Global Optimization of RBF Networks

Shimon Cohen and Nathan Intrator
School of Computer Science
Tel-Aviv University
www.math.tau.ac.il∼nin

Abstract

Several modifications to parameter estimation in a Radial Basis Functions network are introduced. These include a better initializing clustering algorithm and a full gradient descent on centers and weights after weights were found via a matrix inversion. Performance comparison with other RBF algorithms is given on several data-sets. It is found that The proposed method was found superior to Bishop's EM training algorithm, to Orr's method [1] for as well as a conventional implementation.

I. Introduction

Radial basis functions have been extensively used for interpolation [2], [3], [4], [5], [6], [7] regression and classification due to their universal approximation properties and simple parameter estimation. The parameter estimation requires a (pseudo) inversion of a (possibly sparse) matrix. The possible numerical instability of the inversion (which is aggravated when the number of training patterns is small compared to the dimensionality) may be partially alleviated by further parameter estimation. While the usefulness of RBF architecture for interpolation is ensured by Micchelli's theorem [8], the approximation properties are less clear especially in high-dimensional and noisy data. In particular, while the approximation and interpolation properties have been extensively studied, e.g., [9], the problem of optimal radial basis function centers has not been resolved. Poggio and Girosi suggest a regularization approach to the problem [10], while others have proposed approaches related to different estimation of the cluster centers [11], [12], [1].

Gaussian kernels RBF network that is trained with the maximum likelihood goal is identical to a mixture of Gaussians model [13], [14]. When kernel centers are found, the determination of the forward weights between the kernels and the output can be done analytically via a pseudo inverse of the hidden-units activity matrix [6]. Flake [15] used the output of the K-Means algorithm and computed the weights of the net using pseudo inverse as above. By setting the radius values of each cluster as the distance to the nearest cluster center, he obtained good results on the vowel data set [16].

While this is a linear problem, the optimal determination of cluster centers is a non-linear and difficult problem. It is very sensitive to the dimensionality of the input space and to the initial starting positions of the clustering algorithm. Bishop had suggested to optimize the cluster centers via the EM algorithm after initial values were found via a clustering algorithm [12]. This was intended to reduce the sensitivity of the algorithm to the clustering problems. The Expectation-Maximization (EM) approach [17] includes the Expectation part which calculates the likelihood of the model, and the Maximization part which maximizes the likelihood of the data with respect to the centers and radii. A recent experimental investigation into several clustering algorithms [18] has shown that the EM approach to clustering [17], [19] outperforms naive center initialization by using random perturbations or using the output of a hierarchical agglomerative clustering

This work was partially supported by the Israeli Ministry of Science and by the Hermann Minkowski – Minerva Center for Geometry at Tel Aviv University. The analysis of the seismic data was supported by a Grant from the Atomic Energy Commission and the Council for Higher Education through Dr. Gideon Leonard. Part of this work was done while N. I. was affiliated with the Institute for Brain and Neural Systems at Brown University and supported in part by ONR grants N00014-98-1-0663 and N00014-99-1-0009.

[20]. A refinement of the random initialization improved the performance of K-Means clustering [21]; They used several small subsamples of the Training Set, and applied the K-Means algorithm for each subsample. The solution is found by further applying the K-Means algorithm to the centers found in the previous phase. This algorithm may be useful for large training sets only. It is also possible to apply the K-Means algorithm to each data class separately [22]. The radius of each cluster center is often calculated as the standard deviation of the data in the "vicinity" of the cluster. When the cluster centers were calculated for each class separately, clusters could have been combined to a larger supercluster when neighboring clusters belonged only the a single class. The assumption about the radial symmetry of each cluster can be relaxed by estimating a different radius for each data axis [23]. Orr uses a regression tree to find the centers and radii of the RBF units. This is done by creating a large tree and then pruning terminal nodes using a criterion like the General Predication Error (GPE) or the Bayesian Information Criterion (BIC). The members in the remaining terminal nodes create clusters, and the diagonal part of the covariance matrix of these members is used as the radii of the cluster in different axes.

