On Gaussian Radial Basis Function Approximations: Interpretation, Extensions, and Learning Strategies

Mário A. T. Figueiredo
Instituto de Telecomunicações
Instituto Superior Técnico
1049-001 Lisboa, PORTUGAL
E-mail: mtf@lx.it.pt

Abstract

In this paper we focus on an interpretation of Gaussian radial basis functions (GRBF) which motivates extensions and learning strategies. Specifically, we show that GRBF regression equations naturally result from representing the input-output joint probability density function by a finite mixture of Gaussians. Corollaries of this interpretation are: some special forms of GRBF representations can be traced back to the type of Gaussian mixture used; previously proposed learning methods based on input-output clustering have a new meaning; finally, estimation techniques for finite mixtures (namely the EM algorithm, and model selection criteria) can be invoked to learn GRBF regression equations.

1. Introduction

Radial basis functions (RBF) constitute a widely used and researched tool for (nonlinear) function approximation, which is a central theme in pattern analysis and recognition [1], [2], [3], [4]; see also [5] for a recent and comprehensive overview and further references.

The RBF-based (often seen as a neural network [5]) input-output relation has the form

$$y = g(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \, G(\| \mathbf{x} - \mathbf{t}_i \|) \qquad (1)$$

where $\mathbf{x} = [x_1, ..., x_d]^T$ is the input, the α_i are weights, $G(\cdot)$ is a $I\!R \rightarrow I\!R$ (usually nonlinear) function, $\| \cdot \|$ denotes some norm, and the $\mathbf{t}_i$ are called the *centers*. For interpolation purposes (either strict [1], or regularized [4]), given a set of n points $\{(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n)\}$, the centers are placed at the observed points, $\mathbf{t}_i = \mathbf{x}_i$, for $i = 1, ..., n$ (that is $k = n$). For large data-sets it may be prohibitively expensive to use $k = n$ [4]; if fewer than n centers are used, the selection of their number and location becomes the central issue in the design of RBF network learning methods.

Several choices for G have been proposed; here, we focus on Gaussian RBF (GRBF) approximations, for which G is a Gauss function $G(r) = \exp\{-r^2/2\}$. Usually, the norm in Eq. (1) is Euclidean. Generalized versions may use (possibly different) Mahalanobis norms, i.e.,

$$g(\mathbf{x}) = \sum_{i=1}^{k} \alpha_i \, G\left[\sqrt{(\mathbf{x} - \mathbf{t}_i)^T \mathbf{A}_i^{-1} (\mathbf{x} - \mathbf{t}_i)}\right]. \qquad (2)$$

Although here we consider only real-valued functions, vector-valued functions can be approximated by considering coordinate-wise GRBF approximations.

Arguably the main feature of GRBFs is their *universal approximation* property [4], according to which, given any continuous function, there is an arbitrarily close GRBF approximation. An intimately related result states that Gaussian mixtures can approximate a large class of probability density functions; in Bayesian inference, Gaussian mixture approximations have been used to represent either the prior [6], or the likelihood function [7].

Rather than considering an underlying function to approximate (which is the standard perspective in the RBF literature), let us consider the joint probability density function (p.d.f.) over the input-output space, $f_{Y,\mathbf{X}}(y, \mathbf{x})$. If this joint p.d.f. is represented by a finite Gaussian mixture, the resulting regression function (*i.e.*, $E[Y|\mathbf{x}]$) has a GRBF form. Although this seems to be common knowledge in some of the RBF literature (e.g. [8]), the main purpose of this paper is to summarize this perspective and some of its consequences. A similar interpretation based on a nonparametric kernel-based representation of $f_{Y,\mathbf{X}}(y, \mathbf{x})$ (*i.e.*, using as many centers as data points) has been considered in [9]; notice that a kernel-based representation can be seen as an extreme case of a finite mixture with as many components as data points.

Important consequences of the finite-mixture-based interpretation of GRBF regression are:

- We obtain a probabilistic justification for the normalized GRBF form (*e.g.*, [3]).
- Several known types of GRBF forms can be interpreted as resulting from different types of mixture models (*e.g.*, Euclidean versus Mahalanobis norms).

0-7695-0750-6/00 $10.00 © 2000 IEEE

- The good performance of center estimation methods based on clustering of both input and output values (*e.g.*, [10], [11], [12]) is clearly justified: clustering (namely k-means and fuzzy k-means) is closely related to finite mixture fitting.
- Selecting the dimension of the GRBF approximation may be approached with model selection methods for Gaussian mixtures (see [13] and references therein).

