CHAPTER 10

Robust RBF Networks

A.G. Borş
Department of Computer Science
University of York
York YO10 5DD, U.K.
Adrian.Bors@cs.york.ac.uk

I. Pitas
Department of Informatics
University of Thessaloniki
Thessaloniki 540 06, Greece
pitas@zeus.csd.auth.gr

We introduce training algorithms for Radial Basis Function (RBF) networks using robust statistics. The proposed training algorithms have two stages. The first stage rely on a robust learning vector quantization approach which estimates the hidden unit weights. The second stage employs backpropagation for the output weight calculation. We introduce the Median RBF (MRBF) training algorithm and Alpha-Trimmed Mean RBF. The efficiency of MRBF and classical training using learning vector quantization are compared in estimating overlapping Gaussian distributions. Applications to artificial data classification and object modeling are provided for the proposed algorithms.

1 Introduction

Radial Basis Functions (RBF) have been used in several applications for functional modeling and pattern classification. They have been found to have very good functional approximation capabilities. It has been proven that any continuous function can be modeled up to a certain precision by a set of radial basis functions [1], [2], [3]. RBFs have their fundamentals drawn from probability function estimation theory.

RBF network consists of a two layer feed-forward neural network. The hidden units implement functions which geometrically have a radial activation region similar to that of electric charges. Various types of functions have been considered for the hidden unit activation functions. Gaussian, thin-plate, multi-quadric, cubic radius or Cauchy basis function are among those proposed for modeling radial basis functions. Gaussian functions are considered in this study, because they are able to model the first and the second order statistics with their mean and covariance matrix. Modeling using mixtures of Gaussian functions has been considered in many fields of science.

The output layer implements a sum of hidden unit outputs. The network inputs have different meanings depending on the application. When modeling time series, the inputs represent data samples located at various time laps [4], [5], [6], [7], [8]. In a pattern recognition application each input corresponds to a feature entry [9]. The number of inputs determines the size of the hyper space in which the classification takes place. As a consequence of their functional approximation capabilities, RBF networks have been shown to approximate the Bayesian classifier [9], [10]. Usually, in the case of time-series approximation, only one network output is assumed, while in the case of pattern approximation each network output is assigned to a class.

The parameters of the RBF neural network have to be estimated by means of a training algorithm in order to model the functions implied in a specific problem. A large variety of training algorithms has been tested in RBF networks. Most of these training algorithms correspond to supervised training or to a joint unsupervised-supervised paradigm. Each data sample has been assigned a basis function in [15]. In such a case too many data samples are required. In a supervised problem a training set of input-output pairs is given. The estimation of the neural network parameters can be seen as finding a system of equations solution [16]. This task can be achieved by computing the inverse of a matrix corresponding to the RBF hidden unit outputs for a certain set of data samples. The product of this matrix with the vector of network outputs provides the solution of the system. The output unit weights are calculated correspondingly. A large computation time is required for calculating the inverse matrix in the case when the system of equations is big. Due to their

functional approximation capabilities, RBF networks have been seen as a good solution to interpolation problems [17], [18]. A least squares solution has been found suitable for estimating the output parameters of the RBF networks [17]. Orthogonal least squares has been proposed in [19] for estimating the RBF weights. Gram-Schmidt orthogonalization is combined with a merge condition for finding the parameters of the RBF networks in [20]. An adaptive solution is the use of a gradient descent algorithm. Backpropagation algorithm has been proposed for training RBF networks [3], [9], [10], [12], [21]. Backpropagation algorithm provides a near-optimal solution. However, according to the specific problem, it may require a large amount of iterations and training time. Expectation-maximization algorithm is proposed for training a generalized Gaussian network in [13]. Clustering algorithms, e.g., k-means [11] or its adaptive implementation represented by the LVQ [14], have been applied for estimating the hidden unit parameters. In such a case, the center of the basis function corresponds to the first order statistics of data samples from a certain cluster. The covariance matrix associated with a basis function models the second order statistics. The solutions offered by such training algorithms are sub-optimal. However, clustering-based algorithms can handle large amounts of data and they offer good solutions to a variety of problems.

The number of hidden units is chosen using either a criterion which compromises between the performance and the number of required parameters, as Akaike criterion does [19], or based on a condition of cluster merging [20]. Adding new hidden units when certain conditions are fulfilled is proposed in [9], [21]. Clustering merging-splitting [11] can be employed for selecting the number of hidden units when clustering algorithms are employed for training.

Radial Basis Functions have close connections with other classical pattern classification approaches as potential functions [11], [12] or clustering algorithms such as k-means [11] and Learning Vector Quatization [14]. They have interesting properties which make them attractive in very many applications. Chaotic-time series generated by dynamical systems have been modeled by RBFs in [4], [16]. RBF networks have good approximation capabilities when modeling time-series generated by Mackey-Glass differential equation [4], [5], [6], [7]. Control modeling

by RBF Networks has been employed in [18]. Equalization and detection problems in the case when assuming inter-symbol and co-channel interference have been solved using RBF Networks [9], [10], [19], [27]. RBF networks have been used for static speech classification [6], [9], [17]. Several applications of RBF networks have been done in image processing as well. In [13] the neural network is employed for image restoration when considering that the image is distorted by a non-linear distortion function. A face classification system successfully used an RBF network in [22]. Modeling 3-D shapes from shading has been employed based on RBF networks in [23]. RBF networks have been used for modeling 3-D shapes in medical imaging [24], [25].

In this study we present some algorithms for training RBF networks using robust statistics. After introducing the structure of the RBF network and describing the training algorithms based on clustering in Section 2.1, we introduce the Median RBF Network and the Alpha-Trimmed Mean RBF Network in Sections 2.1 and 2.2 respectively. Section 3 provides the analysis of the bias in the case when RBF Networks are used to estimate a mixture of Gaussian functions. A comparison between the RBF network trained using classical estimation and robust statistics estimation is done in Section 3. Experimental results are provided in Section 4 when using the proposed algorithms in a 3-D object modeling application. The conclusions of this study are drawn in Section 5.

2 Estimation of RBF Parameters Using the LVQ Algorithm

In the approach adopted in this study, we consider Gaussian activation functions for the hidden units:

$$\phi_j(\mathbf{X}) = \exp\left[-(\mu_j - \mathbf{X})^T \, \mathbf{\Sigma}_j^{-1}(\mu_j - \mathbf{X})\right], \; j = 1, \ldots, L \qquad (1)$$

where μ is the mean vector and $\mathbf{\Sigma}$ is the covariance matrix. Geometrically, μ represents the center and $\mathbf{\Sigma}$ the shape of the basis functions. A hidden unit function can be represented as a hyper-ellipsoid in the N-dimensional space.

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

$$\psi_k(\mathbf{X}) = \sum_{j=1}^{L} \lambda_{jk}\phi_j(\mathbf{X}) \tag{2}$$

where $k = 1, \ldots, M$. L is the number of hidden units and M is the number of outputs. The weights λ_{kj} show the contribution of the hidden unit j for modeling the output k.