The problems of estimating optimal cluster centers can be alleviated by performing post parameter estimation of the full model after the estimation of the forward weights. An initial step in this direction is to perform a gradient descent on the cluster centers and the forward weights but with no change in the cluster radius [24]. They argue that the performance of the network is relatively insensitive to the radius value, an argument that is in contradiction to the findings of [1], [14].

In this paper, we introduce several modifications to the estimation of RBF model parameters. We show that clustering algorithms are sensitive to their initial choice of cluster centers, and propose a better cluster initialization method. We introduce a gradient descent on the full RBF architecture parameters: the centers, the radii, and forward weights. This is done after those parameters were found via clustering and a pseudo matrix inversion. Performance comparison on several benchmark data-sets is given. Comparison includes a conventional RBF implementation[1] [25], Bishop's EM implementation [14], and Orr's regression trees approach [23].

II. A modified clustering algorithm

A Radial Basis Functions approximation network (RBF) is composed of a set of kernel functions ϕ located at cluster centers m_i in input space with a width r. We use Gaussian kernels, as in the other approaches that we compare our results with.

$$\phi(x, m, r) = \exp^{\frac{-(x-m)^2}{2r^2}} . \tag{1}$$

The output of the network, $(y_1, \ldots, y_K)$ is given by:

$$y_k(x) = \sum_{j=1}^{M} w_{kj}\phi(x, x_j, r_j), \tag{2}$$

where M is the number of the RBFs.

When the RBFs are not radially symmetric, the architecture is far more flexible but the number of estimated parameters grows from $M+MN$ to $M+2MN$ [1] and to $O(N^2)$ in the case of estimation of a full Covariance. When the input dimensionality N is large, this complexity is much larger. While it is expected that the added flexibility may be useful, one has to weigh the added flexibility with the reduced robustness (due to increased variance of the model).

The K-Means algorithm seeks to partition the data set into k disjoint subsets S_j containing N_j patterns

[1]Implemented in the MATLAB$^{\text{TM}}$ toolbox.

such that the following criterion is minimized:

$$J(x) = \sum_{i=1}^{K} \sum_{j=1}^{n_j} (x_j - m_i)^2, \tag{3}$$

$$m_i = \frac{\sum_{j=1}^{n_i} x_j}{n_i}. \tag{4}$$

The classical approach [26]. starts with a random selection from the data of initial cluster centers. In the batch version [27] each pattern is then re-assigned to a new cluster which has the nearest center. This procedure is repeated until no further change in the grouping of patterns occurs. Most approaches do not attempt to do better than random initialization of cluster centers [28], [29]. When some of the initial k vectors are chosen from the same cluster, the K-Means algorithm may get stuck in a local minimum which ignores some of the clusters. This is demonstrated in Figure 1, where two cluster centers were initially chosen from the same cluster and eventually, one cluster is not found. We therefore, suggest to start the clustering algorithm from initial centers which are more likely to be in different clusters.

