Automatic Basis Selection Techniques for RBF Networks

Ali Ghodsi
School of Computer Science
University of Waterloo
200 University Avenue West
N2L 3G1, Canada
Email: aghodsib@cs.uwaterloo.ca

Dale Schuurmans
School of Computer Science
University of Waterloo
200 University Avenue West
N2L 3G1,Canada
Email: dale@cs.uwaterloo.ca

***Abstract*—This paper proposes a generic criterion that defines the optimum number of basis functions for radial basis function neural networks. The generalization performance of an RBF network relates to its prediction capability on independent test data. This performance gives a measure of the quality of the chosen model. An RBF network with an overly restricted basis gives poor predictions on new data, since the model has too little flexibility (yielding high bias and low variance). By contrast, an RBF network with too many basis functions also gives poor generalization performance since it is too flexible and fits too much of the noise on the training data (yielding low bias but high variance). Bias and variance are complementary quantities, and it is necessary to assign the number of basis function optimally in order to achieve the best compromise between them. In this paper we use Stein's unbiased risk estimator (SURE) to derive an analytical criterion for assigning the appropriate number of basis functions. Two cases of known and unknown noise have been considered and the efficacy of this criterion in both situations is illustrated experimentally. The paper also shows an empirical comparison between this method and two well known classical methods, cross validation and the Bayesian information criterion, BIC.**

I. Introduction

Radial basis function (RBF) networks have attracted a lot of interest in the past. One reason is that they form a unifying link between function approximation, regularization, noisy interpolation, classification and density estimation. It is also the case that training radial basis function networks is usually faster than training multi-layer perceptron networks.

RBF network training usually proceeds in two steps: First, the basis function parameters (corresponding to hidden units) are determined by clustering. Second, the final-layer weights are determined by least squares which reduces to solving a simple linear system. Thus, the first stage is an unsupervised method which is relatively fast, and the second stage requires the solution of a linear problem, which is also fast.

One of the advantages of radial basis function neural networks, compared to multi-layer perceptron networks, is the possibility of choosing suitable parameters for the units of hidden layer without having to perform a non-linear optimization of the network parameters. However, the problem of selecting the appropriate number of basis functions remains a critical issue for RBF networks. The number of basis functions controls the complexity, and hence the generalization ability of RBF networks. An RBF network with too few basis functions gives poor predictions on new data, i.e. poor generalization, since the model has limited flexibility. On the other hand, an RBF network with too many basis functions also yields poor generalization since it is too flexible and fits the noise in the training data. A small number of basis functions yields a high bias, low variance estimator, whereas a large number of basis functions yields a low bias but high variance estimator. The best generalization performance is obtained via a compromise between the conflicting requirements of reducing bias while simultaneously reducing variance. This tradeoff highlights the importance of optimizing the complexity of the model in order to achieve the best generalization.

In this paper, we propose a criterion for selecting the number of radial basis functions in an RBF network, and compare it with two well studied classical model selection method, cross validation and the Baysian information criterion (BIC).

To develop a theoretically well motivated criterion for choosing an appropriate number of basis functions , we derive a generalization of Stein's unbiased risk estimator (SURE) (Ker-Chau, 1985) which defines a generic criterion that determines the optimum number of basis functions to use in a given problem.

In Section II of this paper we review RBF networks and their training algorithm. We then explain the under-fitting and over-fitting effects caused by using an inappropriate number of basis functions for RBF networks in Section III. In Section IV we derive a generalization of SURE, and in Section V show how it can be applied to RBF networks. Experimental results of the proposed criterion and its compression with other methods are presented in Section VI.