In pattern classification applications, the outputs are limited within the interval (0,1) by a sigmoidal function:

$$Y_k(\mathbf{X}) = \frac{1}{1 + exp[-\psi_k(\mathbf{X})]} \tag{3}$$

2.1 Classical Statistics Estimation

A two-stage training algorithm has been employed in [6] considering the capabilities of basis functions to represent data clusters. This algorithm has a first unsupervised stage and a second supervised stage. In the first stage, the parameters of the basis function centers are estimated based on a self-organizing algorithm, *e.g.*, Learning Vector Quantization (LVQ) [14]. In the second stage, the algorithm estimates the parameters of the output weights λ based on the LMS algorithm [26]. Such an algorithm even though is not optimal provides a clear speed-up in the training time over other RBF training algorithms. In the first stage a set of centers are generated at random and the algorithm calculates the Euclidean distances between the data samples and the class centers. The closest center to the given data vector is chosen to be updated:

$$\|\mathbf{X} - \mu_k\| = \min_{i=1}^{L} \|\mathbf{X} - \mu_i\| \tag{4}$$

where μ_k is the closest center to the incoming data sample $\mathbf{X}$. The center is updated as follows:

$$\mu_k(t) = \mu_k(t-1) + \eta(\mathbf{X} - \mu_k(t-1)) \tag{5}$$

where η is the training rate and t is the iteration number. Similar algorithms have been used in [27] and [28]. The LVQ algorithm for minimum

output variance [29] is obtained when we use the learning rate from (5) as in [28]:

$$\eta = \frac{1}{N_k(t-1)} \tag{6}$$

where $N_k(t-1)$ represents the number of data samples associated with the k-th hidden unit at iteration $t-1$ according to Equation (4). For the sake of simplicity, we will drop the t-dependency of N_k from now on.

For the covariance matrix, the on-line updating is:

$$\hat{\mathbf{\Sigma}}_k(t) = \frac{N_k - 2}{N_k - 1}\hat{\mathbf{\Sigma}}_k(t-1) + \frac{(\hat{\mu}_i(t-1) - \mathbf{X})(\hat{\mu}_i(t-1) - \mathbf{X})^T}{N_k - 1} \tag{7}$$

In some applications it is worth using the Mahalanobis distance instead of the Euclidean distance for the choice of the winner class. The Mahalanobis distance takes into consideration the covariance matrix for each basis function:

$$(\hat{\mu}_k - \mathbf{X}_i)^T \hat{\mathbf{\Sigma}}_k^{-1}(\hat{\mu}_k - \mathbf{X}_i) = \min_{j=1}^{L}(\hat{\mu}_j - \mathbf{X}_i)^T \hat{\mathbf{\Sigma}}_j^{-1}(\hat{\mu}_j - \mathbf{X}_i) \tag{8}$$

However, at the start of the learning algorithm, an imprecision in estimating the covariance parameters can occur and this can lead to a singular covariance matrix. Thus, for the first few data samples we can use the Euclidean distance (4) and afterwards employ the Mahalanobis distance (8). The initial values for the center estimates $\hat{\mu}$ are randomly generated and the covariance matrix is initialized with **0**.

The second layer is used in order to group the clusters, found in the unsupervised stage, in classes. We use the backpropagation algorithm for finding the output parameters [10]:

$$\lambda_{jk} = \sum_{\mathbf{X}}[F_k(\mathbf{X}) - Y_k(\mathbf{X})]Y_k(\mathbf{X})[1 - Y_k(\mathbf{X})]\phi_j(\mathbf{X}) \tag{9}$$

where $Y_k(\mathbf{X})$ is the output provided by the network (3) and $F_k(\mathbf{X})$ is the data target label for a supervised pattern classification problem.

2.2 Median RBF Networks

The approach described in the previous section corresponds to a classical (non-robust) statistics estimate of the mean and covariance matrix. In the

training stage, it is desirable to avoid using outlying patterns which may cause bias in the estimation of the RBF network parameters. The patterns which do not correspond to the data statistics (noisy patterns) should be rejected rather than used in the training stage [30], [33]. Let us consider the ordering of data samples assigned to a hidden unit $X_{(1)} < X_{(2)} < \ldots < X_{(N_k)}$. A robust statistics estimate of a cluster center is given by [30]:

$$\hat{\mu}_k = \frac{\sum_{i=1}^{N_k} W_i X_{(i)}}{\sum_{i=1}^{N_k} W_i} \tag{10}$$

where $W_i \in (0, 1)$ is the weight depending on the ranking of the respective data sample. The estimate of the covariance matrix can be calculated from

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{i=1}^{N_k} W_i^2 (\mathbf{X}_{(i)} - \hat{\mu}_k(t-1))(\mathbf{X}_{(i)} - \hat{\mu}_k(t-1))^T}{\sum_{i=1}^{N_k} W_i^2 - 1} \tag{11}$$

where the weights $W_i \in (0, 1)$ depend on the ranking as well [31], [32]. If $W_i = 1$ for $i = 1, \ldots, N_k$ (Equations (10) and (11)), then the center and the covariance matrix are calculated as in classical statistics. The adaptive versions of these are provided in Equations (5) and (7).

If W_i in Equation (10) is replaced by a function which decreases with respect to the distance of the ordered sample $X_{(i)}$ from the central ordered data sample $X_{(\lfloor \frac{N_k}{2} \rfloor)}$, we obtain various robust statistics estimators [30]. Robust statistics-based algorithms are known to provide accurate estimates when data are contaminated with outliers or have long-tailed distributions. They are insensitive to extreme observations which make them attractive for parameter estimation. Such algorithms have been used in image processing for noise removal [33]. In the Marginal Median LVQ (MMLVQ) algorithm [34], data samples are marginally ordered and the centroid is taken as the marginal median [33]:

$$\hat{\mu}_j = \text{med}\ \{\mathbf{X}_0, \mathbf{X}_2, \ldots, \mathbf{X}_{N_k - 1}\} \tag{12}$$

where $\mathbf{X}_{N_k-1}$ is the last pattern assigned to the cluster j. This corresponds to $W_i = 0$ for $i = 1, \ldots, \lfloor \frac{N_k}{2} \rfloor - 1$, $i = \lfloor \frac{N_k}{2} \rfloor + 1, \ldots, N_k$, and $W_i = 1$ for $i = \lfloor \frac{N_k}{2} \rfloor$ in Equation (10).

In order to avoid an excessive computational complexity, the median operation can be done on a finite data set, extracted through a moving window that contains only the last W data samples assigned to the hidden unit j:

$$\hat{\mu}_k = \begin{cases} \text{med}\,\{\mathbf{X}_0, \mathbf{X}_1, \ldots, \mathbf{X}_{N_k-1}\} & \text{if } N_k < W \\ \text{med}\,\{\mathbf{X}_{N_k-W}, \mathbf{X}_{N_k-W+1}, \ldots, \mathbf{X}_{N_k-1}\} & \text{if } N_k \geq W \end{cases} \tag{13}$$

where $\mathbf{X}_i$, $i = N_k - W, \ldots, N_k - 1$ are the samples assigned to the k-th neuron according to (4). The size of the window W can be small if the statistics of the sample population changes rapidly in time and large if the sample statistics are relatively unchanged in time and a better estimation of the given population median is desired. Unlike in image filtering [33], where the window is rather small, a rather big window should be employed in most cases in order to have a good estimate of the median for a population of samples.

For the scale parameter associated with kernel functions we use the median of the absolute deviation from the median (MAD) estimator given by :

$$\hat{\sigma}_k = \frac{\text{med}\,\{|\mathbf{X}_0 - \hat{\mu}_k|, \ldots, |\mathbf{X}_{N_k-1} - \hat{\mu}_k|\}}{0.6745} \tag{14}$$

where 0.6745 is a scaling parameter in order to make the estimator Fisher consistent for the normal distribution [30], [33]. MAD calculation is performed along each data dimension independently. The same set of data samples are taken into consideration in Equation (14) and for the marginal median in Equation (13).

