Robust Estimation for Radial Basis Functions

Adrian G. Borş I. Pitas

Department of Electrical and Computer Engineering
University of Thessaloniki
Thessaloniki 540 06, Greece

Abstract

This paper presents a new learning algorithm for radial basis functions (RBF) neural network, based on robust statistics. The extention of the learning vector quantizer for second order statistics is one of the classical approaches in estimating the parameters of a RBF model. The paper provides a comparative study for these two algorithms regarding their application in probability density function estimation. The theoretical bias in estimating one-dimensional Gaussian functions are derived. The efficiency of the algorithm is shown in modelling two-dimensional functions.

1 Introduction

Radial basis function (RBF) neural networks have been used in different applications in order to model unknown functions, providing the network with a training set [4]-[8]. RBFs have suitable properties to be used for function approximation [5], by decomposing a general function in a sum of kernels [2]. All the functions in this structure have similar parameters and can be embedded in a neural network.

The first approach considered in this paper is the second order statistics extension [7] for Learning Vector Quantization (LVQ) algorithm [3]. However, from statistical studies [2] this method is expected to give a large bias in the cases when data are long tailed distributed or contain outliers [1, 9]. In order to overcome these situations, we use an algorithm based on median type learning and called Median RBF (MRBF). Robust estimators are known to find the parameters best fitting to the bulk of the data and to identify outliers [2]. In the MRBF learning algorithm, we use the marginal median estimator in order to find the centers of the Gaussians and median of the absolute deviation for the covariance matrix parameters.

The RBF network has a feed-forward topology and can be used in unsupervised as well as in supervised learning. The network can be fed with real N-dimensional vectors denoted by X and the hidden units implement a Gaussian function:

$$\phi_j(X) = \exp\left[-(\mu_j - X)' \, \Sigma_j^{-1} (\mu_j - X)\right], \; j = 1, \ldots, L \tag{1}$$

L is the number of hidden units, μ is the mean vector and Σ is the covariance matrix. These weights are associated with input to hidden layer connections and geometrically they represent the centers and the shape parameters for the basis functions. Each hidden unit has associated an activation region, similar with the Voronoi partition from vector quantization. In order to assign a new sample to an activation region we have assumed two different metrics: Euclidean and Mahalanobis.

The output layer implements a weighted sum of hidden unit outputs:

$$\psi_k(X) = \sum_{j=1}^{L} \lambda(k, j) \; \phi_j(X), \; k = 1, \ldots, M \tag{2}$$

where M is the number of outputs.

The outputs are binary coded and a sigmoidal function is used in order to limit the output:

$$Y^k(X) = \frac{1}{1 + \exp[-\psi_k(X)]}, \; k = 1, \ldots, M \tag{3}$$

where Y^k is the kth output of the network.

2 Learning Algorithms

The weights in a RBF network can be found on-line by using a combined unsupervised-supervised technique [4]. The unsupervised part is derived from the LVQ algorithm and is similar to the adaptive k-means clustering.

In the first stage, the algorithm computes the distances from the given pattern to all the existing kernel centers. If we use Euclidean distance:

$$\text{If } \| X_i - \hat{\mu}_j \|^2 = \min_{k=1}^{L} \| X_i - \hat{\mu}_k \|^2 \text{ then } X_i \in C_j \tag{4}$$

where C_j is the kernel associated with the given pattern. Only the center of the winner class will be updated, according to the LVQ algorithm [3]:

$$\hat{\mu}_j = \hat{\mu}_j + \frac{1}{n_j} (X_i - \hat{\mu}_j) \tag{5}$$

for $j = 1, \ldots, L$, where n_j is the number of samples assigned to the cluster j. Taking the learning rate equal with the inverse of the number of samples associated with that unit we obtain a minimal output variance [10].

