of Portland cement using flexible neural tree,” *Neural Comput. Appl.*, vol. 21, no. 5, pp. 877–889, 2012.

[13] L. Wang, B. Yang, Y. Chen, A. Abraham, H. Sun, Z. Chen, and H. Wang, “Improvement of neural network classifier using floating centroids,” *Knowl. Inf. Syst.*, vol. 31, no. 3, pp. 433–454, 2012.

[14] J. Sun, “Local coupled feedforward neural network,” *Neural Netw.*, vol. 23, no. 1, pp. 108–113, Jan. 2010.

[15] J. Sun, “Learning algorithm and hidden node selection scheme for local coupled feedforward neural network classifier,” *Neurocomputing*, vol. 79, pp. 158–163, Mar. 2012.

[16] J.-Y. Wu, “MIMO CMAC neural network classifier for solving classification problems,” *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2326–2333, Mar. 2011.

[17] B. M. Ozyildirim and M. Avci, “Generalized classifier neural network,” *Neural Netw.*, vol. 39, pp. 18–26, Mar. 2013.

[18] H.-S. Park and C.-H. Jun, “A simple and fast algorithm for K-medoids clustering,” *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3336–3341, 2009.

[19] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques*. Morgan kaufmann, 2006.