The cross-correlation members of the covariance matrix can be calculated based on MAD estimator [30]. We consider two arrays corresponding to the difference and the sum of each two different components for a data sample from the moving window:

$$Z^+_{i,hl} = X_i(h) + X_i(l) \tag{15}$$

$$Z^-_{i,hl} = X_i(h) - X_i(l) \tag{16}$$

for $i = N_k - W, \ldots, N_k - 1$. First the median of these new populations is calculated according to Equation (13). The squares of the corresponding MAD estimates (14) for the arrays $\mathbf{Z}_{hl}^{+}$ and $\mathbf{Z}_{hl}^{-}$ represent their variances and they are denoted as $\hat{V}_{k,hl}^{+}$ and $\hat{V}_{k,hl}^{-}$. The cross-correlations are derived as:

$$\hat{\sigma}_{k,hl}^2 = \hat{\sigma}_{k,lh}^2 = \frac{1}{4}\,(\hat{V}_{k,hl}^{+} \; - \; \hat{V}_{k,hl}^{-}\,). \tag{17}$$

In marginal median LVQ, both Euclidean (4) and Mahalanobis distances (8) can be used. In the case of Mahalanobis distance, a good estimation is desired for the covariance matrix in order to be appropriately used for the winner class selection. By using a robust estimation of the covariance matrix as in Equations (14) and (17), we can be more confident in the evaluation of the Mahalanobis distance. The order of updating the RBF network weights is well defined: first the kernel center, the covariance matrix (which uses the previously evaluated center) and afterwards the hidden unit to output weights are updated. For the last one we use Equation (9). MRBF network has been successfully used for simultaneous optical flow estimation and moving object segmentation [39], brush stroke and crack pixel classification for painting restoration [40] and data fusion in person autentification [41].

2.3 Alpha-Trimmed Mean RBF networks

The α-trimmed Mean algorithm [30], [33] assigns $W_i = 1$ for $i = \alpha_k\, N_k$,. . .,$N_k - \alpha_k\, N_k$, and $W_i = 0$ for the rest of the data samples in (10). α_k is the percentage of data samples to be trimmed away at each extreme of the k-th hidden unit data distribution.

The adaptive implementation of the algorithm updates the parameters of a basis function using only the data samples which are inside a certain range of ranked samples:

$$\hat{\mu}_{k,t} = \begin{cases} \hat{\mu}_{k,t-1} \quad + \dfrac{\mathbf{X}_{(i)} - \hat{\mu}_{k,t-1}}{N_{k,t} - 2\alpha_k N_{k,t}} & \text{if } \alpha_k N_{k,t} < i < N_{k,t} - \alpha_k N_{k,t} \\ \hat{\mu}_{k,t-1} & \text{otherwise} \end{cases} \tag{18}$$

where $\hat{\mu}_{k,t}$ and $N_{k,t}$ denote the center estimate and the number of data samples assigned to the k-th basis function at the moment t. We can

observe that for α_k=0 we obtain the LVQ algorithm for minimum output variance [29].

The parameter α_k is chosen according to the data distribution. The following measure is used for estimating the tail of the data distribution [35], [36]:

$$Q = \frac{U[0.5] - L[0.5]}{U[0.05] - L[0.05]} \tag{19}$$

where $U[\beta]$, $L[\beta]$ represent the average of the upper and respectively the lower β percent of data samples assigned to a specific basis function. The number of data samples to be trimmed away relies directly on the value of Q:

$$\hat{\alpha}_k = \frac{1-Q}{2}. \tag{20}$$

For long tailed distributions, the amount of data samples to be trimmed is large and for short tailed distributions less samples are trimmed away.

For the second order statistics, we order the data samples assigned to a basis function according to their Mahalanobis distance from the estimated center $\hat{\mu}_k$, starting with:

$$\mathcal{M}_{(0)} = \min_{i=1}^{N_k}[(\mathbf{X}_i - \hat{\mu}_k)^T \hat{\mathbf{\Sigma}}_k^{-1} (\mathbf{X}_i - \hat{\mu}_k)]. \tag{21}$$

After ordering the data samples assigned to the k-th basis function according to this measure $\mathbf{X}_{(0),\mathcal{M}} < \ldots < \mathbf{X}_{(N_k),\mathcal{M}}$, the estimate of the covariance matrix is obtained using ellipsoidal trimming [37]:

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{i=0}^{N_k - \alpha_{k,\mathcal{M}} N_k} (\mathbf{X}_{(i),\mathcal{M}} - \hat{\mu}_k)\,(\mathbf{X}_{(i),\mathcal{M}} - \hat{\mu}_k)^T}{N_k - \alpha_{k,\mathcal{M}} N_k} \tag{22}$$

where $\mathbf{X}_{(i),\mathcal{M}}$ denotes the i-th ordered data sample according to the Mahalanobis distance (21) and $\alpha_{k,\mathcal{M}}$ is the trimming percentage in this case. As with Equation (18), this formula can also be implemented in an adaptive form. Equation (22) corresponds to peeling off observations in shells using a sequence of convex hulls. Equations (18) and (22) make up the Alpha-Trimmed Mean RBF training algorithm. An Alpha-Trimmed Mean RBF network has been applied for segmenting 3-D medical images [43].

3 Theoretical Assessment

In order to compare the results provided by the RBF network training algorithm based on classical statistics and those provided by the algorithms relying on robust statistics we estimate a function made up from overlapping Gaussian functions. Mixture of Gaussians are used for data modeling in many fields. We have chosen the MRBF algorithm as a representative robust training algorithm in order to be compared with the classical RBF network.

The marginal median operates independently on each data axis. Therefore, the performance analysis can be done for the one-dimensional (1-D) case without loss of generality. We shall use mixtures of 1-D normal distributions and we shall estimate the center and the dispersion for each Gaussian function by using robust techniques compared against the classical ones. We shall perform an asymptotic analysis of performance, i.e., when we have a sufficiently large number of observations for each class.

Let us consider a pdf function $f(X)$ being a mixture of L one-dimensional normal distributions $N(\mu_j, \sigma_j)$ each of them with *a priori* probability ε_j:

$$f(X) = \sum_{j=1}^{L} \frac{\varepsilon_j}{\sqrt{2\pi}\sigma_j} \exp\left[-\frac{(X-\mu_j)^2}{2\sigma_j^2}\right] \tag{23}$$

$$\sum_{j=1}^{L} \varepsilon_j = 1 \tag{24}$$

The second equation represents the normalization relationship for the *a priori* probabilities. We can assume that each distribution in the mixture (23) corresponds to one data class. Our aim is to separate each class j by choosing appropriate thresholds T_j, T_{j+1}. We have to consider the overlap among different distributions in estimating the components of Equation (23). The normalized data distribution after thresholding is given by

$$g(X) = \frac{f(X)}{\int_{T_j}^{T_{j+1}} f(X)dx} \tag{25}$$

where T_j and T_{j+1} are the optimal boundaries of the j-th function with its neighboring functions.

The expected value of the center can be obtained from:

$$E[\hat{\mu}_j] = E[X|X \in [\hat{T}_j, \hat{T}_{j+1})] = \frac{\int_{\hat{T}_j}^{\hat{T}_{j+1}} X f(X) dX}{\int_{\hat{T}_j}^{\hat{T}_{j+1}} f(X) dX} \tag{26}$$

where $\hat{T}_j$ and $\hat{T}_{j+1}$ are the estimates of the decision boundaries for the j-th Gaussian kernel and $f(X)$ is given in Equation (23).

In a parameter estimation problem, the bias represents the difference between the estimated parameter value and the optimal one. It is desirable to obtain a small bias. The bias of the boundary estimation between two classes is directly related to the estimation of the class probabilities. If these probabilities are well estimated, the bias is small. If not, the bias is larger.

