

# Radial Basis Functions for Process Control

Lyle H. Ungar                      Tom Johnson                      Richard D. De Veaux  
University of Pennsylvania    Voice Processing Corp.    Princeton University

## Abstract

Radial basis function (RBFs) neural networks provide an attractive method for high dimensional nonparametric estimation for use in nonlinear control. They are faster to train than conventional feedforward networks with sigmoidal activation networks (“backpropagation nets”), and provide a model structure better suited for adaptive control. This article gives a brief survey of the use of RBFs and then introduces a new statistical interpretation of radial basis functions and a new method of estimating the parameters, using the EM algorithm. This new statistical interpretation allows us to provide confidence limits on predictions made using the networks.

KEYWORDS: radial basis functions, nonparametric regression; function approximation; nonlinear control.

## 1 Introduction

Radial basis functions (RBFs), are a form of neural network<sup>1</sup> for approximating nonlinear relationships. They typically take the form

$$\mathbf{y} = \sum_{j=1}^k \mathbf{w}_j \phi_j(\mathbf{x}) \quad (1)$$

where the functions  $\phi_j(\mathbf{x})$  are Gaussian basis functions with centers  $\mu_j$  and widths  $\sigma_j$  or, as we will interpret them in this paper, Gaussian density functions with mean  $\mu_j$  and standard deviation  $\sigma_j$ .

In control applications, the mapping the RBFs represent could be either a model of a plant (where e.g.  $\mathbf{x}$  is the current state of the plant and the current control action and  $\mathbf{y}$  is the next state of the plant) or a controller (where e.g.  $\mathbf{x}$  is the current state of the plant and the desired next state of the plant and  $\mathbf{y}$  is the control action to be taken). In real applications, the true state is generally not measurable, so an ARMA-style input consisting of a window of current and past measurements is used. Once one has a plant model, it can be incorporated into any model-based nonlinear control scheme such as MPC [Psichogios and Ungar, 1991; Hunt et al., 1992].

---

<sup>1</sup>The term “neural network” is sometimes controversial, but radial basis functions are loosely inspired by parts of the visual cortex, where signals from neighboring receptors in the eye are combined with a roughly Gaussian grouping.

Radial basis functions share many of the advantages of conventional feedforward neural networks (“backpropagation networks”). In the limit, both types of networks, with enough neurons and enough data, can approximate any well-behaved function arbitrarily well [Poggio and Girosi, 1990]. It has been claimed that RBFs are superior in that they, unlike feedforward networks, possess the property of “best approximation”: in the set of approximating functions spanned by the set of adjustable parameters, there is one function that has minimum distance from any given function in the larger set of functions being approximated [Girosi and Poggio, 1989]. In practice, RBFs are often superior to backpropagation networks when the input is of relatively low dimension (5-10 inputs), but are usually inferior for high dimensional problems (over 20 inputs).

RBFs offer a number of advantages over backpropagation networks. As we will show below, RBFs lend themselves to producing error bars on their predictions. They can easily be used to learn model mismatch when an approximate first principles model is available [Kramer et al., 1992]. And, most attractively for adaptive control purposes, if one fixes the basis functions (i.e. picks the centers and widths of Gaussians), then the predictions are linear in the coefficients (the weights). These weights can then be adapted, and all of the standard mathematics of linear systems can be applied. There is a large literature on the use of RBFs for control applications; see, for example, Parthasarathy and Narendra [1991] and Sanner and Slotine [1992].

A number of different methods have been used to pick the centers and widths of the basis functions. The simplest is to pick a fixed set of basis functions, for example with evenly spaced centers. This is, in general, a bad idea, as it fails to provide one of the most important properties of radial basis functions. When one is modeling a system with multiple inputs, picking basis functions adaptively, i.e. based on the data at hand, gives much more accurate models with less data than using fixed basis functions.

More precisely, Barron [1993] has shown that when models are estimated using a fixed set of basis functions (e.g. a Fourier series, polynomials, or Gaussians that are picked without examining the data), the approximation error scales as

$$\text{error} \sim n^{-1/d} \tag{2}$$

where  $n$  is the number of basis functions and  $d$  is the dimension of the input vector, whereas when models are estimated using basis functions picked based on the data (as in methods like MARS, backprop nets, or RBFs,) the error scales as

$$\text{error} \sim n^{-1/2} \tag{3}$$

When the dimension of the input space is large, picking the basis functions appropriately clearly offers a major advantage in that far fewer basis functions are needed for the same accuracy. The most widely used method of selecting basis functions for RBFs is to use a k-means clustering on the  $\mathbf{x}$ 's [Moody and Darken, 1989], although other clustering methods have been used (See e.g. Musavi et al., 1992 for the case where RBFs are used as classifiers).

