Evolutionary Bayesian Probabilistic Neural Networks

V. L. Georgiou[1], **S. N. Malefaki** [2], **Ph. D. Alevizos**[1], and **M. N. Vrahatis**[1*]

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

[2] Department of Statistics and Insurance Science, University of Piraeus, GR-18534 Piraeus, Greece

Key words Probabilistic Neural Networks, Bayesian Modelling, Spread Parameters, Particle Swarm Optimization

Subject classification 92B20, 62F15, 65K10

A well-known and widely used model for classification and prediction is the Probabilistic Neural Network (PNN). PNN's performance is influenced by the kernels' spread parameters so recently several approaches have been proposed to tackle this problem. The proposed approach is a combination of two well known methods applied to PNNs. First, it incorporates a Bayesian model for the estimation of PNN's spread parameters and then an evolutionary optimization algorithm is used for the proper weighting of PNN's outputs. The Particle Swarm Optimization (PSO) algorithm is used for a better estimation of PNN's prior probabilities. The new model is called Evolutionary Bayesian Probabilistic Neural Network (EBPNN). Furthermore, a different kernel function, namely the Epanechnikov kernel, is used besides the typical Gaussian kernel. The above approach is applied to two biomedical applications with encouraging results and is compared with Feed-Forward Neural Networks.

1 Introduction

Recently a rapid development of classification models has taken place. One of the models used for supervised classification purposes is the Probabilistic Neural Network (PNN) [1]. PNNs have been used to several bioinformatics and medical informatics applications [2, 3] as well as to other areas of science [4, 5] with promising results.

PNNs are supervised classification and prediction models that classify an unknown input vector $\mathbf{X}$ into K predefined classes. They are closely related to the well-known discriminant analysis [6] using kernel functions for the estimation of the Probability Density Function (PDF) of each class [7]. The PDF of each class is weighted by its prior probability. This model is introduced by Specht in a neural network framework. In our contribution, a Bayesian model is incorporated for the estimation of the spread parameters of PNN's kernels [8] and the Particle Swarm Optimization (PSO) is used for the weighting of PNN's kernels [9, 10, 11]. We named this model Evolutionary Bayesian Probabilistic Neural Network (EBPNN).

The proposed model is applied to two biomedical data sets from UCI data repository [12]. The implementation has been made according to Proben1 specifications [13] and the obtained results are compared with the corresponding results obtained by FNNs.

2 Background Material

For completeness purposes, let us briefly present some background material. PNN was introduced by Specht [1] as 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

* Corresponding author: e-mail: vrahatis@math.upatras.gr Phone: +302610997374, Fax: +302610992965

result a short training time. The training procedure is just the construction of PNN from the available data. The structure of a PNN has always four layers; the *input layer*, the *pattern layer*, the *summation layer*, and the *output layer* [1, 14]. An input feature vector, $\mathbf{X} \in \mathbb{R}^p$, is applied to the p input neurons and is passed to the pattern layer. The pattern layer is fully interconnected with the input layer. The pattern layer is organized into K groups, where K is the number of classes present in the data set. 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 class k, $k = 1, \ldots, K$. The ith neuron in the kth group of the pattern layer computes its output using a kernel function. The kernel function is typically a Gaussian kernel function of the form:

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

where $\mathbf{X}_{ik} \in \mathbb{R}^p$ is the center of the kernel and $\mathbf{\Sigma}_k$ is the matrix of spread (smoothing) parameters of the kernel. Besides the Gaussian kernel, several other kernels can be used such as the Epanechnikov kernel of the form:

$$f_{ik}(\mathbf{X}) = \max\left\{1-\frac{1}{2\kappa^2}\left(\mathbf{X}-\mathbf{X}_{ik}\right)^\top \mathbf{\Sigma}_k^{-1}\left(\mathbf{X}-\mathbf{X}_{ik}\right), 0\right\}, \tag{2}$$

where κ is a known parameter [15]. This results into a faster implementation, since there is no need to calculate time-consuming exponential functional values.

