A Novel Improvement of Neural Network Classification Using Further Division of Partition Space

Lin Wang[1], Bo Yang[1,⋆], Zhenxiang Chen[1], Ajith Abraham[2], and Lizhi Peng[1]

[1] School of Information Science and Engineering, University of Jinan, Jinan, China
[2] Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, Norway
yangbo@ujn.edu.cn

Abstract. Further Division of Partition Space (FDPS) is a novel technique for neural network classification. Partition space is a space that is used to categorize data sample after sample, which are mapped by neural network learning. The data partition space, which are divided manually into few parts to categorize samples, can be considered as a line segment in the traditional neural network classification. It is proposed that the performance of neural network classification could be improved by using FDPS. In addition, the data partition space are to be divided into many partitions, which will attach to different classes automatically. Experiment results have shown that this method has favorable performance especially with respect to the optimization speed and the accuracy of classified samples.

Keywords: Classification, neural network, partition space, further division.

1 Introduction

Classification is an important research area in data mining. In supervised classification tasks, a classification model is usually constructed according to a given training set. Once the model has been built, it can map a test data to a certain class in the given class set. Many classification techniques including decision tree [1, 2], neural network (NN) [3], support vector machine (SVM) [4, 5], rule based classifiers systems etc. have been proposed. Among these techniques, decision tree is simple and easy to be comprehended by human beings. SVM is a new machine learning method developed on the Statistical Learning Theory. SVM is gaining popularity due to many attractive features, and promising empirical performance. SVM is based on the hypothesis that the training samples obey a certain distribution which restricts its application scope. Neural network classification, which is supervised, has been proved to be a practical approach with lots

⋆ Corresponding author.

J. Mira and J.R. Álvarez (Eds.): IWINAC 2007, Part I, LNCS 4527, pp. 214–223, 2007.

of success stories in several classification tasks. However, its training efficiency is usually a problem, which is the current focus of our research in this paper.

In conventional neural network classification, the partition space is a line segment between 0 and 1. A sample from the original data set is mapped to this line segment by the neural network. The sample will be deemed as class 0 if it is more close to 0 than to 1. Otherwise, the sample will be deemed to belong to class 1. If the partition space is divided into partitions, and the mapped sample gets close to these partitions freely, the mapping relationship (using a neural network) could be formed easily. Therefore, the performance of neural network classification including training speed and accuracy could be improved. This is the basic inspiration of our research.

Samples are mapped to the partition space by neural networks, and then partition space is assigned by the distribution of mapped samples. If neural networks are optimized, its corresponding signed partition space will be formed. Particle Swarm Optimization(PSO) is used to optimize neural network in FDPS for its predominant features. The problem, to assign the divided partitions in partition space by a certain category, once perplexed us. Very soon it was found that the category of majority in one partition should control the partition that it belongs to. Process that assigns partitions by category of majority is called color partitions. The traditional neural network classification could be regarded as a specific example of FDPS. Its partition space is divided into 2 partitions, and its dimension of partition vector is 1.

This paper is arranged as follows. Particle swarm optimization is outlined in Section 2. Section 3 illustrates the detailed method of FDPS. Section 4 outlines and discusses our experimental results followed by conclusions in Section 5.

2 Particle Swarm Optimization (PSO)

Particle swarm optimization is a population based stochastic optimization technique [6], inspired by social behavior of bird flocking or fish schooling. There are two reasons that PSO is attractive. There are few parameters to be adjusted and usually PSO achieves better results in a faster, cheaper way compared with other methods.

PSO is initialized with a population of random solutions and searches for optima by updating generations. The potential solutions, called particles, fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space that are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*. When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its *pbest* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration towards *pbest* and *lbest* locations.

3 Further Division of Partition Space (FDPS)

In this section, FDPS is explained in detail. At first, we describe how the data set is mapped by using a neural network in FDPS. We then illustrate how to divide partition space into partitions and how to distribute mapped samples over these partitions.In order to color these partitions fairly, the reason why weights of classes are needed in FDPS is explained in the Subsection 3.3. The formula for getting the weight is also illustrated. Then we narrate the details of coloring partitions which is the kernel part of the training in FDPS. After partitions are colored, neural network and its corresponding colored partition space will be used in the evaluation of training and in the classification of new samples. The whole training algorithm is described in Subsection 3.7.

