

Incorporating Fuzzy Membership Functions into Evolutionary Probabilistic Neural Networks

V. L. Georgiou, Ph. D. Alevizos, M. N. Vrahatis¹

Computational Intelligence Laboratory (CI Lab), Department of Mathematics,
University of Patras Artificial Intelligence Research Center (UPAIRC),
University of Patras, GR-26110 Patras, Greece

Received 15 July, 2006; accepted in revised form 31 July, 2006

Abstract: In this contribution a new supervised classification model is proposed, namely the Fuzzy Evolutionary Probabilistic Neural Network (FEPNN). The proposed model incorporates a fuzzy class membership function into the recently proposed Evolutionary Probabilistic Neural Network (EPNN). EPNN employs an evolutionary algorithm, namely the Particle Swarm Optimization (PSO), for the selection of the spread parameters and prior probabilities of Probabilistic Neural Networks. FEPNN combines efficient and effective evolutionary algorithms as well as techniques from fuzzy set theory. This combination provides an adequate model that achieves similar and superior performance than the well known and widely used Feed Forward Neural Networks (FNNs). FEPNN is applied to a credit card approval task with promising results.

Keywords: Probabilistic Neural Networks, Fuzzy Membership Functions, Particle Swarm Optimization, Spread Parameters

Mathematics Subject Classification: 92B20, 90C70, 65K10

1 Introduction

Computational intelligence methods have been developed rapidly during the last years. A supervised classification model which combines statistical methods and efficient evolutionary algorithms is the recently proposed *Evolutionary Probabilistic Neural Network* (EPNN) [1]. Specifically, EPNN is based on Probabilistic Neural Network (PNN) introduced by Specht [2]. PNNs have been widely used in several areas of science with promising results [3, 4, 5]. They are based on discriminant analysis [6] and incorporate the Bayes decision rule for the final classification of an unknown case. In order to estimate the Probability Density Function (PDF) of each class, the Parzen window estimator or in other words the kernel density estimator is used [7]. For the selection of the spread parameters of PNN's kernels the Particle Swarm Optimization (PSO) algorithm [8, 9] is employed.

In this contribution an extension of the EPNN is proposed which incorporates a Fuzzy Membership Function (FMF) proposed by Keller and Hunt [10]. This function describes the degree of certainty that a given datum belongs to each one of the predefined classes. The FMF provides a way of weighting all the training samples so that an even better classification accuracy can be achieved. The proposed model is applied on a well-known and widely tested data set for the prediction of the approval or not of a credit card to a bank customer [11]. The obtained results are compared to those obtained by FNNs presented in Proben1 [12].

¹Corresponding author: e-mail: vrahatis@math.upatras.gr, Phone: +30 2610 997374, Fax: +30 2610 992965

2 Background Material

For completeness purposes, let us briefly present the necessary background material. PNN is a neural network implementation of kernel discriminant analysis which incorporates the Bayes decision rule and the non-parametric density function estimation of a population according to Parzen [7]. The training procedure of PNN is quite simple and requires only a single pass of the patterns of the training data which has as a result a short training time. The architecture of a PNN always consists of four layers: the *input layer*, the *pattern layer*, the *summation layer* and the *output layer* [1, 2].

Let p be the dimension of sample vectors and K the number of classes present in the dataset. An input feature vector, $\mathbf{X} \in \mathbb{R}^p$, is applied to the p input neurons of PNN and is passed to the pattern layer. The pattern layer is fully interconnected with the input layer and is organized into K groups of neurons. Each group of neurons in the pattern layer consists of N_k neurons, where N_k is the number of training vectors that belong to the class k , $k = 1, 2, \dots, K$. The i th neuron in the k th group of the pattern layer computes its output using a kernel function. The kernel function is typically a Gaussian kernel function of the form:

$$f_{ik}(\mathbf{X}) \propto \exp\left(-\frac{1}{2}(\mathbf{X} - \mathbf{X}_{ik})^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{X} - \mathbf{X}_{ik})\right), \quad (1)$$

where $\mathbf{X}_{ik} \in \mathbb{R}^p$ is the center of the kernel and $\boldsymbol{\Sigma}_k$ is the matrix of spread (smoothing) parameters of the kernel. In fact \mathbf{X}_{ik} is the i th sample vector of the k th group of the training set.

