Research of Neural Network Classifier Based on FCM and PSO for Breast Cancer Classification

Lei Zhang[1], Lin Wang[1], Xujiewen Wang[2], Keke Liu[2], and Ajith Abraham[3]

[1] Shandong Provincial Key Laboratory of Network Based Intelligent Computing, University of Jinan, 250022 Jinan, China
{Zhanglei,ise_wanglin}@ujn.edu.cn
[2] University of Jinan, 250022 Jinan, China
kevinxw@foxmail.com,dp_liukk@ujn.edu.cn
[3] Machine Intelligence Research Labs (MIR Labs), USA
ajith.abraham@ieee.org

Abstract. Breast cancer is one of the most common tumors related to death in women in many countries. In this paper, a novel neural network classification model is developed. The proposed model uses floating centroids method and particle swarm optimization algorithm with inertia weight as optimizer to improve the performance of neural network classifier. Wisconsin breast cancer datasets in UCI Machine Learning Repository are tested with neural network classifier of the proposed method. Experimental results show that the developed model improves search convergence and performance. The accuracy of classification of benign and malignant tumors could be improved by the developed method compared with other classification techniques.

Keywords: Floating Centroids Method, PSO, Neural Network, Breast Cancer Classification.

1 Introduction

Breast cancer is one of the most common malignant tumors among women. Various artificial intelligence techniques [1-6] have been used to improve the accuracy and efficiency of cancer diagnosis. In recent years, neural network has widely been applied to feature extraction, data compression, data classification, clustering etc. [7]. The learning algorithm of neural network is a supervised learning method. A classifier is usually constructed according to the character of a given datasets and analyzed the character of the given datasets. It is produced an exact model for each category data. In recent research, many classification techniques have been proposed, such as genetic algorithm [8], neural network [9-12], support vector machine [13-16], decision tree [17,18], etc.. Neural network model has been successfully used in many classification problems among these techniques.

Conducting neural network classifier, the number of parameters including input layer, hidden layer and output layer is decided by the property, category and characters of dataset. In neural network classifier, sample is mapped by neural network.

E. Corchado et al. (Eds.): HAIS 2012, Part I, LNCS 7208, pp. 647–654, 2012.

Centroid is a point in partition space and denotes the center of a class. In a conventional neural network classifier, position of centroids and the relationship between centroids and classes are set manually. In addition, number of centroids is fixed with reference to the number of classes. This fixed-centroid constraint decreases the chance to find optimal neural network. Therefore, a novel neural network classifier based on float centroids method (FCM) which removes the fixed-centroid constraint and spread the centroids throughout the partition space is proposed in our previous research [19]. However, despite the superior classification results, the training efficiency is not acceptable which limits the scope of application in practice. Therefore, this paper presents a developed method adopted particle swarm optimization algorithm with inertia weight as optimizer to improve the search convergence and performance of FCM.

The rest of the paper is organized as follows: Section 2 describes the particle swarm optimization algorithm with inertia weight. Section 3 provides the floating centroids method algorithm and the process of learning algorithm in detail. Experiment results are presented in Section 4, together with some comparisons with other classification techniques. In the end, Section 5 concludes with a brief summary of our study.

2 Particle Swarm Optimization Algorithm with Inertia Weight

The Particle Swarm Optimization (PSO) algorithm which is put forward by Kennedy and Eberhart is one of evolutionary algorithms [21]. It is an adaptive method that can be used to solve optimization problem. Conducting search uses a population of particle which corresponds to individuals. A population of particles is randomly generated initially. Each particle's position represents a possible solution point in the problem space. Each particle has an updating position vector x_i and updating velocity vector v_i by moving through the problem space. At each step, a fitness function evaluating each particle's quality is calculated by position vector x_i. The vector p_i represents the best ever position of each particle and p_g represents the best position obtained so far in the population.