Only K neurons comprise the summation layer and each one estimates the conditional probability of each class given the unknown vector $\mathbf{X}$:

$$G_k(\mathbf{X}) \propto \sum_{i=1}^{N_k} \pi_k f_{ik}(\mathbf{X}), \quad k \in \{1, \ldots, K\}, \tag{3}$$

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

In order to obtain a faster version of the PNN, only a part of the training data set is used instead of the whole training data set. We can obtain such a training set either by randomly sampling from the available data or by finding some "representatives" of the training data through a clustering technique. We created a small training set from each class by using the well-known and widely used K-medoids clustering algorithm [16], 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

One of the PNN's drawbacks is the influence of its classification accuracy by the spread parameters of its kernels. It is assumed that each class has its own matrix of spread parameters $\mathbf{\Sigma}_k = \mathrm{diag}(\sigma_{1k}^2, \ldots, \sigma_{pk}^2)$, $k = 1, \ldots, K$. For the estimation of the spread parameters, a Bayesian model has been proposed [8]. We consider the following two–parameter Bayesian model for each dimension of the data in each one of the K classes, since it is assumed that the matrix of spread parameters is diagonal:

$$\begin{aligned} \mathbf{X}_{ik} &\overset{\text{iid}}{\sim} \mathcal{N}_p(\mu_k, \mathbf{\Sigma}_k) \quad i = 1, \ldots, N_k, \\ \mu_{jk} &\sim \mathcal{N}(0, 1), \\ \tau_{jk} &\sim \mathcal{G}(\alpha, \beta), \quad j = 1, \ldots, p, \end{aligned}$$

where $\tau_{jk} = \sigma_{jk}^{-2}$ and $\alpha, \beta > 0$ are known parameters.

It is assumed that $\mathbf{X}_{ik}$ are conditionally independent given μ_k, τ_{jk} and μ_k as well as τ_{jk} are themselves independent. The joint posterior distribution of μ_{jk} and τ_{jk} is:

$$\pi(\mu_{jk}, \tau_{jk} | \mathbf{X}_{jk}) \propto \tau^{N_k/2+\alpha-1} \exp\left(-\tau_{jk}\left(\frac{\sum_{i=1}^{N_k}\left(X_{ijk}-\mu_{jk}\right)^2}{2}+\beta\right)-\frac{\mu_{jk}^2}{2}\right),$$

Table 1 Classification accuracy of the models

Dataset	EBPNN Mean	EBPNN St. Dev.	FNN Mean	FNN St. Dev.
Heart1	80.04	0.72	80.11	2.27
Heart2	82.83	0.27	82.12	1.57
Heart3	77.48	0.39	76.57	1.29
WBCD1	98.47	0.38	98.53	0.60
WBCD2	95.98	0.31	95.48	0.70
WBCD3	95.93	0.23	96.63	0.71

In order to estimate τ_{jk}, we use Gibbs sampler [17], with transition kernel the product of the full conditional for μ_{jk} and τ_{jk}, which produces a Markov chain with stationary distribution the posterior of them. In this model, conjugated prior distributions have been chosen so that the full conditional distributions are given in closed form.

In order to further improve the PNN's performance, a new way of weighting the kernels' outputs is proposed. This weighting technique is achieved by an evolutionary optimization algorithm, namely Particle Swarm Optimization (PSO). PSO is a stochastic, population–based optimization algorithm [9, 18] and the concept of the algorithm is to exploit a population of individuals to synchronously probe promising regions of the search space. In this context, the population is called a *swarm* and the search points are called *particles*. Each particle moves with an adaptable velocity within the search space, and retains in a memory the best position it ever encountered. At every iteration, this best position is communicated to other particles of the swarm. Here we use the PSO with constriction factor. For details we refer to [10, 11]. So, instead of estimating the prior probability of each class directly from the training data, we let PSO to find promising values for the weighting of the PNN's kernels with respect to the classification accuracy. The obtained values minimize the misclassification proportion on the whole training set.