The summation layer comprises K neurons and each one estimates the conditional probability of its class given an unknown vector \mathbf{X} :

$$G_k(\mathbf{X}) \propto \sum_{i=1}^{N_k} \pi_k f_{ik}(\mathbf{X}), \quad k = 1, 2, \dots, K, \quad (2)$$

where π_k is the prior probability of class k , $\sum_{k=1}^K \pi_k = 1$. Thus, a vector \mathbf{X} is classified to the class that has the maximum output of the summation neurons.

For the estimation of the spread matrix $\boldsymbol{\Sigma}_k$ as well as the prior probabilities π_k , PSO algorithm is used. PSO is a stochastic population-based optimization algorithm [8] and its concept is to exploit a population of individuals to synchronously probe promising regions of the search space. Here we use the PSO with constriction factor. For details we refer to [9]. The obtained by PSO values minimize the misclassification proportion on the whole training set. It is assumed that each class has its own matrix of spread parameters $\boldsymbol{\Sigma}_k = \text{diag}(\sigma_{1k}^2, \dots, \sigma_{pk}^2)$, $k = 1, 2, \dots, K$.

Moreover, we construct a smaller training set from each class by using the well-known and widely used K -medoids clustering algorithm [13] on the training data of each class. The extracted medoids from each class are used as centers for the PNN's kernels, instead of using all the available training data. This results into a much smaller PNN architecture. The number of medoids that were extracted from each class was only the 5% of the size of each class. Thus, the pattern layer's size of the proposed PNN is about twenty times smaller than the corresponding PNN which utilizes all the available training data.

3 The Proposed Approach

A desirable property of a supervised classification model is the ability to adjust the impact of each training sample vector to the final decision of the model. In other words, vectors of high uncertainty about their class membership should have less influence on the final decision of the model, while vectors of low uncertainty should affect more the model's decision. A way of obtaining this

desirable property is to incorporate a Fuzzy Membership Function (FMF) into the model. Among the large variety of classification models we chose the EPNN due to its simplicity, effectiveness and efficiency [1] and we have incorporated the FMF proposed in [10] for weighting the pattern neurons of the EPNN. We call this composite model *Fuzzy Evolutionary Probabilistic Neural Network* (FEPNN).

Next, let us further analyze the proposed model. To this end, let \mathbf{X}_{ik} , $i = 1, 2, \dots, N_k$, $k = 1, 2, \dots, K$ be a training sample vector that belongs to the class k . Since we are dealing with a two-class classification problem, we consider $K = 2$. Suppose further that $u(\mathbf{X}) \in [0, 1]$ is a fuzzy membership function, then we define:

$$u(\mathbf{X}_{ik}) \equiv u_{ik} = 0.5 + \frac{\exp((-1)^k [d_1(\mathbf{X}_{ik}) - d_2(\mathbf{X}_{ik})] f/d) - \exp(-f)}{2(\exp(f) - \exp(-f))}, \quad (3)$$

where for $k = 1, 2$, M_k is the mean vector of class k , $d_k(\mathbf{X}) = \|\mathbf{X} - M_k\|$, $d = \|M_1 - M_2\|$ and f is a constant that controls the rate at which memberships decrease towards 0.5.

In conclusion, our approach consists of the following steps:

Step 1: Using Relation (3) compute the values u_{ik} .

Step 2: Using Relation (1) and Relation (3) compute for $i = 1, \dots, N_k$ and $k = 1, 2$ the new pattern neurons outputs $f'_{ik} = u_{ik} f_{ik}$

Step 3: Using Relation (2) and the arg max rule [1] obtain the final classification.

4 Experimental Results

We have applied the proposed model FEPNN to the Card dataset from the UCI data repository [11] according to Proben1 specifications [12]. This dataset is used for the prediction of the approval or non approval of a credit card to a customer. Each vector represents a real credit card application and the output describes whether the bank granted the credit card or not. There are 690 instances with 51 inputs. The obtained results are compared with the corresponding ones from Proben1's FNNs. Each permutation of the dataset was applied to the FEPNN for 50 times.