This paper introduces a new method of picking basis functions which takes into account both the  $\mathbf{x}$ 's and the  $\mathbf{y}$ 's. We call this method EMRBF: Expectation Maximization Radial Basis Functions. The next section introduces a statistical interpretation of RBF's as a mixture of Gaussians, and shows how this leads to an efficient algorithm for simultaneously estimated the centers and widths,

$\mu_j$  and  $\sigma_j$ , of the basis functions and the coefficients,  $\mathbf{w}_j$ . The final section describes some of the other benefits that follow from the EMRBF formulation.

## 2 EMRBF

### Mixture Model Formulation

One way of describing radial basis functions is to view them as mixtures of Gaussians. This allows a statistical interpretation which is less ad hoc than standard RBFs and allows the application of the EM algorithm. Assume that one has a set of  $n$  data points,  $\mathbf{z}_i$ , each drawn from one of  $k$  different populations, with probability  $\lambda_j$ , where the subscript  $j$  denotes the population. Further assume that for each population  $j$ , the  $\mathbf{z}_i$  are distributed as multivariate Gaussians with mean  $\boldsymbol{\nu}_j$  and covariance  $\boldsymbol{\Sigma}_j$ ; i.e.  $\mathbf{z}_i \sim N(\boldsymbol{\nu}_j, \boldsymbol{\Sigma}_j)$ . We will show that under specific assumptions about the form of the covariance matrix this gives a radial basis function.

One can consider the points  $\mathbf{z}_i$  as being composed of two parts,  $\mathbf{x}_i$  and  $\mathbf{y}_i$ ,  $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ , where  $\mathbf{x}$  represents inputs to a function  $\mathbf{y} = f(\mathbf{x})$ . If one takes  $\boldsymbol{\nu}_j = (\boldsymbol{\mu}_j, \mathbf{w}_j)$ , i.e. the means of the  $\mathbf{x}$ 's and  $\mathbf{y}$ 's produced by the  $j$ -th population are  $\boldsymbol{\mu}_j$  and  $\mathbf{w}_j$ , respectively, and assumes that the  $\mathbf{x}$ 's and  $\mathbf{y}$ 's are uncorrelated and have variance  $\sigma_j^2$  and  $\phi_j^2$ , then we will show that one can calculate the expected value of the  $\mathbf{y}$  corresponding to a new  $\mathbf{x}$  by

$$\hat{\mathbf{y}} = \sum_{j=1}^k \mathbf{w}_j P_j(\mathbf{x}) \quad (4)$$

where

$$P_j(\mathbf{x}) = \frac{\frac{\lambda_j}{\sigma_j} \exp\left\{\frac{\|\mathbf{x} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right\}}{\sum_{t=1}^k \frac{\lambda_t}{\sigma_t} \exp\left\{\frac{\|\mathbf{x} - \boldsymbol{\mu}_t\|^2}{2\sigma_t^2}\right\}} \quad (5)$$

This is, of course, equivalent to a radial basis function in which the basis functions have been normalized. The weights,  $\mathbf{w}_j$ , are the expected values of the  $\mathbf{y}$ 's and the centers and widths of the radial basis functions are the means and the standard deviations of the  $\mathbf{x}$ 's.

We can use this statistical interpretation to derive an EM algorithm for simultaneously calculating the basis functions as the weights. As above, assume  $k$  populations and  $n$  data points

$$\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i) \quad (6)$$

where each  $\mathbf{z}_i$  is an element of population  $j$  with probability  $\tau_{ij}$ . The probability density function for  $\mathbf{z}_i$  is taken to be given by

$$f_j(\mathbf{z}_i) = N(\boldsymbol{\nu}_j, \boldsymbol{\Sigma}_j) \quad (7)$$



The means and variances of the different populations (i.e. the centers and widths of the basis functions) are:

$$\hat{\boldsymbol{\mu}}_j = \sum_{i=1}^n \hat{\tau}_{ij} \mathbf{x}_i / n \hat{\lambda}_j \quad (14)$$

and

$$\hat{\sigma}_j^2 = \sum_{i=1}^n \hat{\tau}_{ij} (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j)' (\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j) / n \hat{\lambda}_j \quad (15)$$

and the means and variances of the predictions of  $\mathbf{y}$  for each population are:

$$\hat{\mathbf{w}}_j = \sum_{i=1}^n \hat{\tau}_{ij} \mathbf{y}_i / n \hat{\lambda}_j \quad (16)$$

and

$$\hat{\phi}_j^2 = \sum_{i=1}^n \hat{\tau}_{ij} (\mathbf{y}_i - \hat{\mathbf{w}}_j)' (\mathbf{y}_i - \hat{\mathbf{w}}_j) / n \hat{\lambda}_j \quad (17)$$