II. Radial Basis Function Networks

Radial basis function networks are a major class of neural network model, where the distance between the input vector and a prototype vector determines the activation of a hidden unit. Radial basis function methods became a popular technique in the mid 1980s for performing exact interpolation of a set of data points in a high-dimensional space (Powell, 1987). The basic technique provides an interpolating function which passes through every data point: Consider a mapping from a d-dimensional input space to a one-dimensional target space y, where the data set consists of N input vectors x_i, with corresponding targets y_i, $i = 1...N$. An exact interpolation is achieved by introducing a set of N basis functions, one for each data point, and then setting the weights for the linear combination of basis functions. Here, basis functions are non-linear functions, $\Phi(||x - x_i||)$, of the input vectors x_i, and a linear combination of these basis functions can be written as

$$\sum_{i=1}^{N} w_i \Phi(||x - x_i||)$$

The interpolation problem can then be written in a matrix form as

$$\Phi W = y$$

where the square matrix Φ has elements $\Phi_{ii'} = \Phi(||x_i - x_{i'}||)$. This linear system of equations can be solved to yield

$$W = \Phi^{-1} y$$

For the case of Gaussian basis functions we have

$$\Phi(x) = \exp\left(-\frac{x^2}{2v^2}\right)$$

where v is a parameter that controls the smoothness of the interpolating function.

A radial basis function neural network model (Broomhead and Lowe, 1988), (Moody and Darken, 1989) can be obtained by a number of modifications to the exact interpolation procedure as follows: First, the number, M, of basis functions is usually much less than the number, N, of data points. Second, the centers of the basis functions no longer need to be given by input data vectors, and appropriate centers can alternatively be determined during the training process. Third, unlike the exact interpolation procedure, each basis function can have its own width parameter, v_j, whose value is also determined in the training process. Finally, by applying these changes to the original (exact) interpolation formula we obtain the following form for the radial basis function neural network mapping.

$$y_k(X) \quad = \quad \sum_{j=1}^{M} w_{kj}\Phi_j(X) + w_{k0} \qquad (1)$$

By including an extra basis function Φ_0 whose activation is set to 1, the biases w_{k0} can be absorbed into the final summation. Another useful variation is the normalized RBF representation:

$$y_k(X) = \frac{\sum_{j=1}^{M} w_{kj}\Phi_j(X)}{\sum_{r=1}^{M} \Phi_r(X)}$$

This normalized representation is closely related to TSK fuzzy inference systems (Sugeno and Yasakawa, 1993; Takagi and Sugeno, 1985).

Several forms of basis function have been considered in previous research on RBF models, the most common being the Gaussian:

$$\Phi_j(X) = \exp\left(-\frac{||X - \mu||^2}{2v_j^2}\right)$$

where X is the d-dimensional input vector with elements x_i, and μ_j is the center of basis function Φ_j.

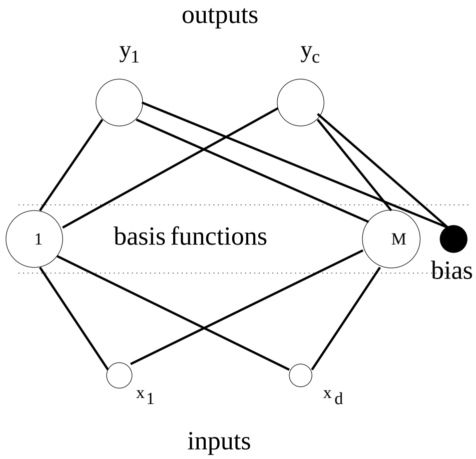

Fig. 1. A graphical form of a radial bassis function neural network architecture. Basis functions act like hidden units. The corresponding elements μ_{ji} of the vector μ_j is represented by the line connecting basis function Φ_j to the inputs. The weights w_{kj} are shown as lines from the basis functions to the output units. An extra basis function whose output is fixed at 1 serves as the bias for each output unit.

In practice, training RBF networks proceeds through two steps. The first step determines the first layer of weights, in which the basis function parameters μ_j and σ_j are selected based on the X-values of the training samples (via unsupervised learning techniques). The basis functions are then kept fixed while the second-layer weights w_i, are estimated via linear least squares.