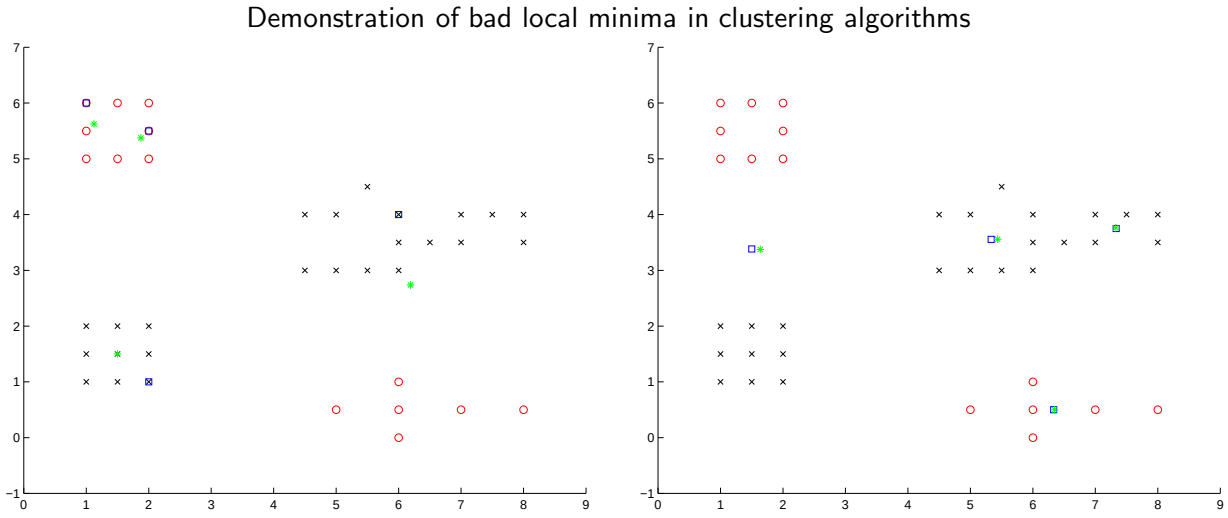

Fig. 1. Data points are the red circles and black crosses; blue rectangles are the initial cluster centers; green stars are the final clusters found. When initial cluster centers are at the same cluster, the K-Means algorithm (left) and the EM clustering algorithm (right) can get stuck in a bad local minima.

A. **Choosing the initial k vectors**

A procedure to choose the initial centers is given here:

1. Choose the first cluster center at random.
2. While the number of clusters is less than k: For-each vector in the the remaining set, computed the nearest cluster center. Choose a vector with the largest distance to the nearest cluster center as the next initial cluster center.

The above procedure increases the probability that the first k cluster centers are not from a smaller number of clusters. This procedure may be iterated few times (with different random initial cluster) to ensure a robust solution. The computational complexity added by the above algorithm is $O(nk)$, where n is the number of data points and k is the number of clusters to be searched in the data. We observe that this added complexity is compensated by an accelerated convergence of the new algorithms.

III. Full architecture parameter estimation

A. Initial computation of the forward weights

After cluster centers were found by a regular k-means or by the above modified k-means algorithm, the forward weights can be analytically found [6]:

$$T = W\Phi^T, \tag{5}$$

$$T^T = \Phi W^T, \tag{6}$$

$$W^T = \mathrm{PInv}(\Phi)T^T, \tag{7}$$

where the design matrix $(\Phi)_{n_j} = \phi_j(X^n)$ and $\mathrm{Pinv}(\Phi)$ is the pseudo inverse of Φ. In practice, the solutions of the linear matrix inversion problem is ill-conditioned due to small variances in some projections of the data. It is solved using a singular value decomposition (SVD).

B. Gradient Descent optimizing centers radii and weights

To summarize, clustering algorithms may find suboptimal solutions due to the following reasons:
1. Bad initial conditions.
2. Wrong number of searched clusters.
3. The search does not take the class labels into account, thus, possibly concentrating on a search for data clusters that have nothing to do with the classification problem.

We therefore, purpose to further estimate the architecture parameters by performing a gradient descent search after the initial parameter estimation. The search should include the cluster centers, the clusters' radii and the forward weights. This concurrent search on all the parameters is non-trivial, as it appears that the force that is driving the cluster radii to zero is stronger than the other optimization forces. Wang and Zhu [24] addressed this problem by assuming a fixed size of the radii and thus, performing the optimization on the remaining parameters, namely the cluster centers and the forward weights.

We address the above problem by applying a constrained optimization, which penalizes small radii. The optimization objective is:

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{M}(y_k^n - t_k^n)^2 + \alpha\sum_{k=1}^{M}\frac{1}{r_k},$$

where r_k is the radius of cluster k. Note that we assume a radially symmetric cluster, this assumption can be relaxed, by performing a local Mahalobis transformation around each cluster. α is a small regularizing parameter, which should be estimated by cross validation on the training data.

Not surprisingly, we have obtained a faster convergence of the optimization algorithm by replacing the gradient descent by a conjugate gradient algorithm. We have used the Polak-Ribeire [30] algorithm which was find to work slightly better than other optimization algorithms that have been suggested in [30], [14]. The gradient optimization formulae are given in the Appendix.

