RBF NETWORKS FOR DENSITY ESTIMATION

Lucia Sardo *Josef Kittler*
Department of Electronic & Electrical Engineering
University of Surrey, Guildford, Surrey GU2 5XH, United Kingdom
Tel: +44 1483 300800; fax: +44 1483 34139
e-mail: L.Sardo@ee.surrey.ac.uk - J.Kittler@ee.surrey.ac.uk

ABSTRACT

A non-parametric probability density function (pdf) estimation technique is presented. The estimation consists in approximating the unknown pdf by a network of Gaussian Radial Basis Functions (GRBFs). Complexity analysis is introduced in order to select the optimal number of GRBFs. Results obtained on real data show the potentiality of this technique.

1 INTRODUCTION

Non-parametric pdf estimation is universally reputed superior to the parametric approach when no hypothesis about the structure of the data can be formulated. The two most established technique, i.e. Parzen window estimator and k-nearest-neighbour estimator, are both computationally demanding. Our aim is the design of a pdf estimator, and then a classifier, that is *local* and *simple* at the same time. We use simple with the meaning that the complexity of the final classifier does not grow linearly with the training set size, as the above mentioned estimator do. Hence the idea is to use the locality property of GRBFs without burdening the estimator with many of them.

The first example of such networks reported in the literature are the feed-forward RBF net used by Renals *et al.*([5]) to describe the transition probabilities of a Hidden Markov Model and the "semiparametric" approach to density estimation by Tråvén ([9]).

We approximate the pdf as a Gaussian mixture, that is using an GRBF NN with m hidden neurons. We try to use as few hidden neurons as possible, still retaining a good performace of the final classifier. This technique, that is referred as Maximum Penalized Likelihood (MPL) in the paper, consists of a modification of the criterion function used to train the network. In general any NN is trained in such a way so as to maximize a criterion of optimality. This criterion for MPL is the Kullback-Leibler (KL) distance between the true and approximated pdf *and* the complexity of the network. This complexity term is added because a criterion based only on a measure of goodness-of-fit between data and the model is inappropriate. It favours complex models with large number of parameters that can be adjusted to fit the data to any desired accuracy ([8]).

In the following section we briefly decsribe the MPL estimator. In Section 3 the classification performance of a GRBF based classifier built using the MPL estimator is assessed. Finally the last section is devoted to some comments and conclusions.

2 MPL ESTIMATOR

Our strategy is to approximate the pdf as a Gaussian mixture. This is done by using a GRBF NN with m hidden neurons as illustrated in Figure 1.

$$p(\mathbf{x}) = \sum_{j=1}^{m} W_j G(\mathbf{x};\ \mu_j, \sigma_j) \tag{1}$$

Such a network comprises effectively three layers: the input layer, which is fully connected to the hidden layer generating GRBFs responses and a single node output layer which combines these responses in terms of a weighted sum to generate a pdf estimate for the input stimulus.

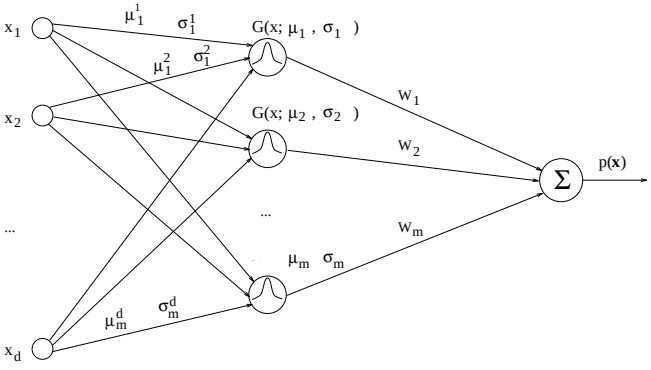

Figure 1: Block diagram of the GRBF neural net. For each class a network like this outputs the pdf. The outputs are then compared to build the classifier with the Bayes rule.

Let $\mathbf{B} \equiv (m, W_1, \ldots, W_m, \mu_1, \ldots, \mu_m, \sigma_1, \ldots, \sigma_m)$ be the parameter vector that uniquely defines the network output and it is estimated by the minimization of the KL distance, which is equivalent to the maximization of the likelihood ([4]). Unfortunately such a choice leads to selecting very complex networks (large m) which have poor generalization property: they are just tightly fit the data. Adding neurons to the network results in an improved likelihood at the expense of an incresing dimensionality of the parameter vector.