2. Regression, Gaussian Mixtures, and GRBFs

If $\mathbf{X}$ and Y are random variables whose joint p.d.f. is $f_{Y,\mathbf{X}}(y,\mathbf{x})$, the *regression function* (of y on $\mathbf{x}$), given by

$$\widehat{y} = g_r(\mathbf{x}) \equiv E[Y|\mathbf{x}] = \int y\, f_Y(y|\mathbf{x})\, dy,$$

is the function that minimizes the *integrated risk* (or *expected risk*) under the quadratic loss

$$\mathcal{R}[g(\cdot)] = \int f_{Y,\mathbf{X}}(y,\mathbf{x})(y - g(\mathbf{x}))^2\, dy\, d\mathbf{x}. \qquad (3)$$

For example, if there is an underlying (a.k.a. true) function $g_t(\cdot)$ such that $Y = g_t(\mathbf{X}) + N$, where N is a random variable (usually called noise in this context), then $f_{Y,\mathbf{X}}(y,\mathbf{x})$ depends on $g_t(\cdot)$, the sampling p.d.f. $f_{\mathbf{X}}(\mathbf{x})$, and the p.d.f. of N, $f_N(n)$:

$$f_{Y,\mathbf{X}}(y,\mathbf{x}) = f_N(y - g_t(\mathbf{x}))f_{\mathbf{X}}(\mathbf{x}).$$

Let us write $\mathbf{Z} = (Y, \mathbf{X})$ for the concatenation of the input and output random variables. Recalling that finite Gaussian mixtures have the universal approximation property, let us represent $f_{Y,\mathbf{X}}(y,\mathbf{x}) = f_{\mathbf{Z}}(\mathbf{z})$ by such a mixture,

$$f_{\mathbf{Z}}(\mathbf{z}) = \sum_{j=1}^{k} w_j \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_j, \mathbf{C}_j), \qquad (4)$$

where $\mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_j, \mathbf{C}_j)$ denotes a $(d+1)$-variate Gaussian p.d.f. with mean $\boldsymbol{\mu}_j$ and covariance matrix $\mathbf{C}_j$. Each covariance matrix has the following block structure

$$\mathbf{C}_j = \begin{bmatrix} C_j^{YY} & \mathbf{C}_j^{Y\mathbf{X}} \\ \mathbf{C}_j^{\mathbf{X}Y} & \mathbf{C}_j^{\mathbf{XX}} \end{bmatrix}, \qquad (5)$$

where C_j^{YY} is a scalar, $\mathbf{C}_j^{Y\mathbf{X}}$ is $(1 \times d)$, $\mathbf{C}_j^{\mathbf{X}Y} = (\mathbf{C}_j^{Y\mathbf{X}})^T$ is $(d \times 1)$, and $\mathbf{C}_j^{\mathbf{XX}}$ is $(d \times d)$. Similarly, each $((d+1) \times 1)$ mean vector can be written as

$$\boldsymbol{\mu}_j = \left[\mu_j^Y, \mu_j^{X_1}, ..., \mu_j^{X_d}\right]^T = \left[\mu_j^Y, (\boldsymbol{\mu}_j^{\mathbf{X}})^T\right]^T. \qquad (6)$$

It turns out that, from this representation, the regression function $g_r(\mathbf{x}) = E[Y|x]$ can be easily obtained in closed-form. Since $f_Y(y|\mathbf{x}) = f_{Y,\mathbf{X}}(y,\mathbf{x})/f_{\mathbf{X}}(\mathbf{x})$,

$$\begin{aligned} E[Y|x] &= \int y \frac{\sum_{j=1}^{k} w_j \mathcal{N}(y,\mathbf{x}|\boldsymbol{\mu}_j,\mathbf{C}_j)}{\int \left(\sum_{j=1}^{k} w_j \mathcal{N}(y',\mathbf{x}|\boldsymbol{\mu}_j,\mathbf{C}_j)\right) dy'}\, dy \\ &= \frac{\sum_{j=1}^{k} w_j\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j^{\mathbf{X}}, \mathbf{C}_j^{\mathbf{XX}})\, E[Y|\mathbf{x},\boldsymbol{\mu}_j,\mathbf{C}_j]}{\sum_{j=1}^{k} w_j\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j^{\mathbf{X}}, \mathbf{C}_j^{XX})}, \end{aligned}$$

which is a weighted average of the conditional means of each mixture component. We may now obtain several special cases by considering particular types of Gaussian mixture representations.