C. Pruning of unnecessary units

The gradient descent algorithm which has been described above produces units which become relatively ineffective. These units, can be easily pruned by evaluating their contribution to the overall error. This is done simply by removing the connections between each hidden unit from the output units, and measuring the change in the objective function. Units which do not increase the error, or increase the error by a value that is smaller than some predetermined regularization constant (which has to be determined via cross-validation on the training data) can be removed. The application of the pruning process, reduced, on average, the number of hidden units by about 1/3 without any significant reduction in performance.

	MacKay	1D Sine	2D Sine	Friedman
Rbf-Orr	0.44 ±0.14	0.44 ±0.14	0.91 ±0.19	0.12±0.03
Rbf-Matlab	0.69±0.41	0.57±0.27	0.74±0.4	0.2 ±0.03
Rbf-Bishop	6.82±0.82	0.33 ±0.16	0.53 ±0.19	0.17 ±0.02
GDRBFN	0.4±0.16	0.33 ±0.12	0.51 ±0.19	0.16 ±0.02

TABLE I

Mean squared error results on test data from four regression data sets. Averaged results over 100 runs are shown including the standard deviation.

IV. Results

This section describes regression and classification results of several variants of RBF and the proposed Gradient Descent RBFN method on several data sets. The results, which are only given for the test data, are an average over 100 runs and include the standard deviation. We start with a comparison on four simulated regression data sets that were used by Orr [1] to asses the performance of RBF. The results are summarized in Table I.

The first data set is a 1D sine wave [1].

$$y = \sin(12x),$$

with $x \in [0,\ 1]$. A Gaussian noise was added to the outputs with a standard deviation of $\sigma = 0.1$. 10 sets of 50 train and 50 test patterns randomly sampled from the data with the additive noise were used.

The second data-set is a 2D sine wave,

$$y = 0.\sin(x_1/4)\sin(x_2/2),$$

with 200 training patterns sampled at random from an input range $x_1 \in [0,\ 10]$ and $x_2 \in [-5,\ 5]$. The clean data was corrupted by additive Gaussian noise with $\sigma = 0.1$. The test set contains 400 noiseless samples arranged as a 20 by 20 grid pattern, covering the same input ranges. Orr measured the error as the total squared error over the 400 samples. We use the same measure.

The third data set [31], [23] is based on a one dimensional Hermite polynomial

$$y = (1 + (x + 2x^2)e^{-x^2}.$$

100 input values are sampled randomly between $-4 < x < 4$, and Gaussian noise of standard deviation $\sigma = .1$ was added to the output.

The fourth data-set is a simulated alternating current circuit with four input dimensions (resistance R, frequency ω, inductance L and capacitance C and one output impedance $Z = \sqrt{R^2 + (\omega L - 1/\omega C)^2}$. Each training set contained 200 points sampled at random from a certain region [23, for further details]. Again, additive noise was added to the outputs. The experimental design is the same as the one used by Friedman in the evaluation of MARS [32]. Friedman's results include a division by the variance of the test set targets. We follow Friedman and report the NMSE on the test set. Orr's regression trees method [23] outperforms the other methods on this data set.

A. Classification

The first classification data set is of sonar returns where an attempt is made to distinguish between a mine and a rock. It was used by Gorman and Sejnowski [33] in their study of the feasibility of neural networks to sonar signal discrimination. The data has 60 continuous inputs and one binary output for the two classes.

Algorithm	Sonar	Vowel	Seismic1	Seismic2	breast-cancer
RBF-Orr	75±0	–	63±0	79±0	95.86±0.003
RBF-Matlab	85±1	51.6±2.9	73±4	81±3	96.02±0.006
RBF-Bishop	82±3	48.4±2.4	60±4	77±5	96.73±0.003
GDRBFN	91±1	65.1±1.2	84±0	85±0	96.75 ±0.003

TABLE II

PERCENT CLASSIFICATION RESULTS USING THE RBF DIFFERENT ARCHITECTURES ON FOUR DATA SETS. AVERAGE RESULTS OVER 100 RUNS INCLUDING THEIR STANDARD DEVIATIONS ARE PRESENTED.