A similar method with (5) can be used to calculate the covariance matrix elements for each Gaussian neuron [7]:

$$\hat{\sigma}_{j,kl} = \frac{n_j - 2}{n_j - 1}\hat{\sigma}_{j,kl} + \frac{(X_i(k) - \hat{\mu}_j(l))(X_i(l) - \hat{\mu}_j(k))}{n_j - 1} \tag{6}$$

for $k, l = 1, \ldots, N$ $j = 1, \ldots, L$. The estimators in (5,6) are consistent with the classical statistical estimators for the first and the second order statistics.

The Mahalanobis square distance takes into consideration the covariance matrix for each hidden unit and can be used instead of (4):

$$\text{If } (\hat{\mu}_j - X_i)' \hat{\Sigma}_j^{-1}(\hat{\mu}_j - X_i) = \min_{k=1}^{L}(\hat{\mu}_k - X_i)' \hat{\Sigma}_k^{-1}(\hat{\mu}_k - X_i) \text{ then } X_i \in C_j \tag{7}$$

In the training stage it is desirable to avoid using patterns which may cause bias in the parameter estimation. The LVQ algorithm together with its extention in RBF network do not have robustness against the outliers or against the erroneous choices for the parameters. Robust estimators are known to provide accurate estimates when data are contaminated with outliers or have long-tailed distributions [1, 2]. The marginal median LVQ algorithm [9] can be used in order to evaluate the reference vectors for each partition region. In order to avoid increasing complexity, the samples assigned to a neuron pass through a running window W. If the data statistics change in time, then W is small. If a better evaluation of the median is desired then W is large. The learning rule is given by:

$$\hat{\mu}_j = \begin{cases} \text{med}\,\{X_0, X_1, \ldots, X_i\} & \text{if } i < W \\ \text{med}\,\{X_{i-W}, X_{i-W+1}, \ldots, X_i\} & \text{if } i \geq W \end{cases} \tag{8}$$

For the robust estimation of the scale parameter we use the median of the absolute deviation (MAD):

$$\hat{\sigma}_{j,hh} = \frac{\text{med}\,\{|X_1 - \hat{\mu_j}|, \ldots, |X_W - \hat{\mu}_j|\}}{0.6745} \tag{9}$$

where 0.6745 is a scaling factor in order to make the estimator consistent for the normal distribution [1, 2]. The cross-correlation components of the covariance matrix can be derived from the MAD calculated for $X_i(h) + X_i(l)$ and $X_i(h) - X_i(l)$ [2].

In both algorithms, for supervised learning, a second layer it is used in order to group the clusters found in the unsupervised stage. The output weights are updated as:

$$\lambda(k, j) = \lambda(k, j) + \eta\,(Y^k(X) - F^k(X))Y^k(X)(1 - Y^k(X))\phi_j(X) \tag{10}$$

for $k = 1, \ldots, M$ $j = 1, \ldots, L$ and $\eta \in (0, 1)$ is the learning rate. $F^k(X)$ is the desired output for the pattern X and it is binary coded. The formula (10) corresponds to the backpropagation for RBF network with respect to the square error cost function [8].

3 The Performance Analysis

We consider the case when we have a mixture of one-dimensional normal functions $\mathrm{N}(\mu_j, \sigma_j)$:

$$f(X) = \sum_{j=1}^{L} \frac{\varepsilon_j}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(X-\mu_j)^2}{2\sigma_j^2}\right] \tag{11}$$

$$\sum_{j=1}^{L} \varepsilon_j = 1 \tag{12}$$

where ε_j is the a priori probability for the function j. In the case of a mixture of multivariate normal distributions, the estimation can be done on marginal data. If we consider more complex distribution functions, they can be decomposed in sums of mixed Gaussians and reduced to the model (11).