Typical approaches for the first phase include using the generalized Lloyd algorithm (GLA) and Konhonen's self-organized maps (SOM). Another common approach is to model the input distribution as a Gaussian mixture model and then estimate the center and width parameters of the Gaussian mixture components via the EM algorithm (Bishop, 1995). For the second phase one can consider the radial basis function network mapping in (1). If we absorb the bias parameters into the weights, this can be written in matrix notation as

$$Y \quad = \quad W\Phi \qquad (2)$$

where Y is a matrix of output values, and $W = (w_{kj})$ is a matrix of second-layer weights to be estimated.

This is a classical least squares estimation problem. A necessary condition for $||Y - W\Phi||^2$ to be minimized is that W must satisfy

$$W = (\Phi^T \Phi)^{-1} \Phi^T Y$$

III. Over-fitting, under-fitting and model selection

Model selection is the task of choosing a model of optimal complexity for a given problem. Learning a radial basis function network from data is a parameter estimation problem. One difficulty with this problem is selecting parameters that show good performance on both training and testing data. In principle, a model is selected to have parameters associated with the best observed performance on training data, although our goal really is to achieve good performance on unseen testing data. Not surprisingly, a model selected on the basis of training data does not necessarily exhibit comparable performance on the testing data. When squared error is used as the performance index, a zero-error model on the training data can always be achieved by using a sufficient number of basis functions. However, training error, err, and testing error, Err, do not demonstrate a linear relationship. In particular, a smaller training error does do not necessarily result in a smaller testing error. In practice, one often observes that, up to a certain point, the model error on testing data tends to decrease as the training error decreases. However, if one attempts to decrease the training error too far by increasing model complexity, the testing error often can take a dramatic increase.

The basic reason behind this phenomenon is that in the process of minimizing training error, after a certain point, the model begins to over-fit the training set. Over-fitting in this context means fitting the model to training data at the expense of losing generality. In the extreme form, as we mentioned in the previous section, a set of N training data points can be modeled exactly with N radial basis functions. Such a model follows the training data perfectly. However, the model is not representative features of the true underlying data source, and this is why it fails to correctly model new data points.

In general, the training error rate err will be less than the testing error on the new data, Err. A model typically adapts to the training data, and hence the training error err will be an overly optimistic estimate of the generalization error Err. An obvious way to estimate generalization error is to estimate the degree of optimism OP inherent in a particular estimate, and then add a penalty term to the training error to compensate, i.e., such that $Err = err + OP$. The method described in the next section works in this way.

IV. Estimating the optimism

A. Stein's unbiased risk estimator

Let:

- $\widehat{f}(X)$ denote the *prediction model*, which is estimated from a training sample by the RBF neural network model.
- $f(X)$ denote the *real model*.
- $MSE(\widehat{f}) = E(\widehat{f} - f)^2$.
- err denote the *training error*, which is the average loss over the training sample.
- Err denote the *generalization error*, which is the expected prediction error on an independent test sample.

Recall that the training error, $err = \sum_{i=1}^{N}(\widehat{y} - y)^2$, is an estimate of the expectation of the squared error on the training data, $E(\widehat{y} - y)^2$, while the generalization error (test error) Err is an estimate of mean squared error, $MSE = (\widehat{f} - f)^2$, where $\widehat{f}(X)$ is the estimated model and $f(X)$ is the true model.