It is divided into 104 training patterns and 104 test patterns. The task is to train a network to discriminate between sonar signals that are reflected from a metal cylinder and those that are reflected from a similar shaped rock. Gorman and Sejnowski report on results with feed-forward architectures [34] using 12 hidden units. They achieved 90% correct classification on the aspect independent test data. This result outperforms the results obtained by the different RBF methods, and is only surpassed by our proposed RBF training algorithm.

The Deterding vowel recognition data [16], [15] is a widely studied benchmark. This problem may be more indicative of the type of problems that a real neural network could be faced with. The data consists of auditory features of steady state vowels spoken by British English speakers. There are 528 training patterns and 462 test patterns. Each pattern consists of 10 features and it belongs to one of 11 classes that correspond to the spoken vowel. The speakers are of both genders. The best score so far was reported by Flake using his SMLP units. SMLP is a hybrid neural network which uses nonlinear as well as linear input features resulting from the expansion of the Euclidean distance around a cluster center:

$$(x-c)^t(x-c) = x^t x - 2x^t c + c^t c,$$

where c is the center of a radial function and x is an input pattern vector. Flake has shown that this kind of network can approximate projection units as well as radial units. The sqaure inputs approximate the radial functions, while the projection it the regular inputs. His average best score was 60.6% [15] and was achieved with 44 hidden units. Our algorithm achieved 65.1% correct classification with only 27 hidden units. As far as we know, it is the best result that was achieved on this data set. Orr's method is not included as it was not available for a multi-class problem.

Seismic1 and seismic2 are two different representations of seismic signals. The data sets include waveforms from two types of explosions and the task is to distinguish between the two types. This data was used for evaluation of various approaches including artificial neural networks in a "Learning" course given at Tel-Aviv University[2]. The dimensionality of seismic1 is 352 representing 32 time frames of 11 frequency bands, and the dimensionality of seismic2 is 242 representing 22 time frames of 11 frequency bands. Principal Component Analysis (PCA) was used to reduce the data representation into 12 dimensions. Both data-sets have 65 training patterns and 19 test patterns which were chosen to be the most difficult for the desired discrimination. Table II summarizes the percent correct classification results on both data sets for the different RBF classifiers. The test set was carefully chosen to include the most difficult discrimination tasks. On the seismic data, due to the use of a single test set, the STD is often zero as only a single classification of the data was obtained in all 10 runs.

[2]For details see `http://www.math.tau.ac.il~nin/learn98,9`

V. Summary

We have demonstrated a possibly strong deficiency of the familiar k-means algorithm and its variants including the EM variant. We have suggested a simple way to amend this problem. The analysis performed in Appendix B demonstrates that the problem of bad initial cluster centers becomes worse when the number of training patterns increases. It is thus more important to correct for this problem in those cases when one expects the size of the data set to be sufficient for robust parameter estimation.

To further improve on the results of a clustering algorithm, we have suggested to perform a constrained optimization on all the parameters or a Radial Basis Function architecture. The optimization has to be constrained so as to avoid a trivial strongly overfitting solution, where the cluster centers converge to zero. We have demonstrated the improved performance of the resulting algorithm on 8 data sets and compared our approach with the current state of the art in RBF training algorithms. Our results demonstrate that the proposed training algorithm often outperforms the competing algorithms and never underperforms; A remarkable result is achieved on the vowel data set.