We estimate the center for the jth Gaussian:

$$E[\hat{\mu}_j] = E[X|X \in [\hat{T}_j, \hat{T}_{j+1})] = \frac{\int_{\hat{T}_j}^{\hat{T}_{j+1}} Xf(X)dX}{\int_{\hat{T}_j}^{\hat{T}_{j+1}} f(X)dX} \tag{13}$$

where $\hat{T}_j$ and $\hat{T}_{j+1}$ are the estimates of the separating boundaries for the jth Gaussian kernel and $f(X)$ is given by (11). In order to evaluate the parameters for one Gaussian from a mixture of normal functions we should also consider parts from neighboring functions which are inside the boundaries $\hat{T}_j$ and $\hat{T}_{j+1}$. Replacing (11) in (13) we derive the stationary value of the mean estimate, valid for (5).

The median is located where the *pdf* of the given data is split in two equal areas [1]. From this condition the stationary value of the center estimate for the jth Gaussian distribution can be obtained by using the median operation:

$$\sum_{i=1}^{L} \varepsilon_i \,\mathrm{erf}\left(\frac{E[\hat{\mu}_j^{med}] - \mu_i}{\sigma_i}\right) = \sum_{i=1}^{L} \frac{\varepsilon_i}{2}\left[\mathrm{erf}\left(\frac{T_{j+1}-\mu_i}{\sigma_i}\right) + \mathrm{erf}\left(\frac{T_j - \mu_i}{\sigma_i}\right)\right] \tag{14}$$

where we consider the definition for the erf function:

$$\mathrm{erf}(X) = \frac{1}{\sqrt{2\pi}} \int_0^X exp\left(-\frac{t^2}{2}\right) dt \tag{15}$$

The stationary value for the estimate of the variance using the classical estimator (6) is given by:

$$E[\hat{\sigma}_j^2] = E[(X - \hat{\mu}_j^{mean})^2 | x \in [\hat{T}_j, \hat{T}_{j+1})] = \frac{\int_{T_j}^{T_{j+1}} (X - E[\hat{\mu}_j^{mean}])^2 f(X)dX}{\int_{T_j}^{T_{j+1}} f(X)dX} \tag{16}$$

where $f(X)$ is from (11) and $E[\hat{\mu}_j^{mean}]$ is the stationary value of the center estimate using the mean estimator.

From similar properties like those used for (14), for the MAD estimator (9) we can derive its expected stationary value from:

$$\sum_{i=1}^{L} \varepsilon_i \left[\text{erf}\left(\frac{E[\hat{\mu}_j^{med}] - \mu_i + cE[\hat{\sigma}_j^{MAD}]}{\sigma_i} \right) - \text{erf}\left(\frac{E[\hat{\mu}_j^{med}] - \mu_i - cE[\hat{\sigma}_j^{MAD}]}{\sigma_i} \right) \right] =$$

$$= \frac{1}{2} \sum_{i=1}^{L} \varepsilon_i \left[\text{erf}\left(\frac{\hat{T}_{j+1} - \mu_i}{\sigma_i} \right) - \text{erf}\left(\frac{\hat{T}_j - \mu_i}{\sigma_i} \right) \right] \tag{17}$$

where c=0.6745.

In order to evaluate the parameters for the Gaussian kernels we must also evaluate the activation domains $V = [\hat{T}_j, \hat{T}_{j+1})$ for each Gaussian function. If the Euclidean distance is used in order to assign a new pattern to an activation region (4), we can estimate the boundary $\hat{T}_j$ between two activation regions j and $j+1$ as:

$$\hat{T}_j = \frac{\hat{\mu}_j + \hat{\mu}_{j+1}}{2} \tag{18}$$

for $j = 1, \ldots, L-1$. The first and the last boundaries are: $T_0 = -\infty$ and $T_L = \infty$.

In the case when the Euclidean distance is replaced by the Mahalanobis distance (7), then the boundary condition in one-dimensional case can be found solving the equation:

$$\left(\frac{\hat{T}_j - \hat{\mu}_j}{\hat{\sigma}_j} \right)^2 = \left(\frac{\hat{T}_j - \hat{\mu}_{j+1}}{\hat{\sigma}_{j+1}} \right)^2 \tag{19}$$

for $j = 1, \ldots, L-1$. Analytical methods can be used in order to find the boundaries as well as the model parameters.