When $\hat{\mu}_j$ is evaluated as in classical LVQ (5), the stationary value of the estimate for the jth Gaussian kernel center is given by:

$$\begin{aligned} E[\hat{\mu}_{j,LVQ}] = & \left\{ \sum_{i=1}^{L} \varepsilon_i \left\{ \mu_i \left[\text{erf}\left(\frac{\hat{T}_{j+1} - \mu_i}{\sigma_i} \right) - \text{erf}\left(\frac{\hat{T}_j - \mu_i}{\sigma_i} \right) \right] \right.\right. \\ & \left.\left. + \frac{\sigma_i}{\sqrt{2\pi}} \left\{ \exp\left[-\frac{(\hat{T}_j - \mu_i)^2}{2\sigma_i^2} \right] - \exp\left[-\frac{(\hat{T}_{j+1} - \mu_i)^2}{2\sigma_i^2} \right] \right\} \right\} \right\} \\ & \frac{1}{\sum_{i=1}^{L} \varepsilon_i \left[\text{erf}\left(\frac{\hat{T}_{j+1} - \mu_i}{\sigma_i} \right) - \text{erf}\left(\frac{\hat{T}_j - \mu_i}{\sigma_i} \right) \right]} \end{aligned} \tag{27}$$

where the erf function is [38] :

$$\text{erf}(X) = \frac{1}{\sqrt{2\pi}} \int_0^X exp\left(-\frac{t^2}{2} \right) dt \tag{28}$$

In the case of median estimator (12) the *pdf* for $N = 2i + 1$ independent and identically distributed data is given by [33]:

$$f_{i+1}(X) = N \begin{pmatrix} N-1 \\ i \end{pmatrix} F^i(X)[1 - F(X)]^i f(X) \tag{29}$$

where $F(X)$ is the cumulative distribution function (cdf) for the data whose pdf is given in Equation (23). If we replace Equation (29) in Equation (26), we obtain the expected value of the median estimator assuming N data samples.

The median is located where the *pdf* of the given data samples is split in two equal areas [33]:

$$\int_{\hat{T}_j}^{E[\hat{\mu}_{j,Med}]} f(X)dX = \int_{E[\hat{\mu}_{j,Med}]}^{\hat{T}_{j+1}} f(X)dX \tag{30}$$

The stationary value of the estimate for the center of the jth Gaussian distribution using the median can be obtained after replacing Equation (23) in Equation (30):

$$\begin{gathered}\textstyle\sum_{i=1}^{L} \varepsilon_i \text{ erf}\left(\frac{E[\hat{\mu}_{j,Med}]-\mu_i}{\sigma_i}\right) = \\ \textstyle\sum_{i=1}^{L} \frac{\varepsilon_i}{2}\left[\text{erf}\left(\frac{\hat{T}_{j+1}-\mu_i}{\sigma_i}\right) + \text{erf}\left(\frac{\hat{T}_j-\mu_i}{\sigma_i}\right)\right]\end{gathered} \tag{31}$$

In the case when we want to find the expectation for the variance estimator $\hat{\sigma}_j^2$, we can use a similar approach. The expected value for $\hat{\sigma}_j^2$ using the estimator (7) is

$$\begin{gathered}E[\hat{\sigma}_j^2] = E[(X-\hat{\mu}_{j,LVQ})^2 | x \in [\hat{T}_j, \hat{T}_{j+1})]) = \\ \frac{\int_{\hat{T}_j}^{\hat{T}_{j+1}} (X - E[\hat{\mu}_{j,LVQ}])^2 f(X)dX}{\int_{\hat{T}_j}^{\hat{T}_{j+1}} f(X)dX}\end{gathered} \tag{32}$$

where $f(X)$ is given by Equation (23) and $E[\hat{\mu}_{j,LVQ}]$ is evaluated in Equation (27).

When MAD is used as dispersion estimator (14), its expected value is given by

$$E[\hat{\sigma}_{j,MAD}] = \frac{\int_{\hat{T}_j}^{\hat{T}_{j+1}} \mid X - E[\hat{\mu}_{j,Med}] \mid f(X)dX}{c \int_{\hat{T}_j}^{\hat{T}_{j+1}} f(X)dX} \tag{33}$$

where $c = 0.6745$.

By taking into account the median property of splitting the data distribution into two equal areas as in Equation (30), we have

$$\int_{E[\hat{\mu}_{j,Med}]-cE[\hat{\sigma}_{j,MAD}]}^{E[\hat{\mu}_{j,Med}]+cE[\hat{\sigma}_{j,MAD}]} f(X)dX = \frac{1}{2}\int_{\hat{T}_j}^{\hat{T}_{j+1}} f(X)dX \tag{34}$$

where $E[\hat{\mu}_{j,Med}]$ is calculated in Equation (31) and $f(X)$ is obtained by replacing Equation (23) in Equation (29). For MAD, the expected stationary value can be found from

$$\begin{aligned}\sum_{i=1}^{L}\varepsilon_i &\left[\text{erf}\left(\frac{E[\hat{\mu}_{j,Med}]-\mu_i+cE[\hat{\sigma}_{j,MAD}]}{\sigma_i}\right)-\right.\\ &\left.\text{erf}\left(\frac{E[\hat{\mu}_{j,Med}]-\mu_i-cE[\hat{\sigma}_{j,MAD}]}{\sigma_i}\right)\right]=\\ =\frac{1}{2}\sum_{i=1}^{L}\varepsilon_i &\left[\text{erf}\left(\frac{\hat{T}_{j+1}-\mu_i}{\sigma_i}\right)-\text{erf}\left(\frac{\hat{T}_j-\mu_i}{\sigma_i}\right)\right]\end{aligned} \tag{35}$$

In order to evaluate the parameters for the Gaussian kernel we must also estimate the domains $[\hat{T}_j, \hat{T}_{j+1})$ of each Gaussian function. If the Euclidean distance is used in order to decide the activation region for a new incoming data sample as in Equation (4), we can estimate the boundary $\hat{T}_j$ between two activation regions j and $j+1$ as:

$$\hat{T}_j = \frac{\hat{\mu}_j + \hat{\mu}_{j+1}}{2} \tag{36}$$

for $j = 1, \ldots, L-1$.

In the case when the Euclidean distance is replaced with the Mahalanobis distance (8), the boundary condition can be found by solving the equation:

$$\left(\frac{\hat{T}_j - \hat{\mu}_j}{\hat{\sigma}_j}\right)^2 = \left(\frac{\hat{T}_j - \hat{\mu}_{j+1}}{\hat{\sigma}_{j+1}}\right)^2 \tag{37}$$

for $j = 1, \ldots, L-1$. The first and the last boundaries are defined as $T_0 = -\infty$ and $T_L = \infty$. The $2L-2$ parameters (Gaussian centers and boundaries) for the case described by Equation (36) and $3L-2$ parameters (including the shape parameters) for Equation (37) have to be evaluated. In order to do this, analytical methods can be employed by evaluating the centers of the Gaussian functions and the boundaries iteratively.

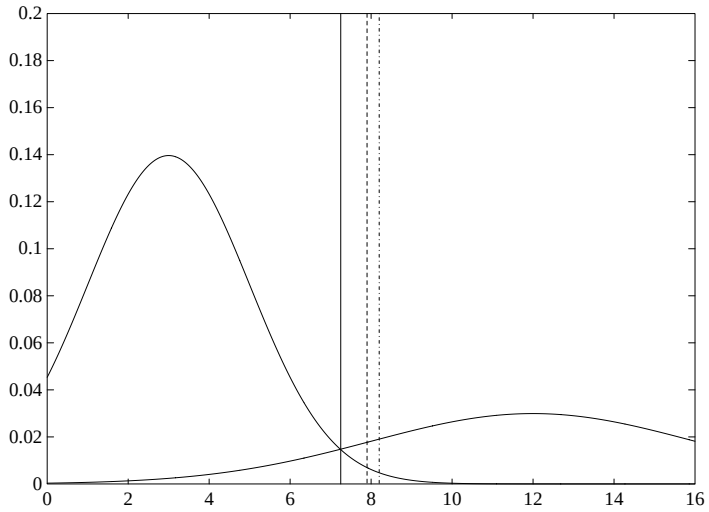


Figure 1. The decision boundary between two Gaussian density probability functions ('-' denotes the optimal boundary, '- -' the boundary found by using marginal median LVQ and '-.-' the boundary found using classical LVQ).