Appendix

I. Full gradient formulae

We provide the equations for the full error gradient of the RBF networks, including the gradient of the radii, the forward weights and the cluster centers. Let E^n be the error due to the n training pattern, the total error is given by

$$E = \sum_{n=1}^{N} E^n. \tag{8}$$

$$E^n = \frac{1}{2} \sum_{k=1}^{M} (y_k(w; x^n) - t_k^n)^2. \tag{9}$$

We define:

$$\delta_k^n = (y_k(w; x^n) - t_k^n). \tag{10}$$

The derivatives of the error with respect to the weights are given by:

$$\frac{\partial E^n}{\partial w_{ij}} = \sum_{n=1}^{N} (\delta_i^n \phi_j^n). \tag{11}$$

The derivatives of the error with respect to cluster center i is:

$$\frac{\partial E^n}{\partial m_i} = \sum_{k=1}^{C} (\delta_k^n w_{ki} \partial \phi_i / \partial m_i), \tag{12}$$

where C is the number of the network outputs. The derivatives of the error function with respect to radius i is:

$$\frac{\partial E^n}{\partial r_i} = \sum_{k=1}^{C} (\delta_k^n w_k i \partial \phi_i / \partial r_i) \tag{13}$$

The derivatives of the RBF function with respect to center i:

$$\frac{\partial \phi_i}{\partial m_i} = \phi_i(x^n) \frac{x^n - m_i}{r_i^2}. \tag{14}$$

The derivatives of the RBF function with respect to the radius i:

$$\frac{\partial \phi_i}{\partial r_i} = \phi_i(x^n) (\frac{\| x^n - m_i \|^2}{r_i^3} - \frac{d}{r_i}), \tag{15}$$

where d is the dimension of the input pattern vector.

II. Upper bound on the probability of K-Means to get stuck in bad minima

The following derivation also holds for the clustering done by the EM algorithm [14]. Here we derive the probability of some vectors to be chosen from the same cluster. Assuming that there are exactly k distinct clusters, let A be the event that the k initial vectors are chosen from k distinct clusters. Assuming that each cluster has roughly the same number of members[3] dentoted by n/k (rounded to the smallest integer), there are $(n/k)^k$ different ways to choose the k initial vectors. There are $\binom{n}{k}$ different ways to choose k vectors out of the total n number of vectors. Thus,

$$\begin{aligned} p(A) &= (n/k)^k / \binom{n}{k}, \\ &= \frac{n^k (n-k)! k!}{k^k n!}. \end{aligned} \tag{16}$$

We proceed by using the Stirling approximation of $n!$ for large n:

$$n! \simeq \sqrt{2\Pi n} n^n e^{-n}. \tag{17}$$

The Stirling approximation gives:

$$p(A) \simeq \frac{n_i^k \sqrt{2\Pi(n-k)}(n-k)^{n-k} e^{-(n-k)} \sqrt{2\Pi k} k^k e^{-k}}{\sqrt{2\Pi n} n^n e^{-n}}. \tag{18}$$

After some arrangement we arrive at:

$$p(A) \simeq \frac{n_i^k \sqrt{2\Pi(n-k)} \sqrt{2\Pi k}(n-k)^n k^k}{\sqrt{2\Pi n} n^n (n-k)^k}. \tag{19}$$

Since $n \gg k > 0$ we make use $(n-k)^{(n-k)} < (n-k)^n$ and arrive at

$$p(A) <= \frac{n_i^k \sqrt{2\Pi(n-k)} \sqrt{2\Pi k} k^k}{\sqrt{2\Pi n}(n-k)^k} \tag{20}$$

$$\Rightarrow P(A) \underset{n \to \infty}{\longrightarrow} 0. \tag{21}$$

Thus, the probability to start with bad local initial cluster centers approaches 1 when the number of data points becomes large. Starting with bad initial cluster centers does not always guarantee that the final solution will not include all clusters, therefore the above analysis gives an upper bound for the probability of arriving to bad cluster solutions. The EM algorithm can not alleviate this problem, as was demonstrated in Figure 1. For that example, the upper bound is 7/9.

References

[1] M. J. Orr, "Regularisation in the selection of RBF centres," *Neural Computation*, vol. 7, no. 3, pp. 606–623, 1995.

[2] R. L. Hardy, "Multiquadratic equations of topography and other irregular surfaces," *J. Aircraft*, vol. 9, pp. 189–191, 1971.

[3] A. Agterberg, *Geomathematics*, Elsevier, Amsterdam, 1974.