Now suppose $y_i = f(x_i) + \varepsilon_i$, where ε is additive Gaussian noise $N(0, \sigma^2)$. We need to estimate $\widehat{f}$ from training data $D = \{(x_i, y_i)\}_i^n$. Consider

$$\begin{aligned} E[(\widehat{y_0} - y_0)^2] &= E[(\widehat{f} - f - \varepsilon)^2] \\ &= E[(\widehat{f} - f)^2] + E[\varepsilon^2] - 2E[\varepsilon(\widehat{f} - f)] \qquad (3) \\ &= E[(\widehat{f} - f)^2] + \sigma^2 - 2E[\varepsilon(\widehat{f} - f)] \qquad (4) \end{aligned}$$

Here, the last term can be written as:

$$E[2\varepsilon(\widehat{f} - f)] = 2E[(y_0 - f)(\widehat{f} - f)] \equiv cov(y_0, \widehat{f})$$

We consider two different cases.

a) Case 1: Consider the case in which a new data point has been introduced to the estimated model, $i.e. (x_0, y_0) \notin D$. Since y_0 is a new point, $\widehat{f}$ and y_0 are independent. Therefore $cov(y_0, \widehat{f}) = 0$ and (4) in this case can be written as:

$$E[(\widehat{f} - f)^2] = \sigma^2 - E(\widehat{y_0} - y_0)^2 \qquad (5)$$

This is the justification behind the technique of cross validation. In cross validation, to avoid overfitting or underfitting, a validation data set is used which is independent from the estimated model. The optimal model parameters should be selected to have the best performance index associated with this data set. Since this data set is independent from the estimated model, it is a fair estimate of $E(\widehat{f} - f)^2$ and consequently of generalization error Err as indicated in (5).

b) Case 2: A more interesting case is the case in which we do not use new data points to assess the performance of the estimated model, and the traing data is used for both estimating and assessing a model $\widehat{f}$. In this case the cross term in (4) cannot be ignored because $\widehat{f}$ and y_0 are not independent. Therefore the cross term, which is $cov(y_0, \widehat{f})$, is not zero. However the cross term can be estimated by Stein's lemma (Ker-Chau, 1985),(Stein, 1981). which was originally proposed to estimate the mean of a Guassian distribution (Stein, 1981).

According to Stein's lemma if $X \sim N(\theta, \sigma^2)$ and $g(x)$ is a differentiable function, such that $E[|g'(x)|] < \infty$ then $E(g(x)(x - \theta)) = \sigma^2 E(g'(x))$. So we let

$$g(\varepsilon) = \widehat{f} - f = \widehat{f} - y - \varepsilon$$

and $x = \varepsilon$. Then by applying Stein's lemma we obtain

$$E(\varepsilon(\widehat{f} - f)) = \sigma^2 E(g'(\varepsilon)) = \sigma^2 E\left(\frac{d\widehat{f}}{dy}\right)$$

Summing over all y we get

$$\begin{aligned} Err &= \sum_{i=1}^{N}(\widehat{y} - y)^2 - N\sigma^2 + 2\sigma^2 \sum_{i=1}^{N} \frac{d\widehat{f}(x_i)}{dy_i} \\ &= err - N\sigma^2 + 2\sigma^2 \sum_{i=1}^{N} \frac{d\widehat{f}(x_i)}{dy_i} \qquad (6) \end{aligned}$$

This is known as Stein's Unbiased Risk Estimator (SURE).

B. The Bayesian information criterion (BIC)

A well studied method for estimating the degree of optimism *OP* is the BIC statistic, which is also known as the Schwarz criterion (Schwarz, 1978). BIC is a classical model selection criterion motivated in quite a different way from SURE. However it yields a very similar result. BIC has the generic form of:

$$BIC = -2\ loglik + (\log N)\ p$$

where in the case of linear models p is the dimensionality of inputs.

Under the Gaussian model, assuming the variance σ^2 is known, Schwarz's procedure can be written as:

$$BIC = (N/2\sigma^2)\left[err + \frac{p}{N}(\log N)\sigma^2\right] \qquad (7)$$

V. Optimum number of basis functions for RBF networks

Based on SURE, the optimum number of basis functions should be assigned to have the minimum generalization error Err in (6). From the least squared solution of (2) we have:

$$\begin{aligned} W &= (\Phi^T\Phi)^{-1}\Phi^T Y \\ \widehat{f} &= \Phi W = \Phi(\Phi^T\Phi)^{-1}\Phi^T Y = HY \end{aligned} \qquad (8)$$

where H depends on the input vector x_i but not on y_i. Note that in practice, the equation (2) is solved using singular value decomposition to avoid problems due to possible ill-conditioning of the matrix Φ.