We consider the following particular examples :

$$f(X) = \frac{1}{2} N(5, \sigma) + \frac{1}{2} N(10, \sigma) \tag{20}$$

$$f(X) = \frac{1}{3} N(3, \sigma) + \frac{1}{3} N(5, \sigma) + \frac{1}{3} N(10, \sigma) \tag{21}$$

We estimate the center and the scale parameter for the distribution $N(5, \sigma)$ using both mean and median estimators for RBF centers. The absolute errors $E[\hat{\mu}] - \mu$ are depicted in Figure 1a for the distribution (20) and in Figure 1b for the distribution (21) with respect to the scale parameter σ. The comparison results in the bias estimation for the scale parameter $E[\hat{\sigma}] - \sigma$ are presented in Figure 1c for the distribution (20) and in Figure 1d for the distribution (21). The estimation of the class means and scale parameters of (20) corresponds to the estimation of parameters of medium-tailed distribution and in the case of (21) to a short tailed distribution. All these plots show that in the cases when it is occuring a certain overlap in the functions to be estimated, the bias given by the robust algorithm it is smaller than that obtained by using classical methods.

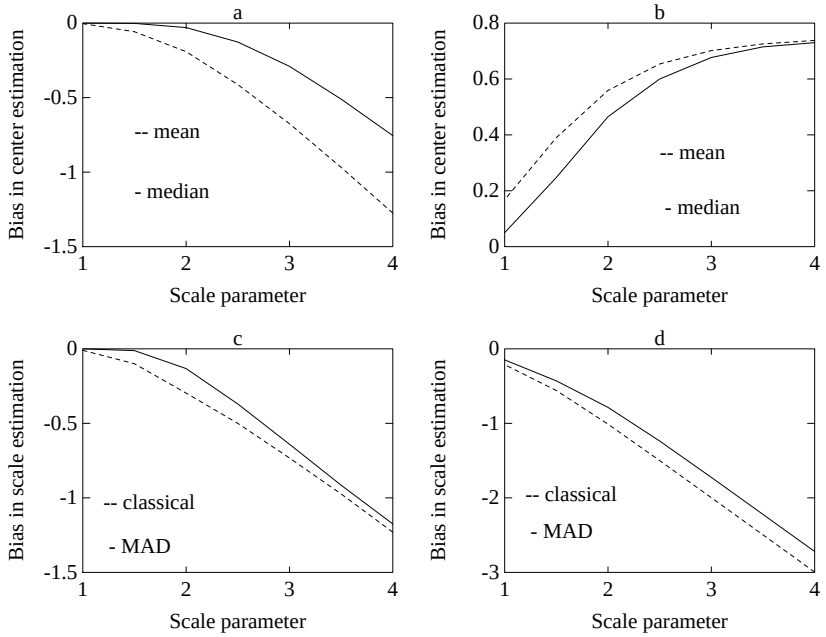

Figure 1: Theoretical analysis for robust and classical statistics estimators in evaluating the RBF parameters: a) estimation of the center for N(5,2) in a long-tailed distribution and b) in a short-tailed distribution; c) estimation of the scale parameter for N(5,2) in the first distribution and d) for the second distribution.

4 Simulation results

We have applied both algorithms presented in Section 2 and analyzed in Section 3 to the estimation of the parameters for mixed bivariate normal distributions. The first algorithm uses classical statistics estimators for finding the RBF parameters and the second uses robust estimators. In these applications we have used both Euclidean and Mahalanobis distances in order to assign a new coming pattern to a cluster.