[4] N. Dyn, D. Levin, and S. Rippa, "Numerical procedures for surface fitting of scattered data by radial functions," *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 2, pp. 639–659, 1986.

[5] M. J. D. Powell, "Radial basis functions for multivariable interpolation: A review," in *Algorithms for Approximation*, J. C. Mason and M. G. Cox, Eds., pp. 143–167. Clarendon Press, Oxford, 1987.

[3] The alternative assumption will further reduce the probability of A.

[6] D.S. Broomhead and D. Lowe, "Multivariate functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[7] N. Dyn and C. A. Micchelli, "Interpolation by sums of radial functions," *Numerische Math.*, vol. 58, pp. 1–9, 1990.

[8] C. A. Micchelli, "Interplation of scattered data: distance matrices and conditionally positive definite functions," *Constructive Approximations*, vol. 2, pp. 11–22, 1986.

[9] N. Dyn and A. Ron, "Radial basis function approximation: from gridded centers to scattered centers," in *Proc. London Math. Soc.*, 1995.

[10] T. Poggio and F. Girosi, "Networks for approximation and learning," *IEEE Proceedings*, vol. 78, no. 9, pp. 1481–1497, 1990.

[11] J. Moody and C. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computation*, vol. 1, pp. 281–289, 1989.

[12] C. M. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–588, 1991.

[13] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, 1985.

[14] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995.

[15] G. W. Flake, "Square unit augmented, radially extended, multilayer percpetrons," in *Neural Networks: Tricks of the Trade*, G. B. Orr and K. Müller, Eds., pp. 145–163. Springer, 1998.

[16] D. H. Deterding, *Speaker Normalisation for Automatic Speech Recognition*, Ph.D. thesis, University of Cambridge, 1989.

[17] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Proceedings of the Royal Statistical Society*, vol. B-39, pp. 1–38, 1977.

[18] M. Meila and D. Heckerman, "An experimental comparison of several clustering methods," Technical report tr-98-06, Microsoft, 1998, www.research.microsoft.com/∼heckerman.

[19] G. Celeux and G. Govart, "A classification em algorithm for clustering and two stochasitc versions," *Computational statistics and data analysis*, vol. 14, pp. 315–332, 1992.

[20] J. Banfield and A. Raftery, "Model based Gaussian and non-Gaussian clustering," *Biometrics*, vol. 49, pp. 803–821, 1993.

[21] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," in *International Conference on Machine Learning*. July 1998, Morgan kaufmann, www.research.microsoft.com/∼fayyad/papers/icml98.htm.

[22] L. Bruzzone and D. F. Prieto, "A technique for the selection of kernel-function parameters in RBF neural networks for classification of remote-sensing images," *IEEE Trans. Geosci.*, vol. 37, no. 2, pp. 1179–1184, 1999.

[23] M. J. Orr, J. Hallam, A. Murray, and T. Leonard, "Assessing rbf networks using delve," *IJNS*, 2000.

[24] Zheng ou Wang and Tao Zhu, "An efficient learning algorithm for improving generalization performance of radial basis function neural networks," *Neural Neworks*, vol. 13, no. 4,5, 2000.

[25] P. D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, New York, 1993.

[26] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley, New York, 1973.

[27] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.

[28] L. Kaufman and P. Rousseeuw89, *Finding groups in data*, John Wiley and Sons, New York, 1989.

[29] E. Rasmussen, "Clustering algorithms," in *Information retrieval data structres and algorithms*, F. Yates and B. Yates, Eds., pp. 419–442. Prentice Hall, NJ, 1992.

[30] E. Polak, *Computational Methods in Optimization. A Unified Approach*, New York: Academic Press., 1971.

[31] D. J. C. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992.

[32] J. H. Friedman, "Mutltivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, pp. 1–141, 1991.

[33] Gorman R. P. and Sejnowski T. J., "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Network*, pp. 75–89, 1988, Vol. 1.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds., vol. 1, pp. 318–362. MIT Press, Cambridge, MA, 1986.