The relationship which gives the optimal boundary between two classes, each of them modeled by a Gaussian pdf, can be derived as:

$$\frac{(T_1-\mu_1)^2}{2\sigma_1^2}-\frac{(T_1-\mu_2)^2}{2\sigma_2^2}=\ln\frac{\sigma_1\varepsilon_2}{\sigma_2\varepsilon_1} \tag{38}$$

Two Gaussian pdf functions are shown in Figure 1: $p_1(X)$ with $\mu_1=3$, $\sigma_1=2$, $\varepsilon_1=0.7$ and $p_2(X)$ with $\mu_2=12$, $\sigma_2=4$, $\varepsilon_2=0.3$. The optimal boundary (38) is compared with the boundary obtained by means of MMLVQ and with the boundary given by classical LVQ. From this figure it is clear that the bias $|\,T_1-\hat{T}_1\,|$ provided by MMLVQ is smaller than that of the classical LVQ.

The particular examples considered here are:

$$f(X)=\frac{1}{2}N(5,\sigma)+\frac{1}{2}N(10,\sigma) \tag{39}$$

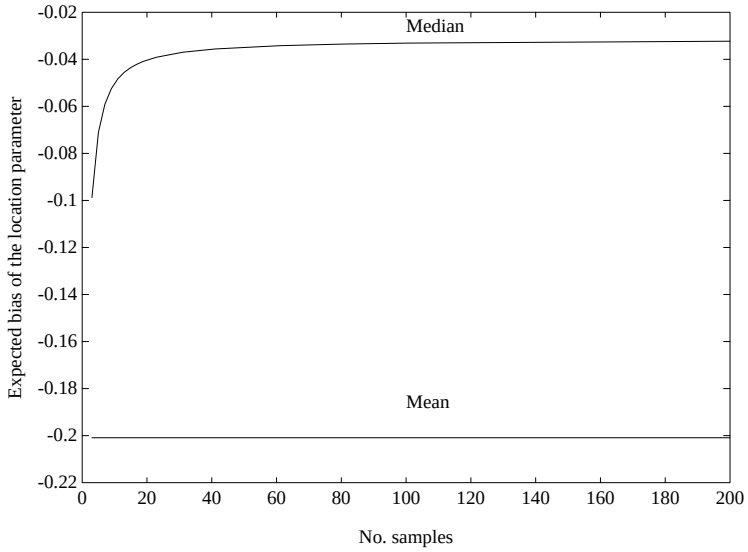
$$f(X)=\frac{1}{3}N(3,\sigma)+\frac{1}{3}N(5,\sigma)+\frac{1}{3}N(10,\sigma) \tag{40}$$

The convergence is the property of a neural network to achieve a stable state after a finite number of iterations in the learning stage. The conver-

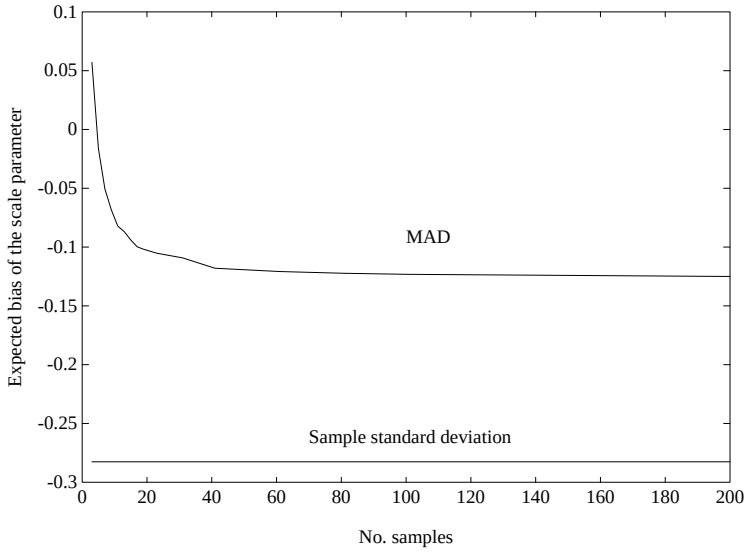
gence can be defined individually for a weight or globally, expressing the state of the network by a cost function. Here we analyze the capability of different MRBF weights to achieve a stable state. Let us assume $\sigma = 2$ in the model of Equation (39). We use MRBF to estimate the parameters of the distribution $N(5, 2)$. We find the expectation for the median by replacing Equation (29) in Equation (26) and by computing the integral numerically. In Figure 2a, we compare the expected bias for the marginal median LVQ against the expected bias of the mean. In Figure 2b, a comparison between the expected bias of the MAD estimator (12) and that of the classical estimator for scale (7) is provided. The expectation for scale parameter using the MAD estimator is obtained after replacing Equation (29) in Equation (33). It is clear from these plots that median and MAD algorithms provide better parameter estimation when estimating overlapping Gaussian functions, than the arithmetic mean and sample deviation.

We estimate the center and the scale parameter for the distribution $N(5, \sigma)$ using both classical and robust type learning for the distributions given in Equations (39) and (40). The class which corresponds to $N(5, \sigma)$ is bounded in the case of Equation (40) and unbounded to the left in the case of Equation (39). Thus in the case of Equation (39), the data samples used for learning are drawn from a "medium-tailed" distribution and in the case of Equation (40) from a "short-tailed" one. The stationary state of the bias from the estimation $E[\hat{\mu}] - \mu$ is depicted in Figure 3a for the distribution (39) and in Figure 3b for the distribution of Equation (40), both with respect to the scale parameter σ. The comparison results for estimating the stationary state of the bias for the scale parameter $E[\hat{\sigma}] - \sigma$ are given in Figure 3c and in Figure 3d. From these plots it is evident that if a certain overlap occur among different Gaussian functions, the respective amount of data samples consist of outliers while median and MAD estimators give less bias than mean and classical sample deviation estimators.

If the Gaussian functions are far away one from each other with respect to the variance, the amount of outliers decreases and both algorithms provide similar results. However, if isolated Gaussian functions are truncated, e.g., due to the decision (4) or (8), then robust estimators are more accurate than those based on classical statistics.



(a) first order statistics estimators



(b) second order statistics estimators

Figure 2. The parameter bias versus the number of data samples in the 1-D case.

We consider four artificially generated distributions, each consisting of two dimensional clusters.

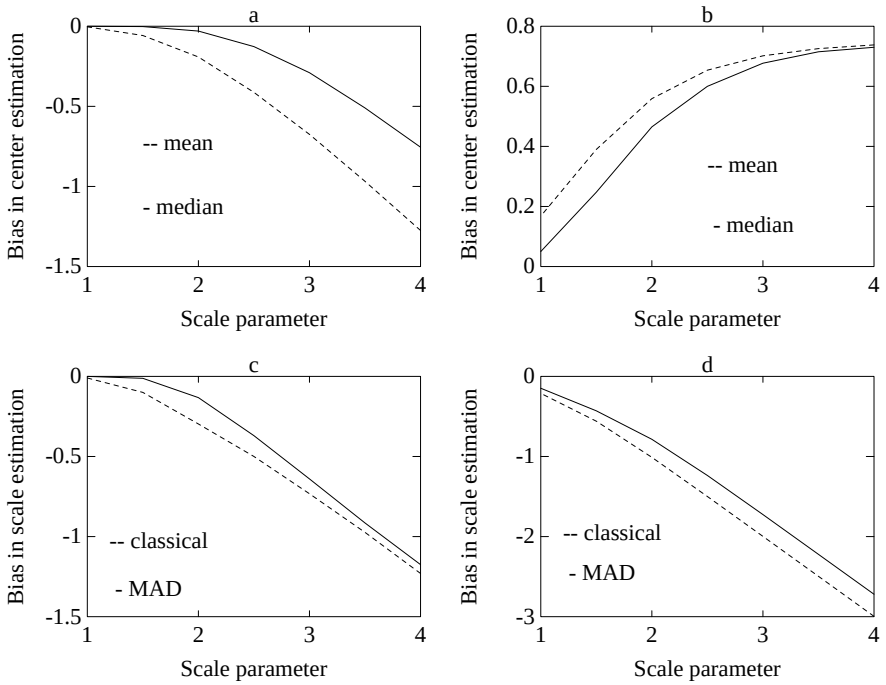


Figure 3. Theoretical analysis of the bias for median type estimators and classical statistics estimators in evaluating the RBF parameters: (a) center for $N(5,\sigma)$ in the distribution of Equation (39) ; (b) center for $N(5,\sigma)$ in Equation (40) ; (c) scale parameter for $N(5,\sigma)$ in Equation (39) ; (d) scale parameter for $N(5,\sigma)$ in Equation (40).