4 Experimental Results

The proposed model has been applied to two biomedical problems from the UCI repository. The first data set is the "Heart Disease" and its aim is to predict whether at least one of the four major vessels of the heart is reduced in diameter by more than 50%. There are 35 inputs and 920 instances. In the second data set, namely "Wisconsin Breast Cancer Database" (WBCD), the aim is to predict whether a tumour in the breast is benign or malignant. There are 9 continuous inputs and 699 instances.

EBPNN is applied to each data set for 50 times. For the Heart data set a Gaussian kernel function obtained better results while in the WBCD an Epanechnikov kernel function achieved a better performance. In Table 1 the mean classification accuracy and its standard deviation are presented for the proposed EBPNN together with the results of Proben1's FNN [13] for the two considered problems. The mean performance of the EBPNN in the Heart data set is slightly superior than the performance of the FNN (80.12 to 79.6). For the WBCD, EBPNN as well as FNN achieved an equivalent mean performance (96.79 to 96.88). It should be noted here that the standard deviation of the classification accuracy is much lower on the EBPNN compared with the FNN's. This seems that the EBPNN is a robust model with adequate performance.

5 Conclusion

In this contribution a new model is proposed for supervised classification tasks. The Evolutionary Bayesian Probabilistic Neural Network incorporates a two–parameter Bayesian model for the estimation of the PNN's spread parameters. Moreover, the Particle Swarm Optimization algorithm is employed for the weighting of PNNs' kernels. In other words, PSO estimates the prior probability of each class. Furthermore, an Epanechnikov kernel function is used in the EBPNN for faster performance. The proposed model has been applied to two biomedical applications with promising results. EBPNN achieved an equivalent performance compared with the FNN, but it seems that EBPNN is faster and more robust.

Acknowledgements 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] D. F. Specht. Probabilistic neural networks. Neural Networks **1**(3), 109–118 (1990).
[2] 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, pp. 374–378 (2002).
[3] 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), pp. 20–27 (Dunedin, New Zealand, 2004).
[4] T. Ganchev, D. K. Tasoulis, M. N. Vrahatis, and N. Fakotakis. Locally recurrent probabilistic neural networks with application to speaker verification. GESTS International Transaction on Speech Science and Engineering **1**(2), 1–13 (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, (2006), to appear.
[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] V. L. Georgiou and S. N. Malefaki. Incorporating Bayesian models for the estimation of the spread parameters of probabilistic neural networks with application in biomedical tasks. In Proceedings of the Int. Conf. on Statistical Methods for Biomedical and Technical Systems, pp. 305–310, (Limassol, Cyprus, 2006).
[9] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In Proceedings Sixth Symposium on Micro Machine and Human Science, pp 39–43, (Piscataway, NJ, 1995).
[10] 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).
[11] K. E. Parsopoulos and M. N. Vrahatis. On the computation of all global minimizers through particle swarm optimization. IEEE Transactions on Evolutionary Computation **8**(3), 211–224 (2004).
[12] P.M. Murphy and D.W. Aha, UCI Repository of machine learning databases, Irvine, CA: University of California, Department of Information and Computer Science, (1994).
[13] L. Prechelt. Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94 (Fakultät für Informatik, Universität Karlsruhe, 1994).
[14] 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).
[15] C. G. Looney. A fuzzy classifier network with ellipsoidal Epanechnikov functions. Neurocomputing **48**, 489–509 (2002).
[16] L. Kaufman and P. J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis (John Wiley and Sons, New York, 1990).
[17] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattn. Anal. Mach. Intel.* **6**, 721–741 (1984).
[18] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In Proceedings IEEE International Conference on Neural Networks Vol. 4, pp 1942–1948 (Piscataway, NJ, 1995).