- Suppose that each component models Y and $\mathbf{X}$ as independent. Of course, this does not mean that Y and $\mathbf{X}$ are globally independent; moreover, mixtures of Gaussians under this restriction maintain their universal approximation property. In this case, $\mathbf{C}_j^{\mathbf{X}Y} = (\mathbf{C}_j^{Y\mathbf{X}})^T = 0$, and $E[Y|\mathbf{x},\boldsymbol{\mu}_j,\mathbf{C}_j] = E[Y|\boldsymbol{\mu}_j,\mathbf{C}_j] = \mu_j^Y$. The resulting regression equation is

$$g_r(\mathbf{x}) \equiv E[Y|x] = \frac{\sum_{j=1}^{k} w_j\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j^{\mathbf{X}}, \mathbf{C}_j^{XX})\, \mu_j^Y}{\sum_{j=1}^{k} w_j\, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j^{\mathbf{X}}, \mathbf{C}_j^{XX})} \qquad (7)$$

which has the Mahalanobis GRBF form (see Eq. (2)), but normalized, as in [3],

$$g_r(\mathbf{x}) = \frac{\sum_{j=1}^{k} \alpha_j\, G\left[\sqrt{(\mathbf{x}-\mathbf{t}_j)^T \mathbf{A}_j^{-1} (\mathbf{x}-\mathbf{t}_j)}\right]}{\sum_{j=1}^{k} G\left[\sqrt{(\mathbf{x}-\mathbf{t}_j)^T \mathbf{A}_j^{-1} (\mathbf{x}-\mathbf{t}_j)}\right]}, \qquad (8)$$

where $G(r) = \exp\{-\frac{r^2}{2}\}$, $\mathbf{t}_j = \boldsymbol{\mu}_j^{\mathbf{X}}$, $\mathbf{A}_j = \mathbf{C}_j^{\mathbf{XX}}$, and

$$\alpha_j = \frac{w_j\, \mu_j^Y}{\sqrt{\det(\mathbf{A}_j)}}.$$

- If the $\mathbf{C}_j^{\mathbf{XX}}$ matrices are assumed diagonal, say $\mathbf{C}_j^{\mathbf{XX}} = \sigma_j^2 \mathbf{I}$, Eq. (8) simplifies to

$$g_r(\mathbf{x}) = \frac{\sum_{j=1}^{k} \alpha_j\, G\left[\frac{\|\mathbf{x}-\mathbf{t}_j\|}{\sigma_j}\right]}{\sum_{j=1}^{k} G\left[\frac{\|\mathbf{x}-\mathbf{t}_j\|}{\sigma_j}\right]}, \qquad (9)$$

which is a more standard (still normalized) GRBF representation. Of course, we can further impose $\sigma_i^2 = \sigma^2$ and obtain an even simpler GRFB regression equation.

- Finally, let us address the unrestricted case where the $\mathbf{C}_j$ matrices are arbitrary. Since each component models Y and $\mathbf{X}$ as jointly Gaussian, it is well known that

$$E[Y|\mathbf{x}, \boldsymbol{\mu}_j, \mathbf{C}_j] = \mu_j^Y + \mathbf{C}_j^{Y\mathbf{X}}(\mathbf{C}_j^{\mathbf{XX}})^{-1}(\mathbf{x} - \boldsymbol{\mu}_j^{\mathbf{X}}).$$

This means that the regression equation becomes

$$g_r(\mathbf{x}) = \frac{\sum_{j=1}^{k} \alpha_j(\mathbf{x})\, G\left[\sqrt{(\mathbf{x}-\mathbf{t}_i)^T \mathbf{A}_i^{-1}(\mathbf{x}-\mathbf{t}_i)}\right]}{\sum_{j=1}^{k} G\left[\sqrt{(\mathbf{x}-\mathbf{t}_i)^T \mathbf{A}_i^{-1}(\mathbf{x}-\mathbf{t}_i)}\right]} \tag{10}$$

where

$$\alpha_j(\mathbf{x}) = \frac{w_j\, E[Y|\mathbf{x}, \boldsymbol{\mu}_j, \mathbf{C}_j]}{\sqrt{\det(\mathbf{A}_j)}};$$

this can be seen as a extended GRBF representation with input-dependent weights (rather than the usual fixed weights). Since GRBF forms with fixed weights are a special case of this representation, nothing is lost in terms of universal approximation properties.