We apply the networks for estimating the following distributions:

Distribution I: $P_1^I(X)$=N(2,1;3,1;0)+N(8,7;3,1;0)
$P_2^I(X)$=N(8,2;1,3;0)+N(2,6;1,3;0)

Distribution II: $P_1^{II}(X)$=N(6,0;4,1;0)+N(0,6;1,4;0)
$P_2^{II}(X)$=N(6,6;2,2;0)

Distribution III: $P_1^{III}(X)$=ϵ P_k^I+(1-ϵ)U([-5,15],[-5,15])

Distribution IV: $P_1^{IV}(X)$=ϵ P_k^{II}+(1-ϵ)U([-5,15],[-5,15])

where we denote a Gaussian distribution through $N(\mu_1, \mu_2; \sigma_1, \sigma_2; r)$, r is the correlation factor and a uniform distribution through U and $k \in \{1, 2\}$, $\epsilon = 0.9$.

Table 1: Comparison between RBF and MRBF algorithms

Distribution	Method	Distance Measures			
		Euclidean		Mahalanobis	
		Error (%)	MSE	Error (%)	MSE
I	RBF	21.26	13.69	17.17	6.90
	MRBF	17.58	8.65	13.75	2.75
	Optimal	12.13	0.00	12.13	0.00
II	RBF	3.89	3.69	2.95	1.24
	MRBF	2.90	1.20	2.61	0.82
	Optimal	2.52	0.00	2.52	0.00
III	RBF	26.63	34.22	35.05	48.59
	MRBF	21.11	10.11	18.82	5.74
	Optimal	15.78	0.00	15.78	0.00
IV	RBF	15.28	32.36	22.21	39.61
	MRBF	8.78	5.50	7.24	2.49
	Optimal	7.18	0.00	7.18	0.00

The comparison measures are the miss-classification error and mean square error (MSE) between the true functions and those modeled by means of the neural network. The problem of multi-distribution estimation is seen as a pattern classification task. The optimal network is obtained when its parameters are identical to those of the given Gaussian distributions. The MSE is defined as:

$$MSE = \frac{1}{M} \sum_{k=1}^{M} \int_{\mathcal{D}} (Y^k(X) - \hat{Y}^k(X))^2 \, dX \tag{22}$$

where the domain is $\mathcal{D} = (-\infty, \infty) \times (-\infty, \infty)$ in our case , $\hat{Y}^k(X)$ is the surface for the kth output unit and $Y^k(X)$ is the target function.

In the learning stage, we consider a window of W=401 samples (8) (for MRBF) and 4000 learning samples with equal number of samples for each cluster. The comparison results between the two methods are given in Table 1 where the same data were used for both algorithms. The simulations were repeated with different data, consistent with the same distribution functions and the presented results are the average of all these trials.

In all these cases, we have obtained a clear improvement by using the MRBF algorithm. When the mixture of bivariate normal distributions is contaminated with uniform distributed patterns the difference is very large because the median type learning is insensitive to the presence of outliers. Using the Mahalanobis distance instead of the Euclidean distance, we obtain better results, except for the classical estimators in the case of models contaminated by uniform noise.