Finally, the posterior probability that data point  $\mathbf{z}_i$  is an element of population  $j$  is estimated by:

$$\hat{\tau}_{ij} = \frac{\frac{\hat{\lambda}_j}{\hat{\sigma}_j^l \hat{\phi}_j^m} \exp \left\{ -\frac{1}{2\hat{\sigma}_j^2} \|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_j\|^2 - \frac{1}{2\hat{\phi}_j^2} \|\mathbf{y}_i - \hat{\mathbf{w}}_j\|^2 \right\}}{\sum_{t=1}^k \frac{\hat{\lambda}_t}{\hat{\sigma}_t^l \hat{\phi}_t^m} \exp \left\{ -\frac{1}{2\hat{\sigma}_t^2} \|\mathbf{x}_i - \hat{\boldsymbol{\mu}}_t\|^2 - \frac{1}{2\hat{\phi}_t^2} \|\mathbf{y}_i - \hat{\mathbf{w}}_t\|^2 \right\}} \quad (18)$$

where  $l$  is the length of  $\mathbf{x}$  and  $m$  is the length of  $\mathbf{y}$ .

## The EM Algorithm

The EM algorithm is an iterative approach to maximum likelihood estimation. Each iteration of an EM algorithm is composed of two steps, which for this problem take the following form: an Expectation (E) step in which given assumed centers and widths for the basis functions, one determines the probability that each point in the training set comes from each Gaussian, and a Maximization (M) step in which the centers and widths of the Gaussians and the means and standard deviations of each of their outputs are estimated, assuming that the assignment of data points to experts is known. In statistical terms, the M step involves maximization of a likelihood function that is redefined in each iteration of the E step.

The EM algorithm is most easily understood in its standard clustering application. If one *knows* which cluster each point is in, it is trivial to find the center and width (standard deviation) of the clusters. Similarly, if one knows the center and width of each cluster, one can determine the probability that each point belongs to a particular cluster by a likelihood ratio of the respective densities. This alternation between the E and M step forms the basis of the EM algorithm. See Dempster, Laird, & Rubin (1977) for more detail. The algorithm given here also takes into account the fact that we can use the  $\mathbf{y}$ 's as well as the  $\mathbf{x}$ 's to determine the clusters.

The detailed algorithm is as follows:

To initialize, specify  $k$ , the number of populations, and specify the starting values:

Choose all subpopulations to be of equal size,

$$\lambda_j = 1/k \quad (19)$$

choose the means to be evenly spaced over the range of values that  $\mathbf{x}$  and  $\mathbf{y}$  take on,

$$\boldsymbol{\mu}_j = \min(\mathbf{x}) + (j/k) \text{ range}(\mathbf{x}) \quad (20)$$

$$\mathbf{w}_j = \text{avg}(\mathbf{y}) \quad (21)$$

and choose all of the variances for the subpopulations to be equal to the population variances

$$\sigma_j = \text{std}(\mathbf{x})/k \quad (22)$$

$$\phi_j = \text{std}(\mathbf{y})/k \quad (23)$$

where “std” indicates standard deviation.

After initializing, iterate as follows:

- (1) Compute  $\hat{\tau}_{ij}$  from (18)
- (2) Compute  $\hat{\lambda}_j^{(t)}$ ,  $\boldsymbol{\mu}$ ,  $\mathbf{w}$ ,  $\sigma$ ,  $\phi$  from Equations (13–17)
- (3) Check convergence using the change in the log likelihood

$$\mathcal{L}_0(\mathbf{x}) = \sum_{i=1}^n \log(f(\mathbf{z}_i)) \quad (24)$$

where  $f(\mathbf{z}_i)$  is given by equation 9 and depends on all the parameters  $\boldsymbol{\mu}_j, \mathbf{w}_j, \sigma_j, \phi_j$ .

Continue iterating until the change is less than some cutoff.