3. Learning Strategies

In practice, the regression function has to be learnt from a set of n samples $\{(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n)\}$; the joint density is, of course, unknown. The equivalence between GRBF regression and the joint Gaussian mixture suggests that we learn this mixture from the data; the parameters of the GRBF regression equation are simple functions of the parameters of the joint mixture. The standard approach to obtaining *maximum likelihood* (ML) estimates of the parameters of a finite mixture is the well-known *expectation-maximization* (EM) algorithm [14]. If incremental, rather than batch, learning is desired, on-line versions of EM may be used [15]. Of course is may be argued that by learning the joint density we are solving a more general problem than regression [16]; this is of course true, and specially relevant for small samples. The main advantage of EM is its simplicity, and we only advocate its use for large samples.

The mixture-based interpretation also explains the success of (and gives a formal justification to) learning methods that find the centers by performing clustering using both input and output values [10], [11], [12]. In fact, many clustering algorithms (namely k-means, and fuzzy k-means), can be seen as approximate versions of the EM algorithm for Gaussian mixtures. However, the EM algorithm has an important advantage; by using EM to learn the GRBF parameters, we not only learn the center locations (as input-output clustering methods do) but also the width (or widths, or Mahalanobis distance matrices) and weights of each component. That is, we learn the whole set of parameters of the GRBF representation with a single simple algorithm.

A central issue in GRBF regression is the selection of an appropriate dimension (number of centers) for the representation. The standard tradeoff of model order selection problems arises: with too many centers, the learned representation may over-fit the data and have poor generalization properties; a representation with too few components may not be rich enough to approximate the underlying relation. Under the interpretation here studied, selecting the dimension of the GRBF representation becomes equivalent to selecting the number of components of the joint mixture. Several model selection criteria for finite mixtures have been proposed; see recent work in [13], [17], and references therein. For lack of space, we do not focus on this issue here; in the examples presented ahead we use the method that we have proposed in [13].

Finally, a relevant feature of the EM approach for learning the GRBF parameters concerns unlabeled samples [8]. Suppose that, in addition to the data $\{(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n)\}$, we have a set of m unlabeled samples $\{\mathbf{x}_{n+1}, ..., \mathbf{x}_{n+m}\}$, *i.e.*, without the corresponding responses; then, these responses can be treated as *missing data*, and we can use an adequate version of the EM algorithm. We can also use a set of outputs (without the corresponding inputs) $\{y_{n+1}, ..., y_{n+l}\}$, by treating the corresponding inputs as missing data.

4. Illustrative Examples

We illustrate the ideas here presented with a couple of simple examples. A more thorough experimental evaluation, for which there is no space here, is clearly needed.

Fig. 1 shows the function used in the example, $f(x) = \sin(6.5(x - 0.5)) + 4\exp(-6(x - 0.5)^3)$, and a set of 200 samples generated as described in the caption of that figure. In Fig. 2, the Gaussian mixture fitted to the samples (under the constraint of diagonal covariances) is shown, together with the corresponding regression function (Eq. (8) or (9), which are equivalent because, in this case, $X \in \mathbb{R}$). Finally, Fig. 3 displays the Gaussian mixture fitted to the same data, now with unconstrained covariances, and the corresponding regression function (Eq. (10)). Notice that this extended GRBF regression (Eq. (10)) can not be learned by standard methods due to the input-dependent weights. The extended regression achieved a smaller MSE using fewer components due to its more flexible nature.

5. Conclusions

By representing the joint input-output probability density functions by Gaussian mixtures, the resulting regression equations have GRBF form. By exploiting this fact, we have presented an extended version of GRBF approximations where the weights are input-dependent. This equivalence also suggests that we may learn the GRBF representation by using the EM algorithm to fit the input-output joint mixture; this possibility was illustrated experimentally.