In order to avoid such a problem, we have modified the criterion function, adding a term which quantifies the architecture complexity. The new optimality criterion is then

$$J(\mathbf{B}) = \log L(\mathbf{B}|m) + NC(m) \tag{2}$$

where $L(\mathbf{B})$ is the likelihood, given the network architecture, i.e. m the number of hidden nodes, and $NC(m)$ is the network complexity. This last term is an increasing function of the number of free parameters involved by the network and decreases as the sample size increases.

Once the criterion function (2) has been defined a training has been carried out using the backpropagation technique.

First a set of admissible candidate mixtures is selected. A Gaussian mixtures of m components is admissible as a candidate function if the number of parameters to estimate

$$k(m) = m(2d+1) - 1$$

is smaller than the number of training set points available. For all the gaussian mixtures fulfilling the condition

$$k(m) < n$$

where k is the number of free parameters and n is the sample size, the network complexity function is defined as

$$NC(m) = -C_n \frac{dk(m) \log n}{rn} \tag{3}$$

with the dimesionality d, a relaxation factor $r \geq 2$ and a normalization constant C_n are chosen so that

$$\sum_{\text{all } \textit{admissable} \text{ m}} P(m) = 1 \tag{4}$$

$$P(m) = \exp(NC(m)) \tag{5}$$

The addition of this *penalty* term has the effects of weighting different network architectures according to their complexity. The likelihood increases steadily with the network complexity, while the penalty term decreases: the choice of m is dictated by the balance of this two terms.

Such strategy has rigourous information theoretic foundation ([1]). Consider our criterion function (2). It can be written as

$$J(\mathbf{B}) = \log(L(\mathbf{B}|m) \;\; P(m)) \tag{6}$$

where $P(m)$ is the probability assigned to the mixture of m components, defined in equation (5). The criterion (2) has a simple interpretation in a Bayesian framework: the parameter vector $\mathbf{B}$ is chosen in such a way that it maximize the joint probability of the data (likelihood) and the probability of the m-components mixture.

It can also be interpreted as a Minimum Description Length principle ([6]). For a detailed analysis, see [7].

3 EXPERIMENTAL RESULTS

We have tested the methodology outlined in the prevoius section on three different sets of data: 4-dimensional iris data, 8-dimensional mammographic data and 15-dimensional speech data. For the speech data a second independent set of data was available and we have checked the model selected by a Leave-One-Out (LOO) test, Cross Valaidation and MPL, while for the other two only a comparison between the MPL and LOO test was possible.

3.1 Iris data

This set of data is constituted of three classes, each containing 50 samples. The three classes represent three different species of iris plant, that are described by four features: sepal length and width, petal length and width. This database has been the object of many studies since the publication of Fisher's paper in 1936. It is known that one of the classes is linearly separable while the other two are not. The k-NN classifier, according to [2], gives the smallest error. We have applied our method and compared the result obtained (see Figure 2).

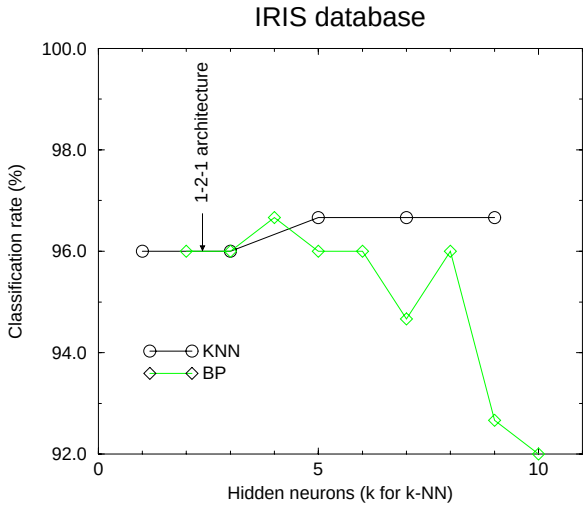

Figure 2: Plot showing the classification performance of a LOO test with architectures n-n-n for Back Propagation (BP) and odd k for k-NN