For each iteration *t*, following the individual best position p_i and the global best position p_g , the velocity vector of particle *i* is updated by:

$$v_i(t+1) = v_i(t) + c_1\phi_1(p_i(t) - x_i(t)) + c_2\phi_2(p_g(t) - x_i(t)) \tag{1}$$

where c_1 and c_2 are positive constant and ϕ_1 and ϕ_2 are uniformly distributed random number in [0,1]. The velocity vector v_i is range of [$-V_{max}$, V_{max}]. Updating the velocity vector of particle by this method enables the particle to search around its individual best position and global best position. Based on the updated velocity, each particle changes its position according to the following formula:

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Based on this method, the population of particles tends to cluster together. To avoid premature convergence, the following formula can improve the velocity vector of each particle adding the inertia weight factor ω [21].

$$v_i(t+1) = \omega v_i(t) + c_1\phi_1(p_i(t) - x_i(t)) + c_2\phi_2(p_g(t) - x_i(t)) \tag{3}$$

where inertia weight factor ω is a real number. The inertia weight factor ω controls the magnitude of the velocity vector v_i.

The inertia weight factor ω is a ratio factor related to the previous velocity. The inertia weight factor ω decides the influence from the previous velocity to current velocity. The large inertia weight factor ω can make globally search and the small inertia weight factor ω will make local search [22]. The fellow method is adopted by the inertia factor ω from $\omega_{\max}$ to $\omega_{\min}$:

$$\omega = (\omega_{\max} - \omega_{\min}) * (\max iter - iter) / \max iter + \omega_{\min} \tag{4}$$

where *maxiter* is the total number of iteration, *iter* is the current iteration. Linearly decreasing the inertia factor ω from $\omega_{\max}$ to $\omega_{\min}$ can improve the search area firstly and locate the best position quickly. With the inertia weight factor ω gradually decreasing, the velocity of particle becomes slower and makes local search subtly. The convergence becomes faster owing to add the inertia weight factor.

3 The Procedure of Learning Algorithm for FCM and PSO

3.1 Floating Centroids Method

FCM [19] is an approach which has some floating centroids with class labels spread in the partition space. A sample is labeled by a certain class when the closest centroid of its corresponding mapped point is the closest centroids of this certain class. The partition space is decided by an irregular regions controlled by centroids. To get these floating centroids, training samples are mapped to partition space by neural network, and then, corresponding mapped points in partition space will be partitioned into several numbers of disjoint subsets using the K-Means clustering algorithm [20]. The floating centroids are defined as computing the cluster centers of these subsets. So, the number of centroids may be more than the number of classes. Finally, each of these centroids will be labeled by one class. If training points of one class are majority among all the points which belong to a centroid, the centroid is labeled by this class. More than one centroid can be labeled by one class.

The floating centroids method can achieve the optimization objective of neural network classifier, because it is no constraint in the number of centroids and the class label of centroids are computed by the mapped points in partition space. The floating centroids method has no fixed-centroid. The function of floating centroids method is that allocates points in the same class together as close as possible and separates points in the different classes from the partition space.

3.2 Model of Neural Network Classifier

A neural network classifier is a pair (F, P), where centroid in P have been colored. The model neural network classifier is described by Fig. 1.

If a new dataset need to be categorized, it can be mapped one point in P used by F, then predict the category according to the distance which measured by Euclidian distance. This process is illustrated in Fig.2. The dimension of partition space is set to two and the number of classes is three.

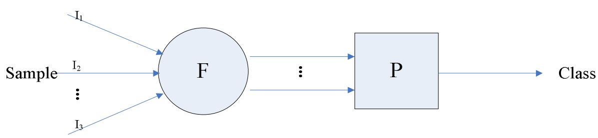

Fig. 1. Model of classifier

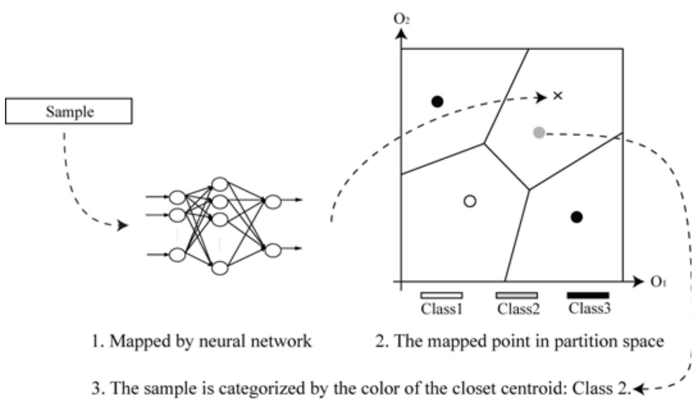

Fig. 2. Categorizing a new sample

3.3 Centroids Generation

Firstly, the training dataset is mapped to partition space using by neural network. The mapped point corresponding to training data is named colored point. The colored points in partition space are divided into *k* disjoint clusters used by K-means algorithm. Then each centroid is labeled category tag according to the principle which is that if the color points of one class are majority in all of color points, the class should color the centroid. One class can color one or more centroids.