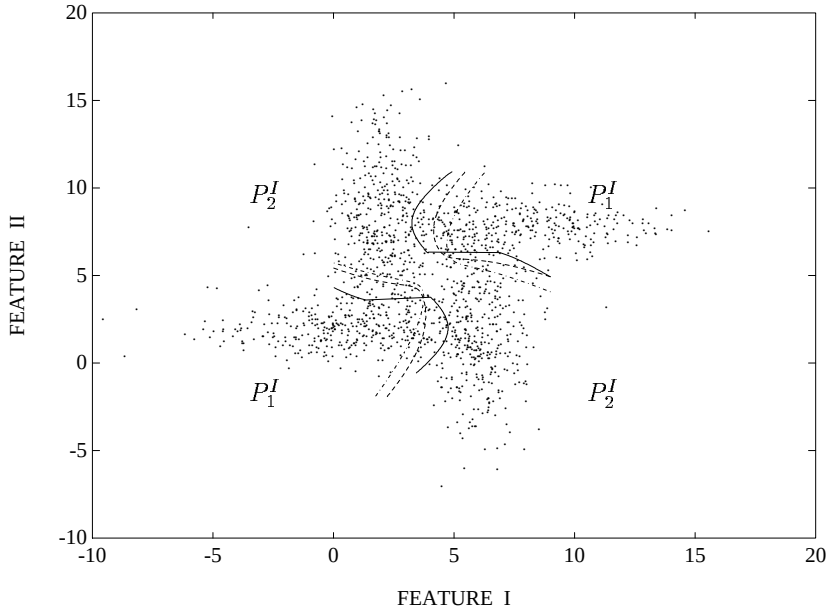

Figure 2: Samples from the distribution I and the boundaries between the classes: '-' optimal classifier, '- -' MRBF and '-·' RBF.

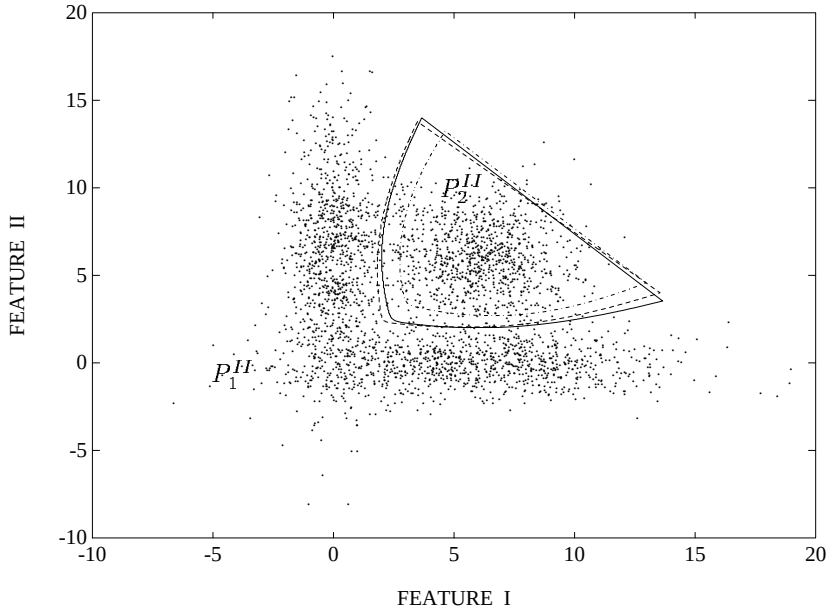

Figure 3: Samples from the distribution II and the boundaries between the classes: '-' optimal classifier, '- -' MRBF and '-·' RBF.

Samples drawn from the distributions I and II are depicted in Figures 2 and 3. The separation boundaries found by means of the RBF and MRBF networks as well as the optimal boundary are marked in these Figures. The separation boundaries are situated where two neighboring classes have equal probabilities. The decision rule for the assignment of a new pattern was based on Euclidean distance in Figure 2 and on Mahalanobis distance in Figure 3. It can be seen from these Figures that we obtain a better approximation of the optimal boundary by using MRBF compared with the classical algorithm.

In Figure 4 we evaluate the convergence of these algorithms in the case of distribution I. The learning curves represent the estimation of the *pdf* functions (MSE) with respect to the number of samples. From this plot the improvement given by MRBF compared with classical RBF learning and by using the Mahalanobis distance instead of the Euclidean distance is clear.

From the Table 1 we can see that MRBF gives better results in estimating the *pdf* functions and it is not biased by the presence of the outliers. MRBF gives more accurate approximations for the Bayesian boundaries then the classical statistical algorithms in the case of bivariate mixtures of Gaussians, as can be seen in Figures 2, 3. Median type learning applied to radial basis functions converge smoothly to a stationary value smaller than that obtained in classical estimation for RBF as can be seen in Figure 3.