The best architecture selected by MPL gives the best but one performance with as little as only 4 hidden neurons (1 for the first and third classes and 2 for the second one). MPL or, for that matter, any other complexity analysis are often considered computationally intensive, but, according to our experience, they are not worse than a cross validation technique or a Leave-One-Out (LOO) test. In the particular case of the IRIS data, as in any other case when only a small sample size is available, cross validation is unthinkable and a LOO evaluation should be carried out on the exhaustive set of architectures. Just to give an idea, in order to exhaustively try all architectures with a maximum of 10 neurons in each class, we should have tried 10^2 architectures and this applies when neglecting the first class which is known to be linearly separable. For all our experiment with cross validation we have decided to test only architectures with the same number of hidden neurons for each class: they will be denoted as n-n-n (or n-n for two classes) architectures.

3.2 Mammograms

This data was obtained by digitalization of mammographic scans. A feature extraction and selection process was performed, providing a set of 1228 training patterns, each described by an 8-dimensional feature vector ([3]). The two classes of 921 and 307 samples refer respectively to normal healthy tissues and microcalcifications, that could indicate the presence of an early stage tumor. As already done for the other data, we checked the performance given by an $n-n$ architecture and then compared it with the best architecture selected by the MPL estimator. The results are plotted in Figure 3.

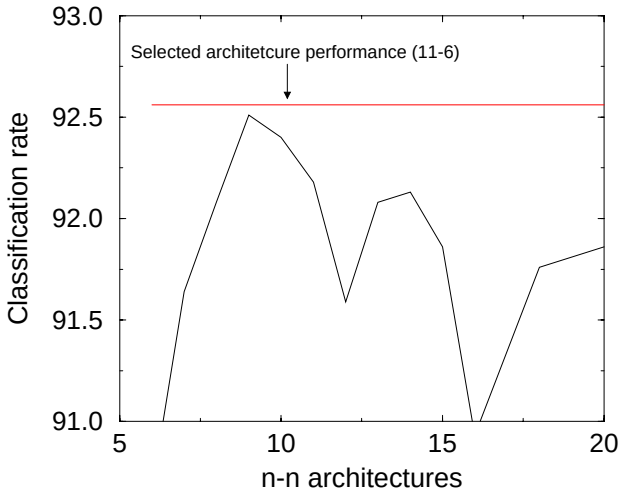

Figure 3: A comparison of the MPL selected architecture with the $n-n$ architectures (LOO test).

This example clearly shows how the MPL can guide in the choice of the architecture. The architecture selected gives the best performance compared to the $n-n$ architectures. Note that the criterion to train the network (Maximum Likelihood) is different from the criterion used to assess the performance: no information coming from the opposite class is used to estimate the pdf.

3.3 Speech data

The data used was a set of pattern vectors derived from the utterances *YES* and *NO* over the public switched telephone network. Each 15-dimensional vector contained 5 segments of 3 features derived by a low order linear prediction analysis. An independent test set was available for this data. The following table shows the sample sizes of these data sets.

	Yes	No
Training set	381	417
Test set	301	319

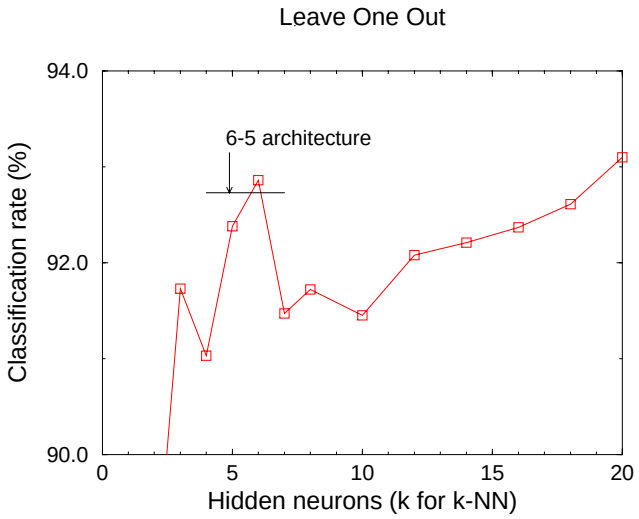

Figure 4: A comparison of the MPL and LOO test for different architectures as described in Figure 2 .