A 2-D Gaussian distribution is denoted by $N(\mu_1,\mu_2;\sigma_1,\sigma_2;r)$, where r is the correlation factor. The Gaussian clusters are grouped in two classes in order to form more complex distributions:

Distribution I:

$$\begin{aligned} P_1^I(X) &= N(2,1;3,1;0) + N(8,7;3,1;0) \\ P_2^I(X) &= N(8,2;1,3;0) + N(2,6;1,3;0) \end{aligned} \tag{41}$$

Distribution II:

$$\begin{aligned} P_1^{II}(X) &= N(6,0;4,1;0) + N(0,6;1,4;0) \\ P_2^{II}(X) &= N(6,6;2,2;0) \end{aligned} \tag{42}$$

Two more distributions are obtained from the first two by adding uniformly distributed data samples:

Distribution III:

$$P_1^{III}(X) = \epsilon\ P_k^I + (1-\epsilon)\ U([-5,15],[-5,15]) \tag{43}$$

Distribution IV:

$$P_1^{IV}(X) = \epsilon\ P_k^{II} + (1-\epsilon)\ U([-5,15],[-5,15]) \tag{44}$$

where $k \in \{1,2\}$ and $\epsilon = 0.9$. We denote by $U([-5,15],[-5,15])$ a uniform distribution having the domain $[-5,15] \times [-5,15]$.

For MRBF we consider a running window of W=401 samples in Equation (13) and topologies of 2-4-2 and 2-3-2 respectively. Both Euclidean and Mahalanobis distances were considered in order to decide which neuron will be updated for a new data sample. Equal numbers of data samples were used in the learning stage for each of the training algorithms. We have tested the ability of classification for both networks after the learning stage was concluded. The miss-classification error compares the given true output $F_k(\mathbf{X})$ with the network output $Y_k(\mathbf{X})$:

$$\{\mathbf{X} \in I\!R^N \mid \exists k\ |F_k(\mathbf{X}) - Y_k(\mathbf{X})| > 0.5\} \tag{45}$$

and is represented as a fraction of the total number of samples. This factor evaluates the accuracy of the Bayesian decision rule implemented by each of the networks. The second comparison criteria is the approximation of the pdf functions, when we implement the respective functions using the parameters estimated by the network.

The optimal network is obtained when the network weights are equal with the parameters of the Distributions I and II (see Equations (41) and (42)). The mean square error (MSE) calculated between the ideal function and the estimated one is defined as:

$$MSE = \frac{1}{M}\sum_{k=1}^{M}\ \int_{\mathcal{D}}(p_k(X) - \hat{p}_k(X))^2\ dX \tag{46}$$

where the domain is $\mathcal{D} = (-\infty,\infty) \times (-\infty,\infty)$ and $\hat{p}_k(X)$ is the hyper surface modeled by the k-th output unit.

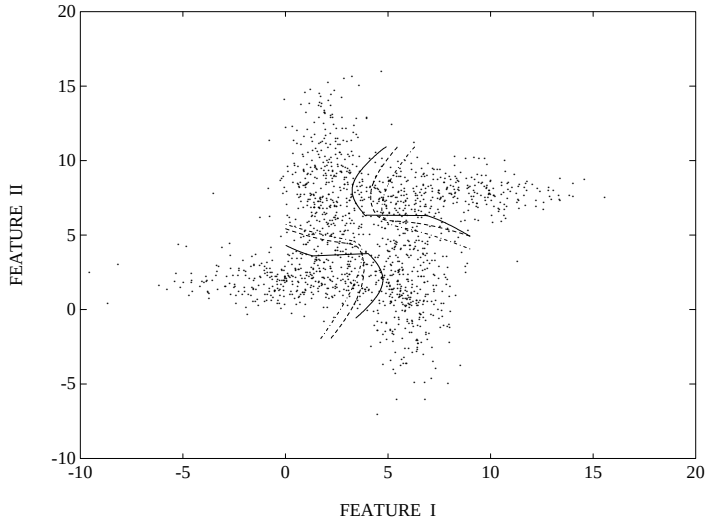


Figure 4. Samples from the distribution I and the boundaries between the classes marked with: '-' for optimal classifier, '- -' for MRBF and '-·' for RBF.

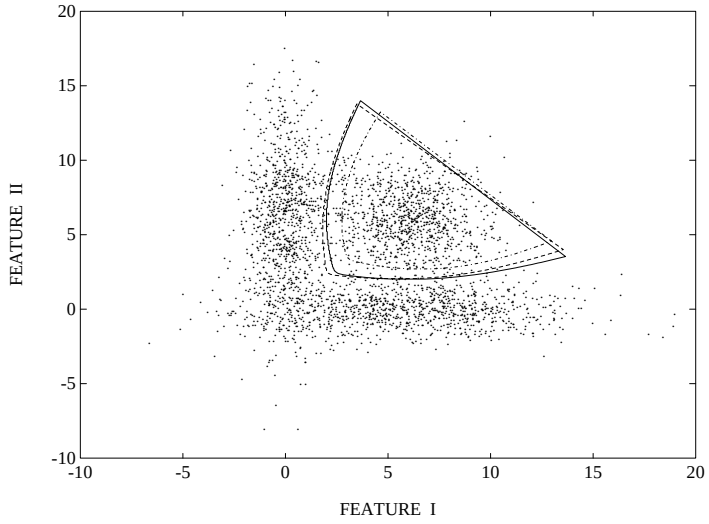


Figure 5. Samples from the distribution II and the boundaries between the classes marked with: '-' for optimal classifier, '- -' for MRBF and '-·' for RBF.

Data samples from the first two distributions are presented in Figures 4 and 5. The same figures display also the boundaries found by means of

Table 1. Comparison between the classical estimation for RBF and the MRBF algorithm.

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
	Optim	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
	Optim	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
	Optim	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
	Optim	7.18	0.00	7.18	0.00

neural networks as well as the optimal boundaries. From these figures it is evident that MRBF approximates better the boundaries between classes than classical statistical estimators for the RBF. The improvement is clear for MRBF in all cases considered in Table 1. However, when the mixture of bivariate normal distributions is contaminated with uniform noise (e.g. in distributions III and IV), the difference is very large because the robust type learning is insensitive at extreme observations. By using the Mahalanobis distance (8) instead of the Euclidean one, we obtain better results for both algorithms, except in the case when we use classical estimators for the uniform contaminated model, Equations (43) and (44). We can see from Table 1 that the MRBF algorithm with the Mahalanobis distance gives the best results.

In Figure 6, we evaluate the global convergence of the algorithms in the case of distribution I. The learning curves represent the estimation of the pdf functions given by MSE (46), with respect to the number of samples drawn. From this plot it is clear that the MRBF network results to a smaller MSE when compared with classical RBF network. The improvement produced by using the Mahalanobis distance instead of the Euclidean distance is also clear from this plot.