Once the model has been learned, if one knows both  $\mathbf{x}$  and  $\mathbf{y}$  then equation 18 gives an estimate of the probability that a point  $\mathbf{z}_i$  is in the population  $j$ . In the more usual case where  $\mathbf{x}$  is known and one wishes to estimate  $\mathbf{y}$ , first estimate the probability that  $\mathbf{x}_i$  is in population  $j$  using

$$p_{ij} = \frac{\frac{\lambda_j}{\sigma_j} \exp\left\{-\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}\right\}}{\sum_{t=1}^k \frac{\lambda_t}{\sigma_t} \exp\left\{-\frac{\|\mathbf{x}_i - \boldsymbol{\mu}_t\|^2}{2\sigma_t^2}\right\}} \quad (25)$$

Then use the above probability and the estimate of  $\mathbf{w}_j$  from Eqn. 16 in

$$\hat{\mathbf{y}} = E(\mathbf{y}_i | \mathbf{x}_i) = \sum_{j=1}^k p_{ij} \mathbf{w}_j \quad (26)$$

where  $\mathbf{w}_j$  can be interpreted as the expected value of  $\mathbf{y}_i$  given that  $\mathbf{y}_i$  is an element of  $j$ . The variance of  $\mathbf{y}_i$  is estimated using:

$$\text{Var}(\mathbf{y}_i | \mathbf{x}_i) = \sum p_{ij} \phi_j^2 \quad (27)$$

### 3 Discussion

We have presented a modified RBF architecture and developed a learning algorithm for this architecture within the framework of maximum likelihood estimation. Radial basis functions offer several advantages over conventional neural network approaches. They give a relatively fast and simple representation for nonlinear models for process control, and have coefficients which are relatively easy to interpret and which can be modified by adaptive control algorithms which require linear parameters. The EM algorithm presented above is a significant improvement over previous methods of fitting the RBF parameters, giving rapid convergence to a guaranteed local optimum in the centers and widths of the radial basis functions, as well as their coefficients.

An appealing attribute of the EMRBF architecture is that it allows the application of standard tools from statistical theory. For example, as part of the model, a local error estimate is generated, and so that for each new point presented to the network, a local confidence limit is generated, as is a probability that the data point is from a region in which data have been previously seen. This later feature allows the network to warn the user when extrapolation is attempted [Leonard, Kramer and Ungar, 1992]

Further advantages of the EMRBF mixture of Gaussians formulation include that EMRBF:

1. Uses both  $\mathbf{x}$  and  $\mathbf{y}$  data in determining RBF clusters.
2. Can use fast and robust EM. [DeVeaux and Krieger, 1989]
3. Gives estimates of local variance.
4. Has natural extensions to pick the number of basis functions.
5. Allows users to fit the model without being concerned about which variables will be taken to be independent and which to be dependent.
6. Gives a natural mechanism for “inverting” the network. If the network was trained to predict  $y_{t+1}$  given  $y_t$  and  $u_t$ , then it can also be used to estimate  $y_t$  given  $y_{t+1}$  and  $u_t$ . This, in effect, computes the control action.

The EMRBF formulation also suggests many extensions. Techniques can be drawn from the statistics literature for selecting the optimal number of basis functions. Robust estimation methods can be used. We are currently working in these directions, and on developing an online version of the algorithm in which we monitor new points for being novel and re-estimate number of Gaussians when sufficient novel points have been found.

### References

- [1] A.R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

- [2] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 93:1–88, 1977.
- [3] F. Girosi and T. Poggio. Networks and the best approximation property, C.B.I.P. Paper No. 45, Massachusetts Institute of Technology, 1989.
- [4] K.J. Hunt, D. Sbarbaro, R. Zbikowski, and P.J. Gawthrop. Neural Networks for control-systems—A survey. *Automatica*, 28(6):1083–1112, 1992.
- [5] M.A. Kramer, M.L. Thompson, and P.M. Bhagat. Embedding theoretical models in neural networks. *Proceedings of the 1992 ACC* 475–479, 1992.
- [6] J.A. Leonard, M.A. Kramer, and L.H. Ungar. Using radial basis functions to approximate a function and its error bounds. *IEEE Transactions on Neural Networks*. 3:624–627, 1992.
- [7] MacLachlan and Basford. *Mixture Models*. Chapman and Hall, 1988.
- [8] J. Moody and C.J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281–294, 1989.
- [9] M.T. Musavi et al. On the training of radial basis function classifiers. *Neural Networks*. 5:595–603, 1992.
- [10] K. Parthasarathy and K. Narendra. *Stable adaptive control of a class of discrete-time nonlinear systems using radial basis neural networks..* Report No. 9103, Electrical Engineering, Yale University, 1991.
- [11] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*. 247:978–982, 1990.
- [12] D.C. Psichogios and L.H. Ungar. Direct and indirect model based control using artificial neural networks. *Industrial Engineering Chemical Research*. 30:2564–2573, 1991.
- [13] R.M. Sanner and J.-J.E. Slotine. Gaussian networks for direct adaptive control. *IEEE Trans. Neural Networks*. 1992.