If the number of datasets between classes is different, it should keep equitableness using weights or other solutions. One class colors a centroid by a higher probability, if the number of this class dataset is larger than others.

3.4 The Process of Learning Algorithm

The goal of learning algorithm is to get the best neural network and it corresponding colored partition space. The process of learning algorithm is described by Algorithm 1.

Algorithm 1: The process of learning algorithm
Input: User-defined parameters and training data set;
Output: Best neural network and its corresponding colored partition space;
Step 1: Code neural network to form individual according to the particle swarm optimization mentioned with inertia weight in Section 2;
Step 2: while maximum generation has not been reached do;
Step 3: for *i*=1 to the number of individuals do;
Step 4: Decode individual id to form a neural network;
Step 5: Perform centroid generation with this neural network to get colored partition space;
Step 6: Use this neural network and its corresponding colored partition space to compute the value of optimization target function as the fitness of individual *i*;
Step 7: end for;
Step 8: Perform an iteration of improved particle swarm optimization algorithm, update individuals by their fitness;
Step 9: end while;
Step 10: return the best neural network and its corresponding colored partition space.

3.5 Target Function

The optimization target function is defined as the follow formula,

$$F = \sum_{i=1}^{s} \frac{1}{1 + e^{a(1-2Z_i)}} \tag{5}$$

where Z_i is the value of point *i* sample which is mapping to the partition space. *S* is total number of samples in the training dataset and *a* is a real constant which controls tortuosity. The target function is to put points in the same class together as close as possible and keep points in the different class away from others as far as possible.

4 Experiments and Results

The criterion of testing accuracy is defined to evaluate the performance of this method. Testing accuracy (TA) which is a method to obtain better generalization capability is adopted testing data to achieve the accurate rate.

$$TA = \frac{TP_{test}}{P_{test}} * 100\% \tag{6}$$

where TP_{test} represent the number of correctly classified samples in testing dataset and P_{test} is number of total samples in testing dataset.

To test the effectiveness of the proposed method, we select the Wisconsin breast cancer datasets for the experiments which is selected from the UCI machine learning database generated by Dr. William from the University of Wisconsin Madison [24]. Breast cancer can be diagnosed benign or malignant by puncture sampling. Wisconsin breast cancer dataset consists of 569 records and 2 classes, which 357 records are benign type and 212 records are malignant type. Each record has 30 attributes including ten quantifying feature, such as diameter, perimeter, symmetry etc.

The breast cancer datasets is divided into two subsets randomly, one is selected as the training dataset and the other is the testing dataset. There are 285 records as training dataset and 284 records as testing dataset for benign and malignant type. This two type experimental data have been done respectively. One class of sample is used for the benign and the other is used for the malignant. The normal data is labeled as 1 and the abnormal data is labeled as 0.

In our experiments, a three-layered feed-forward neural network with 15 hidden neurons is adopted and the particle swarm optimization algorithm with inertia weight is selected as optimizer. This proposed method is compared with the other neural network classifiers, including Flexible Neural Tree (FNT) [23], Neural Network (NN) and Wavelet Neural Network (WNN) [25]. The parameters of PSO algorithm are important factors on searching the optimal solution. Through trials on these datasets, the parameters setting of particle swarm optimization algorithm is defined by the follow Table 1.

Table 1. Parameters for experiments

Population size	20
Generation	3000
W_{max}	0.8
W_{mix}	0.4
C_1	2
C_2	2
V_{max}	3
V_{min}	-3

Table 2 depicts the accuracy results for testing data with comparison between our method and other methods. The result indicates that our proposed method is fully capable of improving the generalization ability of neural network classifier.

Table 2. Testing accuracy (%)

Cancer type	FNT	NN	WNN	Our method
Benign	93.31	94.01	94.37	96.47
Malignant	93.45	95.42	92.96	95.70

5 Conclusions

In this paper, a novel neural network is proposed based on floating centroids method and particle swarm optimization algorithm. This approach makes sample to be mapped into neural network and partition space is used to categorize data sample. The proposed method removes the fixed-centroid constraint. An improved particle swarm optimization with inertia weight is adopted as optimizer in this paper. It increases the variety of particles and improves the performance and convergence of standard particle swarm optimization algorithm. To evaluate the performance of our proposed approach, Wisconsin breast cancer datasets from UCI machine learning repository is selected to compare the accuracy measure with FNT, NN and WNN. Experimental results illustrate that our method is more efficient than other methods in classification accuracy of breast cancer dataset.