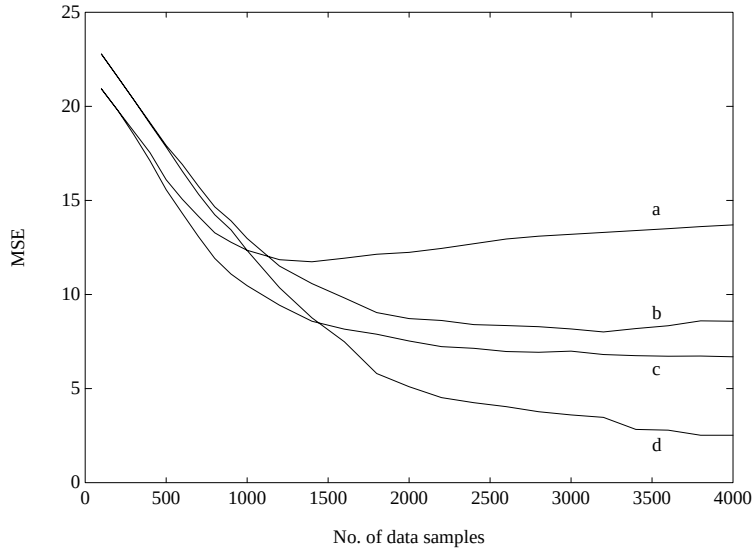
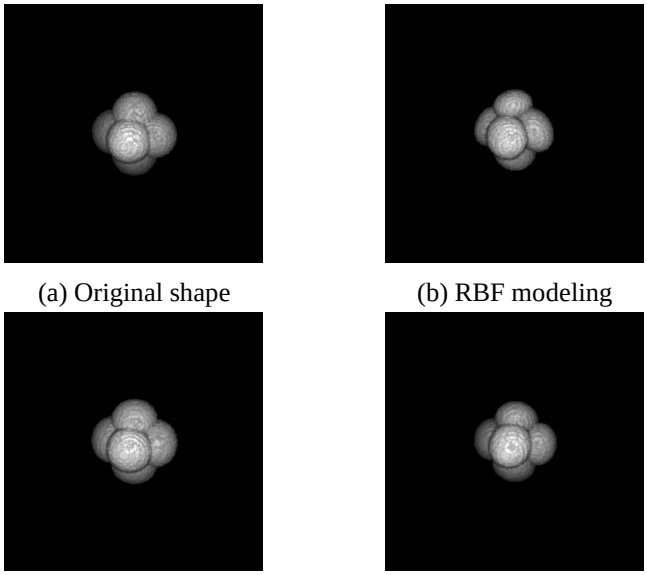


Figure 6. The learning curves in the case when the samples are drawn from the distribution I. We have marked with a the RBF and with b the MRBF when using Euclidean distance and with c, d the RBF and the MRBF when using Mahalanobis distance, respectively.

4 Experimental results

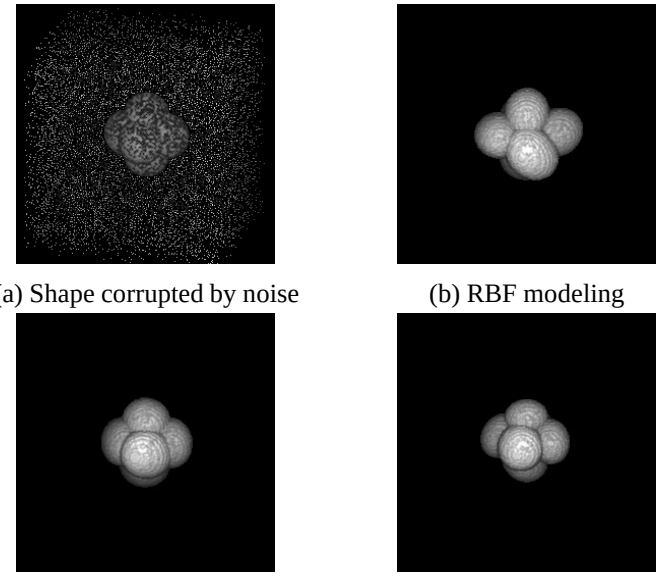
We have tested the robust statistics based training algorithms as described in Section 2, when modeling a synthetic object. We consider the modeling of a 3-D shape consisting of six spheres of constant intensity with the centers located at (40,50,50), (60,50,50), (50,40,50), (50,60,50), (50,50, 40), (50,50,60). The resulting shape is displayed in Figure 7a. The coordinates of the voxels composing the shape are considered as inputs in the neural network. The modeling results when using RBF, Alpha-Trimmed Mean RBF and MRBF network are displayed in Figures 7b, 7c, and 7d, respectively. The algorithm employed for the RBF network has been described in Section 2.1, that for the MRBF network in Section 2.2 and for Alpha-Trimmed Mean RBF in Section 2.3. After training, the shape is reconstructed from the parameters embedded in the neural network. When calculating the spread of the ellipsoids by trimming as in Equation (22) we use a factor in order to compensate for data loss [42], [43]. We distort the shape of the artificial object by adding noise. The noise



(a) Original shape (b) RBF modeling

(c) Alpha-Trimmed Mean RBF modeling (d) MRBF modeling

Figure 7. Modeling a synthetic shape.



(a) Shape corrupted by noise (b) RBF modeling

(c) Alpha-Trimmed Mean RBF modeling (d) MRBF modeling

Figure 8. Modeling a noisy synthetic shape.

Table 2. Modeling a synthetic 3-D shape when using LVQ-based RBF, MRBF and Alpha-Trimmed Mean RBF algorithms.

Algorithm	Noise-free Model		Noisy Model	
	Center Bias Estimation	Modeling Error (%)	Center Bias Estimation	Modeling Error (%)
RBF	2.02	11.59	5.38	59.03
MRBF	3.64	27.03	4.30	32.32
α-Trimmed Mean RBF	1.40	5.99	2.93	20.37

is uniformly distributed in the volume $[0, 100] \times [0, 100] \times [0, 100]$ and has probability of occurrence $0.2\ \%$ in the entire volume. In the locations where the noisy samples are added, the pixel values are switched from the grey level of the object to that of the background and vice versa. The noisy image is displayed in Figure 8a. The objects reconstructed based on the classical training algorithm for RBF, Alpha-Trimmed Mean RBF and MRBF algorithm are displayed in Figures 8b, 8c, and 8d, respectively. Two measures are considered for numerical comparison. The first error criterion measures the average bias in center location:

$$E = \frac{1}{G} \sum_{k=1}^{G} \|\mu_k - \hat{\mu}_k\|^2 \tag{47}$$

where $G = 6$ represents the number of spheres and $\mu_k, \hat{\mu}_k$ are the original and the estimated position of their centers respectively. A second measure is the volume error and it is calculated as the absolute difference between the volume of the original shape (Figure 7a) and the volume representing the reconstructed object based on the estimated parameters, normalized at the size of the original object. The average of the results when applying several times RBF, MRBF and Alpha-Trimmed Mean RBF are provided in Table 2. We can observe from this Table as well as from Figures 7 and 8 that Alpha-Trimmed Mean RBF provides a better result than that of RBF when estimating the original shape from Figure 7a, and better modeling capabilities than MRBF when estimating the parameters from the noisy shape. These results show the capability of the proposed algorithm to model a composite shape as well as its robustness to noise corruption.

5 Conclusions

Robust statistics has been employed in a variety of applications and is known for providing good estimates in the case when data is corrupted by noise. Due to its efficiency and speed in this study, we chose a two stage learning algorithm for training RBF networks. In the first stage the center and the scale parameter of the Gaussian functions are estimated using an approach similar to the learning vector quantization. A general model of training algorithms based on robust statistics is introduced. We present in detail Median RBF and Alpha-Trimmed Mean RBF algorithms. In these algorithms both first and second order statistics are estimated using robust statistics. Algorithms relying on robust statistics are not sensitive to outliers and they are known to provide lesser bias when data is corrupted by noise. Better estimates are obtained for the basis centers and their spread in the case when using robust statistics for estimating the hidden unit parameters, compared with classical training algorithms. We have computed the bias when estimating a mixture of overlapping Gaussians and found that it is smaller for the MRBF network when compared to the RBF network using the classical LVQ algorithm. The Alpha-Trimmed Mean RBF algorithm provides a parameter which regulates the number of data samples to be excluded from estimation. This parameter is calculated from the data distribution and is a function of its tail length. The backpropagation algorithm is used in the second stage for estimating the output weights. Modeling a 3-D shape has been considered for comparing the proposed algorithms. MRBF and Alpha-Trimmed Mean RBF networks proved to be efficient in a variety of artificial or real applications. They have better data modeling and probability estimation capabilities. When the amount of noisy data is big, or the data corresponds to overlapping Gaussian distributions, the difference with respect to the classical approach is very big.