References

[1] M. Powell, "Radial basis functions for multivariate interpolation," in *Algorithms for Approximation* (J. Ma-

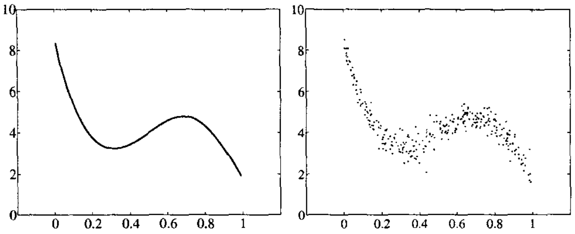

Figure 1. Left: the test function $f(x) = \sin(6.5(x - 0.5)) + 4\exp(-6(x - 0.5)^3)$. Right: 200 (x, y) random samples such that the x's are uniformly distributed over $[0, 1]$, and each y is given by $y = f(x) + n$, where n is zero-mean Gaussian with variance 1.25.

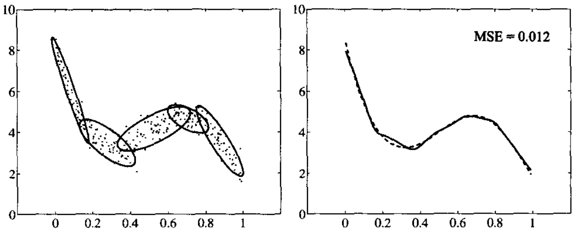

Figure 3. Left: a Gaussian mixture, with arbitrary covariance matrices, fitted to the data from Fig. 1. Right: the corresponding regression function (solid line) plotted together with the original function (dashed line).

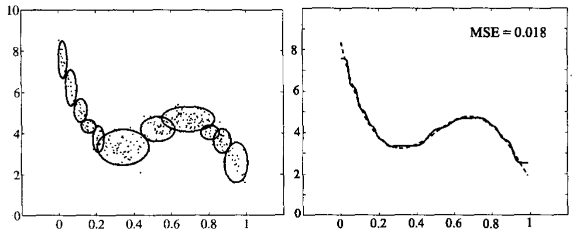

Figure 2. Left: a Gaussian mixture, with diagonal covariance matrices, fitted to the data from Fig. 1. Right: the corresponding regression function (solid line) together with the original function (dashed line).

son and M. Cox, eds.), (Oxford), pp. 143–167, Clarendon Press, 1987.

[2] D. Broomhead and D. Lowe, "Multivariate function approximation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.

[3] J. Moody and C. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, pp. 281–294, 1989.

[4] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. of the IEEE*, vol. 78, pp. 1481–1497, 1990.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, N.J.: Prentice Hall, 1999. 2nd Edition.

[6] S. Dalal and W. Hall, "Approximating priors by mixtures of natural conjugate priors," *Journal of the Royal Statistical Society (B)*, vol. 45, 1983.

[7] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society (B)*, vol. 58, pp. 155–176, 1996.

[8] D. Miller and H. Uyar, "Combined learning and use for a mixture model equivalent to the RBF classifier," *Neural Computation*, vol. 10, pp. 281–293, 1998.

[9] A. Krzyżak, T. Linder, and G. Lugosi, "Nonparametric estimation and classification using radial basis functions," *IEEE Trans. on Neural Networks*, vol. 7, pp. 475–487, 1996.

[10] C. Chen, W. Chen, and F. Chang, "Hybrid learning algorithm for Gaussian potential function networks," *IEE Proc.-D*, vol. 140, pp. 442–448, 1993.

[11] Y. Zhang, X. Rong Li, Z. Zhu, and H. Zhang, "A new clustering and training method for radial basis function networks," in *Proc. of the Intern. Conf. on Neural Networks - ICNN*, pp. 311–316, 1996.

[12] W. Pedrycz, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. on Neural Networks*, vol. 9, pp. 601–612, 1998.

[13] M. Figueiredo and A. K. Jain, "Unsupervised selection and estimation of finite mixture models," in *Proceedings of the International Conference on Pattern Recognition – ICPR-2000*, (Barcelona), 2000.

[14] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. New York: John Wiley & Sons, 1997.

[15] Y. Singer and M. Warmuth, "A new parameter estimation method for Gaussian mixtures," in *Advances in Neural Inform. Proc. Systems 11* (M. Kearns, S. Solla, and D. Cohn, eds.), MIT Press, 1999.

[16] V. Vapnik, *Statistical Learning Theory*. New York: John Wiley, 1998.

[17] M. Figueiredo, J. Leitão, and A. K. Jain, "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition* (E. Hancock and M. Pellilo, eds.), pp. 54–69, Springer Verlag, 1999.