3.1 Mapping Data Set

Mapping is the process to transform the training data set from original space to partition space. The mapping relation we used is the back propagation learning method for artificial neural networks. The dimension of input data set vector is defined as n and dimension of partition vector is defined as m. Every element of partition vector is mapped by an isolated neural network from the same input vector $\boldsymbol{I}$. The i th back propagation neural network structure which is used in FDPS shown in Figure 1.

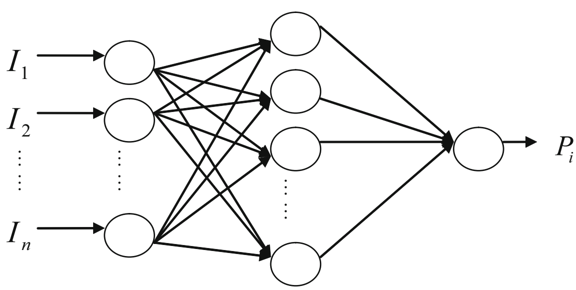

Fig. 1. The structure of i th neural network

Where the target partition space vector is $\boldsymbol{P}$. $\boldsymbol{P}_i$ is the i th element of $\boldsymbol{P}$ and $\boldsymbol{I}$ is an input data set vector. The mapping formula is:

$$P_i = NN_i\left(\boldsymbol{I}\right), i = 0...m-1 \tag{1}$$

Whole data set is mapped to partition space using (1).

3.2 Division of Partition Space

The partition space is divided into many partitions. If m is equal to 1, the partition space is one-dimensional and every partition is a line segment. If m is equal to 2, the partition space is two-dimensional and every partition is a rectangle. If m is equal to 3, the partition space is three-dimensional and every piece is cube, and so on.

$$TotalPartitionNumber = \prod_{d=1}^{m} partitionsnumber_d \tag{2}$$

$partitionsnumber_d$ is the number of partitions in d axes, $TotalPartitionNumber$ is the number of partitions in the whole partition space. Figure 2 illustrates this concept.

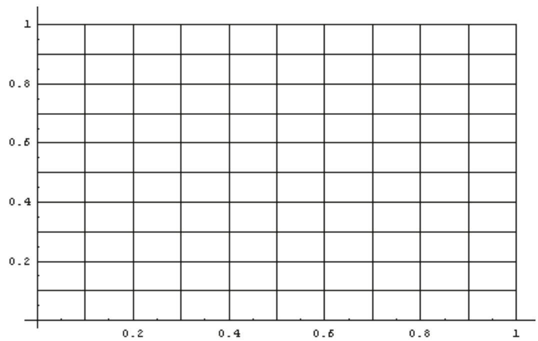

Fig. 2. Divided partition space. The m is 2, each $PiecesNumber$ is 10, and so the $TotalPiecesNumber$ is 100.

3.3 Analyzing Data Set and Computation of the Weight of Classes

It is unfair for every class data to color the partition space without using weight, because the number of each class data in the data set is different. A class will control most area in the partition space, if its number is much bigger than any other in the same data set. In order to solve the problem, a weight is needed. The weight of the first class should be smaller than second one, if the number of the first one is bigger than the second. The proportion of class c in the whole data set is defined as R_c:

$$R_c = \frac{Num_c}{Num_{total}} \tag{3}$$

Where number of samples in class c is Num_c, and Num_{total} is the total number of samples in the whole data set. The weight of class c is calculated by:

$$W_c = \frac{1}{R_c} \tag{4}$$

3.4 Coloring Partitions

When every sample in the data set is mapped to the partition space by neural networks, parts of which are used to color the partitions,is related to proportion DP. These selected mapped samples are called color points. Color points are taken out from each class in the mapped data set by the proportion DP. There are two methods to generate the sequence of color points. Order sequence and random sequence could be used for generation in one class.