References

[1] Hartman, E.J., Keeler, J.D., and Kowalski, J.M. (1990), "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Computation*, vol. 2, pp. 210-215.

[2] Park, J. and Sandberg, J.W. (1991), "Universal approximation using radial basis functions network," *Neural Computation*, vol. 3, pp. 246-257.

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

[4] Casdagli, M. (1989), "Nonlinear prediction of chaotic time series," *Physica D*, vol. 35, pp. 335-356.

[5] Chng, E.S., Chen, S., and Mulgrew, B. (1996), "Gradient radial basis function networks for nonlinear and nonstationary time series prediction," *IEEE Trans. on Neural Networks*, vol. 7, no. 1, pp. 190-194.

[6] Moody, J. (1990), "Fast learning in networks of locally-tuned processing units," *Neural Networks*, vol. 3, pp. 437-443.

[7] Platt, J. (1991), "A resource-allocating network for functional interpolation," *Neural Computation*, vol. 3, pp. 213-225.

[8] Whitehead, B.A. and Chaote, T.D. (1996), "Cooperative-competitive genetic evolution of radial basis function centers and width for time series prediction," *IEEE Trans. on Neural Networks*, vol. 7, no. 4, pp. 869-880.

[9] Borş, A.G. and Gabbouj, M. (1994), "Minimal topology for a radial basis function neural network for pattern classification," *Digital Signal Processing: a Review Journal*, vol. 4, pp. 173-188.

[10] Haykin, S. (1994), *Neural Networks: a Comprehensive Foundation*, Upper Saddle River, NJ: Prentice Hall.

[11] Tou, J.T. and Gonzalez, R.C. (1974), *Pattern Recognition Principles*, Reading, Massachusetts: Addison-Wesley.

[12] Niranjan, M., Robinson, A.J., and Fallside, F. (1989), "Pattern recognition with potential functions in the context of neural networks," *Proc. of the Scandinavian Conf. on Pattern Recognition*, Oulu, Finland, pp. 96-103.

[13] Cha, I. and Kassam, S.A. (1996), "RBFN restoration of nonlinearly degraded images," *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 964-975.

[14] Kohonen, T.K. (1989), *Self-Organization and Associative Memory*, Berlin: Springer-Verlag.

[15] Specht, D.F. (1990), "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification," *IEEE Trans. on Neural Networks*, vol. 1, no. 1, pp. 111-121.

[16] Broomhead, D.S. and Lowe, D. (1988), "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321-355.

[17] Niranjan, M. and Fallside, F. (1990), "Neural networks and radial basis functions in classifying static speech patterns," *Computer Speech and Language*, vol. 4, pp. 275-289.

[18] Sanner, R.M. and Slotine, J.-J.E. (1992), "Gaussian networks for direct adaptive control," *IEEE Trans on Neural Networks*, vol. 3, no. 6, pp. 837-863.

[19] Chen, S., Cowan, C.F.N., and Grant, P.M. (1991), "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. on Neural Networks*, vol. 2, no. 2, pp. 302-309.

[20] Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., and Hummels, D.M. (1992), "On the training of radial basis function classifiers," *Neural Networks*, vol. 5, pp. 595-603.

[21] Lee, S. and Kil, R.M. (1991), "A Gaussian potential function network with hierarchically self-organizing learning," *Neural Networks*, vol. 4, pp. 207-224.

[22] Howell, J. and Buxton, H. (1995), "Invariance in radial basis function neural networks in human face classification," *Proc. of Int. Workshop on Automatic Face- and Gesture-Recognition*, Zurich, Switzerland, pp. 221-226.

[23] Wei, G.-Q. and Hirzinger, G. (1997), "Parametric shape-from-shading by radial basis functions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 353-365.

[24] Matej, S. and Lewitt, R.M. (1996), "Practical considerations for 3-D image reconstruction using spherically symmetric volume elements," *IEEE Trans. on Medical Imaging*, vol. 15, no. 1, pp. 68-78.

[25] Carr, J.C., Fright, W.R., and Beatson, R.K. (1997), "Surface interpolation with radial basis functions for medical imaging," *IEEE Trans. on Medical Imaging*, vol. 16, no. 1, pp. 96-107.

[26] Widrow, B. and Stearns, S.D. (1985), *Adaptive Signal Processing*, Englewood Cliffs, NJ: Prentice Hall.

[27] Chen, S., Gibson, G.J., Cowan, C.F.N., and Grant, P.M. (1991), "Reconstruction of binary signals using an adaptive radial-basis-function equalizer," *Signal Processing*, vol. 22, pp. 77-93.

[28] Chen, S. and Mulgrew, B. (1992), "Overcoming co-channel interference using an adaptive radial basis function equaliser," *Signal Processing*, vol. 28, pp. 91-107.

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

[30] Seber, G. (1984), *Multivariate Observations*, John Wiley.

[31] Campbell, N.A. (1980), "Robust procedures in multivariate analysis. I: Robust covariance estimation," *Appl. Statist.*, vol. 29.

[32] Hardie, R.C. and Arce, G.R. (1991), "Ranking in R^p and its use in multivariate image estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 1, no. 2, pp. 197-209.

[33] Pitas, I. and Venetsanopoulos, A.N. (1990), *Nonlinear Digital Filters: Principles and Applications*, Kluwer Academic.

[34] Pitas, I., Kotropoulos, C., Nikolaidis, N., Yang, R., and Gabbouj, M. (1996), "Order statistics learning vector quantizer," *IEEE Trans. on Image Processing*, vol. 5, no. 6, pp. 1048-1053.

[35] Hogg, R.V. (1974), "Adaptive robust procedures: a partial review and some suggestions for future applications and theory," *J. Am. Stat. Assoc.*, vol. 69, no. 348, pp. 909-923.

[36] Prescott, P. (1978), "Selection of trimming proportions for robust adaptive trimmed means," *J. Am. Stat. Assoc.*, vol. 73, no. 361, pp. 133-140.

[37] Titterington, D.M. (1978), "Estimation of correlation coefficients by ellipsoidal trimming," *Appl. Stat.*, vol. 27, no. 3, pp. 227-234.

[38] Papoulis, A. (1984), *Probability, Random Variables and Stochastic Processes*, McGraw-Hill.

[39] Borş, A.G. and Pitas, I. (1998), "Optical flow estimation and moving object segmentation based on median radial basis function network," *IEEE Trans. on Image Processing*, vol. 7, no. 5, pp. 693-702.

[40] Giakoumis, I. and Pitas, I. (1998), "Digital restoration of painting cracks," *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS'98)*, Monterey, California, USA.

[41] Chatzis, V., Borş, A.G., and Pitas, I. (1999), "Multimodal decision-level fusion for person authentication," *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 29, no. 6, pp. 674-680.

[42] Borş, A.G. and Pitas, I. (1999), "Object classification in 3-D images using alpha-trimmed mean radial basis function network," *IEEE Trans. on Image Processing*, vol. 8, no. 12, pp. 1744-1756.

[43] Borş, A.G. and Pitas, I. (1998), "Object segmentation in 3-D images based on alpha-trimmed mean radial basis function network," *Proc. European Conference on Signal Processing (EUSIPCO'98)*, vol. II, pp. 1093-1096, Rhodes, Greece.