The cross validation and LOO tests have been applied to the data and results are reported in the following table and in Figures 3.3 and 5. Again only architectures (n-n) with the same number of neurons for each class have been tested by cross-validation.

Method	Nodes
LOO test	6 and 6
Cross validation	7 and 7
MPL $r \in [6, 14]$	3 and 3
MPL $r \geq 15$	6 and 5

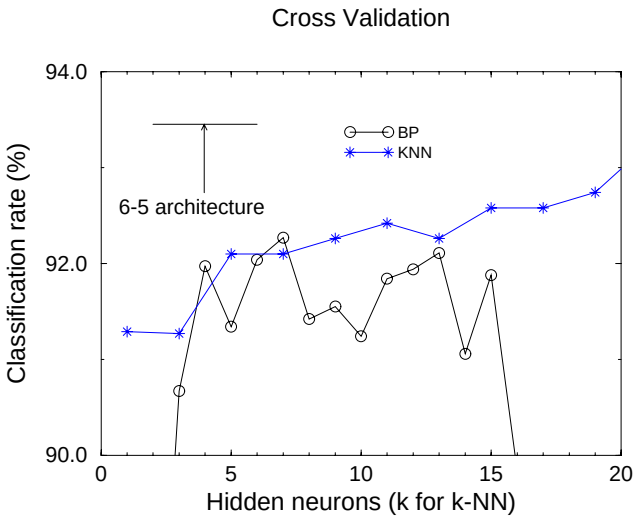

Figure 5: MPL and cross validation comparison.

The two different architectures selected correspond to the same MPL selector run with a more strict and a more relaxed Penalty term. This is obtained by changing the value of parameter r in equation (3): a larger r corresponds to a lighter penalization.

4 CONCLUSIONS

A GRBF network used as a non-parametric pdf estimator has been presented. Unlike the traditional non-parametric estimators, the use of a few GRBFs can ensure locality of the final estimation at a relative low computational cost.

The problem of the optimal selection of the number of GRBF units has been investigated, with particular attention to small sample set size cases. We have proposed an information criterion, denoted as MPL, that does not require large samples and is not very computationally intensive. The comparison with the more traditional techniques of cross-validation and leave-one-out error estimation shows that the estimator is reliable.

There is still an open problem regarding the choice of the relaxation factor of the complexity model, which is currently the object of investigation. Further improvement could be obtained by refining the learning algorithm.

We noted in Section 3.2 that our algorithm approximates the pdf of each class using the information coming only from the training set of that class. The introduction of a delearning phase, in which the network delearns (learns with a negative step) data from all the other classes, could be a solution, as some experiment have demonstrated, but a further investigation is needed.

ACKNOWLEDGEMENTS

The work reported in this paper has been supported by the EPSRC Research Grant GR/J89255.

References

[1] A. R. Barron and T. M. Cover. Minimum complexity density-estimation. *IEEE trans. on Information Theory*, 37(4):1034–1054, 1991.

[2] F. Blayo, Y. Cheneval, A. Guérin-Dudué, R. Chentouf, C. Aviles-Gruz, J. Madrenas, M. Moreno, and Voz J. L. Elena, esprit basic research project no. 6891, benchmarks. Technical Report Deliverable R3-B4-P, June 1995.

[3] S. A. Hojjatoleslami and J. Kittler. A system for the detection of clusters of microcalcifications in digitized mammograms. Technical Report VSSP-TR-1/9, Dept of Electrical & Electronic Eng., University of Surrey, UK, 1996.

[4] P. Pudil, J. Novovičová, and J. Kittler. Simultaneous learning of decision rules and important attributes for classification problems in image analysis. *Image and Vision Computing*, 12:193–198, 1994.

[5] S. Renals, N. Morgan, and H. Bourlard. Probability estimation by feed-forward networks in continous speech recognition. In *Proc. of the 1991 Workshop on Neural Networks for Signal Processing*, pages 309–318, 1991.

[6] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[7] L. Sardo and J. Kittler. Minimum complexity estimator for rbf networks architecture selection. To appear in Proceeding of ICNN96.

[8] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[9] H. G. C. Tråvèn. A neural network approach to statistical pattern-classification by semiparametric estimation of probability density-functions. *IEEE trans. on Neural Networks*, 2(3):366–377, 1991.