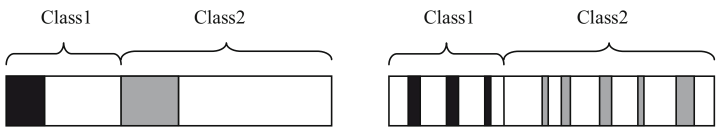

Fig. 3. This is a mapped training data set. The shadowed parts are color points with the proportion DP in each class. The left sequence is an ordered sequence. Color points are in the front position of the data set, while the other mapped samples followed after. The right sequence is random. Color points are taken out randomly in each class, but the total number of random samples are according with DP in each class.

Every partition is blank after dividing the partition space into partitions. Each of them are colored. Our goal is that different classes are to be colored with different colors. One or more partitions could be colored by one class. The principle is that if color points of one class are majority in one partition, this class will control the partition. The partition is colored by the color of this class. The partitions, which are still blank after the partition space is colored, are called unclassified partitions. Test points (training point will never drop into these pieces, if they do, the piece will be colored), which fall into these partitions are called unclassified points. The corresponding sample is unclassified. If the data set is uneven, the number of color points of each class should be multiplied by its weight. Color Algorithm in one piece is shown as;

```
Majority: =0;
For C=0 to Number of Classes-1
  Begin
  If Color points' number of class c in this partition*Weight of class C>
  Number of color points of class Majority in this partition *Weight of
  class Majority
  Then Majority: =C;
  End;
partition Color:=Color(Majority);
```

All the partitions should run this algorithm to color themselves. For example: in the training data set, the number of points of class 1 is twice as much as class 0. We calculate W_0=3 and W_1=1.5 and set each $PartitionNumber$ to 10 and m=2. So $TotalPartitionNumber$ is 100. Figure 4 illustrates the distribution of color points and the coloring of the partition space.

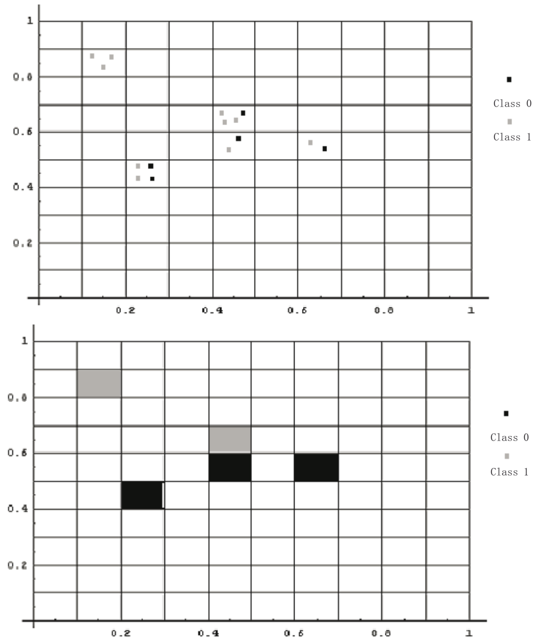

Fig. 4. The figure on top illustrates the distribution of color points. The partition space should be colored as per bottom figure.

3.5 Calculation of Correctness

The performance of neural network is to evaluated, because the network should be optimized to achieve the best performance. FDPS is supervised, and so the evaluation of neural network should be calculated at each generation. The correctness is calculated by the following formula, where L is the total number of samples of training data set:

$$Right = \begin{cases} 1, & Color(point) = Color(partition) \\ 0, & Else. \end{cases} \tag{5}$$

$$Correctness = \frac{\sum_{i=0}^{L-1} Right_i}{L} \tag{6}$$

3.6 Categorizing New Sample by Using FDPS

After training, an optimized neural network and its corresponding colored result (the colored partition space) is produced. A new sample should be mapped by this neural network and then, its category should be judged according to the color of the partition, which the corresponding mapped point of this sample belongs to. If the partition color is not blank, this sample is categorized by the corresponding class of the color. Otherwise, partition color is blank and the sample is unclassified in FDPS.