From (8) we can easily obtain the required derivative of $\widehat{f}(x_i)$ with respect to y_i.

$$\sum_{i=1}^{N} \frac{d\widehat{f}(x_i)}{dy_i} = \sum_{i=1}^{N} H_{ii}$$

Now, substituting this into (6) we obtain

$$Err \quad = \quad err - N\sigma^2 + 2\sigma^2 \sum_{i=1}^{N} H_{ii}$$

Here we observe that $\sum_{i=1}^{N} H_{ii} = Trace(H)$, the sum of the diagonal elements of H. Thus, we can obtain the further simplification that $Trace(H) = Trace(\Phi(\Phi^T\Phi)^{-1}\Phi^T) = Trace(\Phi^T\Phi(\Phi^T\Phi)^{-1}) = Trace(I) = p$, where p is the dimension of Φ. Since Φ is a projection of input matrix X onto a basis set spanned by M, the number of basis functions, one can show generally $p = M + 1$.

To use this method to find the optimum number of basis functions, we simply choose the model that obtains the smallest Err over the set of models considered. Given a set of models $\widehat{f_M}(x)$ indexed by the number of basis functions, M, denote the training error for each model by $err(M)$. We then obtain

$$Err(M) \quad = \quad err(M) - N\sigma^2 + 2\sigma^2(M+1) \qquad (9)$$

where N is the number of training samples and the noise, σ^2, can be estimated from the mean squared error of the model.

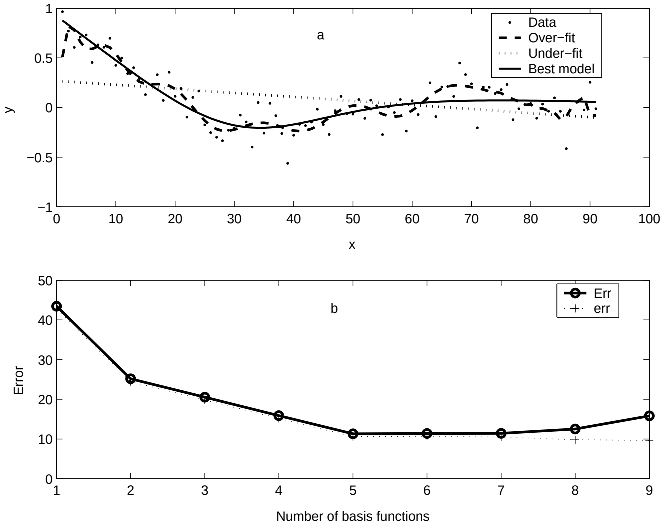

Fig. 2. (a) Overfitting, underfitting, and the best estimated model for $y = \frac{sin(x)}{x}$. (b) Err obtained in (9) used to find the optimum number of basis functions for model $y = \frac{sin(x)}{x}$.

Based on the discussion in this section applying BIC to RBF networks is trivial. One just needs to substitute p with $M + 1$ in (7) to obtain:

$$BIC(M) = (N/2\sigma^2)[err(M) + \frac{M+1}{N}(\log N)\sigma^2] \qquad (10)$$

A. Estimating unknown noise

The proposed criterion in (9) depends on the standard deviation of additive noise σ. In practice, however, the noise and therefore its standard deviation are often unknown. When this is the case, noise can be estimated from the training data as

$$\widehat{\sigma^2} \quad = \quad \frac{1}{N-p}\sum_{i}^{n}(\widehat{y} - y) \qquad (11)$$

In this case, one can use (11) in conjunction with (9) in one of two possible ways. In the first approach, noise is estimated via (11) using a high-variance/low-bias estimator, and then this estimate is plugged into (9) to select optimal model complexity. Under the second approach, noise is estimated via (11) for each model complexity. Therefore different estimation are used in (9) for each model complexity. In this paper we use the earlier approach to conduct empirical compression.