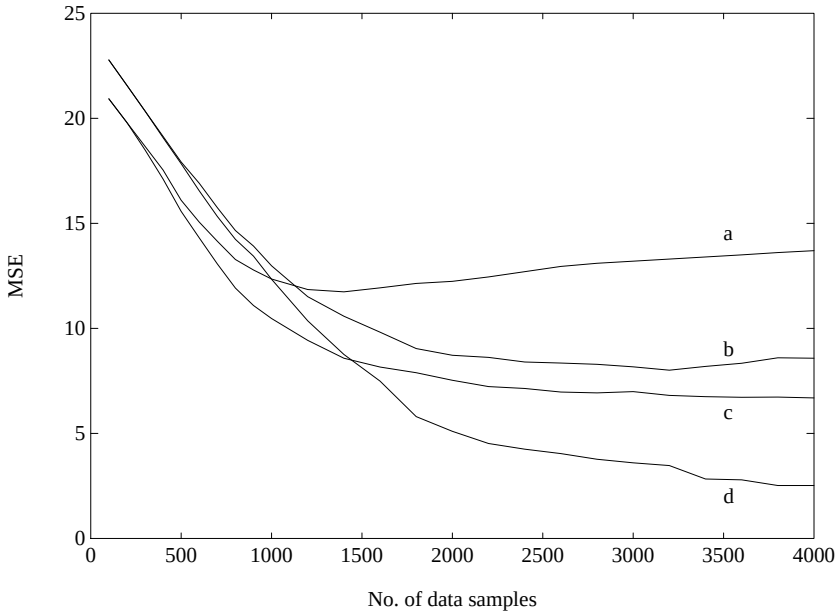

Figure 4: The learning curves in the case when the samples are drawn from the distribution I. Classical estimators are used together with the Euclidean distance for curve a and together with the Mahalanobis distance for curve c ; robust estimators are used together with the Euclidean distance for curve b and together with the Mahalanobis distance for curve d.

5 Conclusions

This paper presents a comparative study of two learning algorithms, one based on classical statistics estimators and the other on robust estimators. The algorithm derived from robust statistics and called Median RBF uses the median in order to find the centers in the network and median of absolute deviations for the estimation of the scale parameters. Both algorithms can be implemented on-line. We have derived theoretical analysis in a parameter estimation problem. The algorithm based on robust statistics is proved to give more accurate results in the one-dimensional estimation problem as well as in a two dimensional density function approximation. Possible fields of application for this algorithm are in communication systems, image processing and speech recognition.

References

[1] I. Pitas, A. N. Venetsanopoulos, *Nonlinear Digital Filters: principles and applications*, Hingham, MA: Kluwer Academic, 1990.

[2] G. Seber, *Multivariate Observations*, John Wiley, 1986.

[3] T. K. Kohonen, *Self-organization and associative memory*, 3rd edition, Berlin, Germany: Springer-Verlag, 1989.

[4] J. Moody, C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation,* vol. 1, no. 2, pp. 281-294, 1989

[5] T. Poggio, F. Girosi, "Networks for approximation and learning," *Proc. of the IEEE,* vol. 78, no. 9, pp. 1481-1497, Sep. 1990

[6] D. F. Specht, "A general regression neural network," *IEEE Trans. on Neural Networks,* vol. 2, no. 6, pp. 568-576, Nov. 1991

[7] S. Chen, B. Mulgrew, P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. on Neural Networks,* vol. 4, no. 4, pp. 570-579, Jul 1993

[8] A. G. Borş, M. Gabbouj, "Minimal topology for a radial basis functions neural network for pattern classification," to appear in *Digital Signal Processing , A review Journal*, 1994

[9] I. Pitas, P. Kiniklis, "Median learning vector quantizer," *Proc. SPIE, vol. 2180, Nonlinear Image Processing V,* San Jose, CA, pp. 23-34, 7-9 Feb. 1994

[10] E. Yair, K. Zeger, A. Gersho "Competitive learning and soft competition for vector learning quantizer design," *IEEE Trans. on Signal Processing,* vol. 40, no. 2, pp. 294-309, Feb. 1992