Acknowledgements. This work was supported by the Natural Science Foundation of Shandong Province No. ZR2011FL021, the Natural Science Foundation of China No.61173078, the Science and Technology Development Program of Shandong Province No. 2011GGX10116.

References

1. Hulka, B.S., Moorman, P.G.: Breast Cancer: Hormones and Other Risk Factors. Maturitas 38(1), 103–113 (2001)
2. Corchado, E., Graña, M., Wozniak, M.: New trends and applications on hybrid artificial intelligence systems. Neurocomputing 75(1), 61–63 (2012)
3. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180(10), 2044–2064 (2010)
4. Corchado, E., Abraham, A., Carvalho, A.: Hybrid intelligent algorithms and applications. Information Sciences 180(14), 2633–2634 (2010)
5. Pedrycz, W., Aliev, R.: Logic-oriented neural networks for fuzzy neurocomputing. Neurocomputing 73(1-3), 10–23 (2009)
6. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid learning machines. Neurocomputing 72(13-15), 2729–2730 (2009)

7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
8. Platt, J., Cristianini, N., Shawe Taylor, J.: Large Margin DAGs for Multiclass Classification. In: Advances in Neural Information Processing Systems, pp. 547–553 (2000)
9. Lu, H., Rudy, S., Huan, L.: Effect data mining using neural networks. IEEE Transcation Knowledge Data Engineer. 8(6), 957–961 (1996)
10. Misraa, B.B., Dehurib, S., Dashc, P.K., Pandad, G.: A reduced and comprehensible polynomial neural network for classification. Pattern Recognition 29(12), 1705–1712 (2008)
11. Daqi, G., Yan, J.: Classification methodologies of multilayer perceptions with sigmoid activation functions. Pattern Recognition 38(10), 1469–1482 (2005)
12. Chen, Y., Abraham, A., Yang, B.: Feature selection and classification using flexible neural tree. Neurocomputing 70(1-3), 305–313 (2006)
13. Vapnik, V.N.: The nature of statistical learning theory. Springer, New York (1995)
14. Chua, Y.S.: Efficient computations for large least square support vector machine classifiers. Pattern Recognition 24(1-3), 75–80 (2003)
15. Koknar-Tezel, S., Latecki, L.J.: Improving SVM classification on imbalanced time series data sets with ghost points. Knowledge and Information Systems 28(1), 1–23 (2011)
16. Benjamin, X.W., Japkowicz, N.: Boosting support vector machines for imbalanced data sets. Knowledge and Information Systems 25(1), 1–20 (2011)
17. Qinlan, J.R.: Introduction of decision trees. Machine Learning 1(1), 81–106 (1986)
18. Wang, L., Yang, B., Chen, Y., Abraham, A., Sun, H., Chen, Z., Wang, H.: Improvement of Neural Network Classifier using Floating Centroids. Knowledge and Information Systems (2011), doi: 10.1007/s10115-011-0410-8
19. Freund, Y.: Boosting a weak learning algorithm by majority. Information Computing 121(2), 256–285 (1995)
20. Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. Applied Stat. 28(1), 100–108 (1979)
21. Eberhart, R.C., Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. In: Proc. of the Congress on Evolutionary Computation, pp. 81–86 (2001)
22. Huang, C.-P., Xiong, W.-L., Xu, B.-G.: Influnce of Inertia Weight on Astringency of Particle Swarm Algorithm and Its Improvement. Computer Engineering 34(12), 31–33 (2008)
23. Yang, B., Wang, L., Chen, Z., Chen, Y., Sun, R.: A novel classification method using the combination of FDPS and flexible neural tree. Neurocomputing 73(4-6), 690–699 (2010)
24. Blake, C.L., Merz, C.J., Newman, D.J., et al.: UCI Repository of Machine Learning Databases (2002), `http://www.ics.uci.edu/mlearn/MLRepository.html`
25. Chen, Y., Abraham, A.: Hybrid Neurocomputing for Detection of Breast Cancer. In: The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST 2005), pp. 884–892 (2005)