VI. Experimental results

To explore the effectiveness of our complexity control method, we considered the problem of fitting a RBF network model to a set of points (Figure 2). The goal is to minimize the squared generalization error Err. To determine the efficacy of the method we compared its performance to the well studied standard cross validation and BIC (Craven and Wahba, 1979).

We first conducted a simple series of experiments by fixing a uniform distribution on the unit interval [0,1], and then fixing various target functions $f : [0, 1] \rightarrow R$. To generate

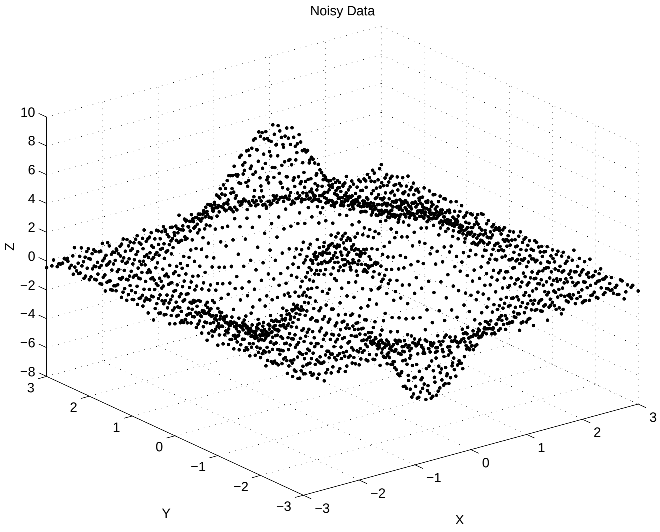

Fig. 3. Data generated in $3D$ dimensional space by nonlinear function 'Peaks', and distorted by additive Gaussian noise $N(0, 0.25)$

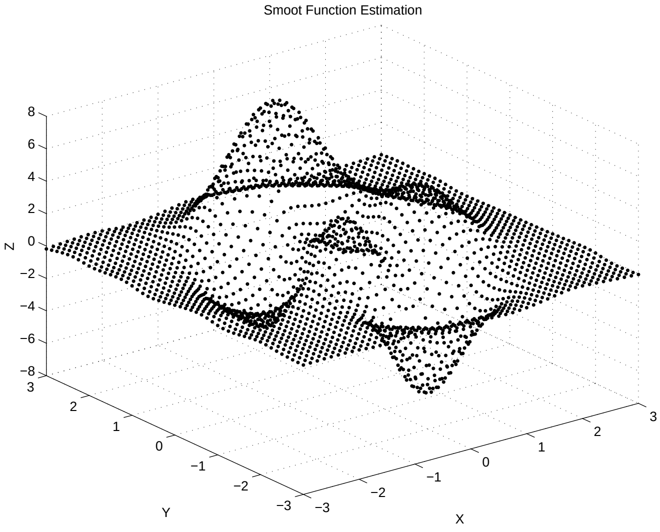

Fig. 4. The smooth true function 'Peaks' has been recovered from noisy observation. Err obtained in (9) used to find the optimum number of radial basis functions for the model.

training samples, a sequence of values $x_1, ..., x_t$ is drawn from $[0, 1]$, the target function values $f(x_1), ..., f(x_t)$ are computed, and independent Gaussian noise is added to each value. In the first step of RBF network training, centers μ_j and width parameter v_j are estimated using subtractive clustering (Chiu S. L., 1994), an unsupervised training technique.