Table 1: Classification accuracy of the models on Card dataset

dataset	FEPNN mean	FEPNN s.d.	FNN mean	FNN s.d.
Card1	86.87	0.95	86.36	0.85
Card2	82.19	0.83	80.77	0.80
Card3	84.45	2.70	82.64	1.61
Average	84.50		83.26	

In Table 1 the mean classification accuracy and its standard deviation (s.d.) of the two models is presented. FEPNN's mean accuracy (84.50%) is superior to the corresponding FNN's (83.26%). The s.d. of the accuracy is similar between FEPNN and FNN. Performing a corrected t -test [14] for each permutation of the data, there is a statistically significant superiority of FEPNN in Card2 (p -value = 0.005). In Card1 and Card3 there is no statistical significance between FEPNN and FNN (p -value = 0.350 and 0.172 respectively). It seems that FEPNNs achieve similar or superior classification accuracy compared with FNNs.

5 Conclusion

In this contribution a new supervised classification model is proposed, namely the fuzzy evolutionary probabilistic neural network. The proposed model incorporates a fuzzy membership function

into the evolutionary probabilistic neural network. The employed fuzzy membership function gives less impact to vectors with high uncertainty about their class membership and more impact to vectors with low uncertainty. Fuzzy evolutionary probabilistic neural network is applied to a credit card approval data set and is compared with feedforward neural networks with encouraging results.

Acknowledgment

We thank European Social Fund (ESF), Operational Program for Educational and Vocational Training II (EPEAEK II) and particularly the Program IRAKLEITOS for funding the above work.

References

- [1] V. L. Georgiou, N. G. Pavlidis, K. E. Parsopoulos, Ph. D. Alevizos, and M. N. Vrahatis, New self-adaptive probabilistic neural networks in bioinformatic and medical tasks, *International Journal on Artificial Intelligence Tools*, **15**(3) 371–396(2006).
- [2] D. F. Specht, Probabilistic neural networks, *Neural Networks*, **1**(3) 109–118(1990).
- [3] C. J. Huang, A performance analysis of cancer classification using feature extraction and probabilistic neural networks, In *Proceedings of the 7th Conference on Artificial Intelligence and Applications*, pages 374–378, 2002.
- [4] J. Guo, Y. Lin, and Z. Sun, A novel method for protein subcellular localization based on boosting and probabilistic neural network, In *Proceedings of the 2nd Asia-Pacific Bioinformatics Conference (APBC2004)*, pages 20–27, Dunedin, New Zealand, 2004.
- [5] T. Ganchev, D. K. Tasoulis, M. N. Vrahatis, and N. Fakotakis, Generalized locally recurrent probabilistic neural networks with application to text-independent speaker verification, *Neurocomputing*, to appear (2006).
- [6] J. D. Hand, *Kernel Discriminant Analysis*. Research Studies Press, Chichester, 1982.
- [7] E. Parzen, On the estimation of a probability density function and mode, *Annals of Mathematical Statistics*, **3** 1065–1076(1962).
- [8] J. Kennedy and R.C. Eberhart, Particle swarm optimization, In *Proceedings IEEE International Conference on Neural Networks*, volume IV, pages 1942–1948, Piscataway, NJ, 1995.
- [9] K. E. Parsopoulos and M. N. Vrahatis, Recent approaches to global optimization problems through particle swarm optimization, *Natural Computing*, **1**(2–3) 235–306(2002).
- [10] J. M. Keller and D. J. Hunt, Incorporating fuzzy membership functions into the perceptron algorithm, *IEEE Trans. Pattern Anal. Machine Intell.*, **7**(6) 693–699(1985).
- [11] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz, UCI repository of machine learning databases, 1998.
- [12] L. Prechelt, Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe, 1994.
- [13] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley and Sons, New York, 1990.
- [14] R. R. Bouckaert and E. Frank, Evaluating the replicability of significance tests for comparing learning algorithms, In *Proc Pacific-Asia Conference on Knowledge Discovery and Data Mining, LNAI 3056*, pages 3–12, Sydney, Australia, 2004.