3.7 FDPS Training Algorithm

```
The weight and threshold of neural network should be coded, to form
a vector; Form a population of vectors and initialize vectors by
random constant; Analyze training data set and get the weight of
classes; Take out color points sequence from training data set
according to DP; Divide partition space into partitions; Current
generation: =0; While current generation < maximum generation do
   Begin
   For every vector of the population
      Begin
      Decode the vector, and form a neural network;
      Map training data set from original space to partition
      space by neural network;
      Color partitions and record the result;
      Calculate the correctness as the vector's fitness;
      End;
    Optimize vectors using PSO;
    Current generation: = Current generation+1;
    End.
The goal is to get the elitist neural network and its corresponding
colored results.
```

4 Experiments and Results

In order to evaluate the performance of this algorithm, four criterions are defined.

Training Accuracy (TA), a method to adapt training data to achieve better TA.

$$TA = \frac{NumberOfCorrectlyClassifiedSamplesInTrainingDataSet}{NumberOfTotalSamplesInTrainingDataSet} \quad (7)$$

Generalization Accuracy (GA), a method to obtain better generalization capability.

$$GA = \frac{NumberOfCorrectlyClassifiedSamplesInTestDataSet}{NumberOfTotalSamplesInTestingDataSet} \quad (8)$$

Accuracy of Classified Samples (ACS), displays the classification ability of the method on the data which is to be categorized. On the condition that a sample could be classified by a method, higher values of ACS ensures higher classification accuracy.

$$ACS = \frac{NumberOfCorrectlyClassifiedSamplesInTestDataSet}{NumberOfClassedSamplesInTestingDataSet} \quad (9)$$

Proportion of Unclassified Samples (PUS) illustrate unclassified proportion of data sets. Lower PUS represents more samples to be be categorized. The PUS in traditional neural network classification is 0, because all of the samples are categorized by this method. In FDPS, PUS is not 0 for the reason that some samples are mapped to unclassified partitions.

$$PUS = \frac{NumberOfUnclassedSamplesInTestingDataSet}{NumberOfTotalSamplesInTestingDataSet} \tag{10}$$

Experiments need many trails for one data set. So Mean (M) and Standard Deviation (SD) are also needed for performance evaluation. Many data sets have been used for testing and evaluation of FDPS procedure. This paper uses the breast cancer data as the representative data set. Breast cancer is the most common cancer in women in many countries. Most breast cancers are detected as a lump/mass on the breast, or through self examination or mammography [7,8,9]. The Wisconsin breast cancer data set has 30 attributes and 569 instances of which 357 are of benign and 212 are of malignant type. The data set is randomly divided into a training data set and a test data set. The first 285 data is used for training and the remaining 284 data is used for testing. Binary classification is adopted in this research.

Experiments were conducted to evaluate TA, GA and ACS and the training speed. Ten trails were conducted and the mean and standard deviation is reported. We used a three-layered back propagation neural network with 30 hidden neurons. Parameters used for this data set are: DP=0.5, m=2, $PartitionNumber_0$ =10, $PartitionNumber_1$=10, Max Generation=1000, Population Size=50,ϕ_1= 0.05,ϕ_2= 0.05 and VMAX=1.2. Table 1 and Figure 5 illustrate the overall performance of the FDPS.

Some accuracy results are shown in Table 1. The average performance of both TA and ACS of FDPS is obviously higher than the traditional method. But, average GA is lower in FDPS, at the same time SD of GA is higher. The maximum GA in FDPS is higher than maximum GA in the traditional method, while the minimum GA in FDPS is lower than the traditional method.

Table 1. Performance results for breast cancer data

	Traditional Method			FDPS			
	TA	GA	ACS	TA	GA	ACS	PUS
1	93.21%	93.21%	93.21%	94.28%	94.64%	94.64%	0%
2	94.28%	93.93%	93.93%	93.57%	87.50%	87.5%	0%
3	92.85%	90.35%	90.35%	94.64%	91.43%	92.42%	1.07%
4	93.57%	91.79%	91.79%	95.00%	93.21%	93.21%	0%
5	92.85%	92.14%	92.14%	95.35%	90.72%	91.70%	1.07%
6	93.21%	91.79%	91.79%	94.64%	90.00%	90.97%	1.07%
7	93.21%	88.21%	88.21%	95.00%	93.21%	93.21%	0%
8	93.21%	93.21%	93.21%	94.64%	89.64%	89.64%	0%
9	94.28%	89.64%	89.64%	94.64%	89.64%	89.96%	0.36%
10	91.42%	92.5%	92.5%	96.07%	93.57%	94.24%	0.71%
M	93.21%	**91.68%**	91.68%	**94.78%**	91.36%	**91.75%**	0.43%
SD	0.81%	1.78%	1.78%	0.66%	2.25%	2.25%	0.50%