For a given training sample, the series of best fit functions corresponding to a number of basis functions $M = 1, 2, ...,$ *etc.* are computed. Given this sequence, the cross validation strategy will choose some particular model $\widehat{f_M^*}$ on the basis of the observed empirical errors on the validation data set (generated the same way as training data). Our technique will alternatively chose the model corresponding to minimum Err in (9) and BIC will select the model corresponding to min BIC in (10).

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	1.021	1.082	1.041
$\widehat{\sigma}\_SURE$	1.022	1.089	1.056
BIC	1.020	1.082	1.038
CV	1.074	1.083	1.029

TABLE I

FITTING DIFFERENT TARGET FUNCTIONS WITH $\sigma = 0.25$. TABLE REPORTS RATIO OF TEST ERRORS RELATIVE TO BEST POSSIBLE TEST ERROR ACHIEVED BY DIFFERENT METHODS. A SMALLER RATIO IS BETTER. RESULTS ARE REPORTED AT TRAINING SAMPLE SIZES $N = 10$ AND AVERAGED OVER 100 REPEATED TRIALS IN EACH CASE. THE FIRST ROW CORRESPONDING TO SURE WHEN σ IS KNOWN AND THE SECOND ROW SHOWS SURE WHEN σ IS ESTIMATED. BIC IS EVALUATED WHEN σ IS KNOWN.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	1.0044	1.0068	1.0166
$\widehat{\sigma}\_SURE$	1.0049	1.0063	1.0169
BIC	1.0030	1.0063	1.0169
CV	1.0044	1.005	1.0181

TABLE II

SAME AS TABLE I BUT WITH TRAINING SAMPLE SIZE $N = 100$.

To determine the effectiveness of these strategies, the ratio of the test error of the model selected by them to the best test error on a new test data set among the models in sequence $M = 1, 2, ...$ is measured.

Table I through IV show the results obtained for fitting various functions. These results are obtained by repeatedly generating training samples of a fixed size, and recording the ratio of test error achieved relative to the best possible test error, for each technique (BIC, CV and SURE).

In another experiment we considered a highly nonlinear function distorted by Gaussian noise.

$$z = 3(1-x)^2 e^{(-(x^2)-(y+1)^2)} - 10\left(\frac{x}{5} - x^3 - y^5\right) e^{(-x^2-y^2)} - \frac{1}{3} e^{(-(x+1)^2-y^2)} + \varepsilon$$

$$\varepsilon \sim N(0, 0.25)$$

We call this function Peaks. Here the goal is to estimate

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	1.0254	1.0686	1.0701
$\widehat{\sigma}\_SURE$	1.0292	1.0891	1.0672
BIC	1.0252	1.0681	1.0649
CV	1.0544	1.0721	1.0635

TABLE III

SAME AS TABLE I , WITH TRAINING SAMPLE SIZE $N = 10$ AND $\sigma = 0.5$.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	1.0056	1.0052	1.0120
$\widehat{\sigma}\_SURE$	1.0068	1.0045	1.0142
BIC	1.0031	1.0037	1.0112
CV	1.0050	1.0051	1.0132

TABLE IV

SAME AS TABLE I, WITH TRAINING SAMPLE SIZE $N = 100$ AND $\sigma = 0.5$.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	-0.3	0.57	-0.42
$\widehat{\sigma}\_SURE$	-0.1	0.24	-0.15
BIC	-0.33	0.54	-0.48
CV	-0.18	-0.22	-0.62

TABLE V

FITTING DIFFERENT TARGET FUNCTIONS WITH $\sigma = 0.25$. TABLE REPORTS THE DIFFERENCE BETWEEN THE OPTIMUM NUMBER OF RBF FUNCTIONS AND THE NUMBER OF RBF FUNCTIONS CHOSEN BY DIFFERENT TECHNIQUES. RESULTS ARE REPORTED AT TRAINING SAMPLE SIZES $N = 10$ AND AVERAGED OVER 100 REPEATED TRIALS IN EACH CASE.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	0.14	0.37	0.41
$\widehat{\sigma}\_SURE$	0.41	0.07	0.57
BIC	-0.11	0.1	0.19
CV	0.51	0.15	0.57

TABLE VI

SAME AS TABLE V, WITH TRAINING SAMPLE SIZE $N = 100$ AND $\sigma = 0.25$.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	-0.33	0.09	0.01
$\widehat{\sigma}\_SURE$	-0.11	-0.4	0.18
BIC	-0.34	-0.01	-0.07
CV	-0.19	-0.32	0.19

TABLE VII

SAME AS TABLE V, WITH TRAINING SAMPLE SIZE $N = 10$ AND $\sigma = 0.5$.

	Target function		
Method	step($x \geq 0.5$)	$sin(\frac{1}{x})$	$sin^2(2\pi x)$
σ_SURE	0.01	0.07	-0.19
$\widehat{\sigma}\_SURE$	0.22	-0.05	0.39
BIC	-0.19	-0.15	-0.53
CV	0.38	0.3	0.26

TABLE VIII

SAME AS TABLE V, WITH TRAINING SAMPLE SIZE $N = 100$ AND $\sigma = 0.5$.

the true smooth function based on noisy observations (Figure 3). In this experiment, our proposed criterion SURE and cross validation CV chose 35 basis functions , and therefore they achieved the same generalization error on the test data. However, the SURE technique does not require an extra validation set to choose the number of basis functions, and in fact used half the data available to CV in this case. Therefore, we claim that it achieved more effective performance on this example.

The degree of optimism *OP* in BIC is similar to SURE. They produce the same result if the factor 2 in SURE is replaced by log N. This means that when $N > e^2$, BIC penalizes complex models more and gives preference to simpler models that SURE.

VII. CONCLUSION

We have proposed a new approach to choosing the optimum number of basis functions for RBF networks. Our approach minimizes a theoretically unbiased estimate of generalization error of the model. Our experimental results validate the effectiveness of this approach. A comparison cross validation illustrates that the generalization error of the models selected by our approach can be less than models selected by cross validation. Importantly, this is achieved while requiring much less computation than cross validation. The utility of our method is greatest when there is insufficient data to hold out a validation set for cross validation.

Our method based on SURE and BIC behave similarly in our experiments. However one can show analytically that BIC generally gives preference to simpler models because it penalizes complex models more harshly.

REFERENCES

Bishop, C. M.(1995). Neural Networks for Pattern Recognition. Oxford: Oxford University Press.

Broomhead, D. S. and Lowe D.(1988). Multivariable functional interpolation and adaptive networks. Complex Systems. **2** , 321-355.

Chiu S. L.(1994) . Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems. **2**, 267-278.

Craven P. and Wahba G. (1979). Smoothing noisy data with spline functions. Numer. Math. **31**, 377-403.

Ker-Chau L.(1985). From Stein's Unbiased Risk Estimates to the Method of Generalized Cross Validation. Annals of Statistics. **13(4)** , 1352-1377.

Moody, J. and Darken C. J. (1989). Fast learning in networks of locally-tuned processing units. Neural Computation. **1** (2), 281-294.

Powell, M. J. D. (1987). Radial basis functions for multivariable interpolation: a review. In J.C. Mason and M.G. Cox (Eds.), Algorithms for Approximation(pp.143-167) Oxford: Clarendon Press.

Schwarz, G.(1978) Estimating the dimension of a model. Annals of Statist. **6** , 461-464.

Stein M. C. (1981). Estimation of the Mean of a Multivariate Normal Distribution. Annals of Statistics. **9(6)** , 1135-1151.

Sugeno M. & Yasukawa T.(1993). A fuzzy-logic-based approach to qualitative modeling. IEEE Transaction on Fuzzy System. **l** , 7-31.

Takagi T. & Sugeno M.(1985). Fuzzy identification of systems and its application to modeling and control. IEEE Transaction on System Man & Cybernetic. **SMC-15(3)**, 116-132.