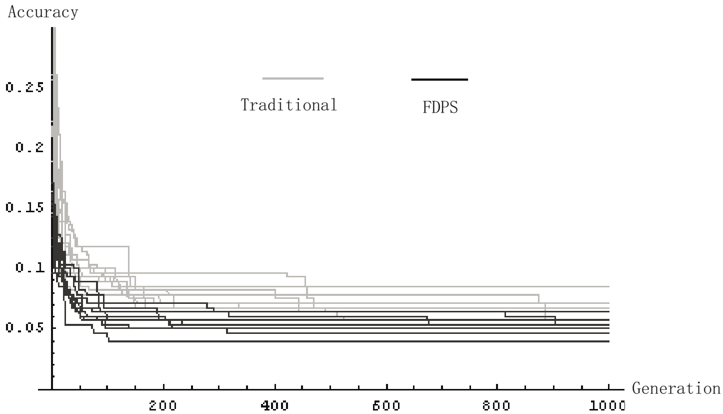

Fig. 5. Speed of convergence with Y axis representing (1.0 - TA)

Figure 5 illustrates that FDPS performed very well and has converged within the first 400 generations. During the training process, accuracy of FDPS is optimized faster than the traditional method. The final accuracy of FDPS is better than traditional method, which clearly means that the FDPS method aids the neural network learning process. It has faster optimization speed than the direct approach because mapping of neural network in FDPS is flexible and unrestricted.

5 Conclusions

This paper proposes novel technique to improve traditional neural network classification. This technique which maps data sample freely and easily is based on further division of partition space. From the experiment results, it is evident that performance measures and optimization speed of FDPS is better and faster than the traditional method. There are still some limitations in FDPS. The mean of GA is lower in FDPS, notwithstanding that the standard deviation is larger than the traditional method (leading to the best GA in FDPS is higher). More experiments on different data sets are required to better analyze FDPS's performance.

Acknowledgments. This research was supported by the National Natural Science Foundation of China under Grant No.60573065; the Natural Science Foundation of Shandong Province of China under Grant No.Z2006G03. Ajith Abraham is supported by the Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, appointed by The Research Council of Norway, and funded by the Research Council, NTNU and UNINETT.

References

1. J. R. Qinlan: Introduction of decision trees. Machine Learning.**1**(1986)86-106.
2. Freund Y:Boosting a weak learning algorithm by majority.Information Computation.**121**(1995)256-285.
3. Lu Hongjun, Setiono Rudy, Liu Huan: Effect data mining using neural networks. IEEE Transaction on knowledge and data engineering. **8**(1996)957-961.
4. B. E. Boser, I. M. Guyon, V. N. Vapnik:A training algorithm for optimal margin classifiers.Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, ACM Press.(1992)144-152.
5. V. N. Vapnik:The Nature of Statistical Learning Theory.Springer Verlag.(1995).
6. J. Kennedy and R. C. Eberhart:A new optimizer using paritcle swarm theory in Proceeding of the Sixth Int. Symposium on Micromachine and Human Science, Nagoya, Japan.(1995)39-43.
7. DeSilva, C.J.S. et al.: Artificial Neural networks and Breast Cancer Prognosis.The Australian Computer Journal. **26**(1994)78-81.
8. Shieu-Ming Chou, Tian-Shyug Lee, et al.:Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines.Expert Systems with Applications.**27**(2004)133-142.
9. Ravi Jain and Ajith Abraham, A Comparative Study of Fuzzy Classifiers on Breast Cancer Data, Australiasian Physical And Engineering Sciences in Medicine, Australia, **27** (4), 2004, pp. 147-152.