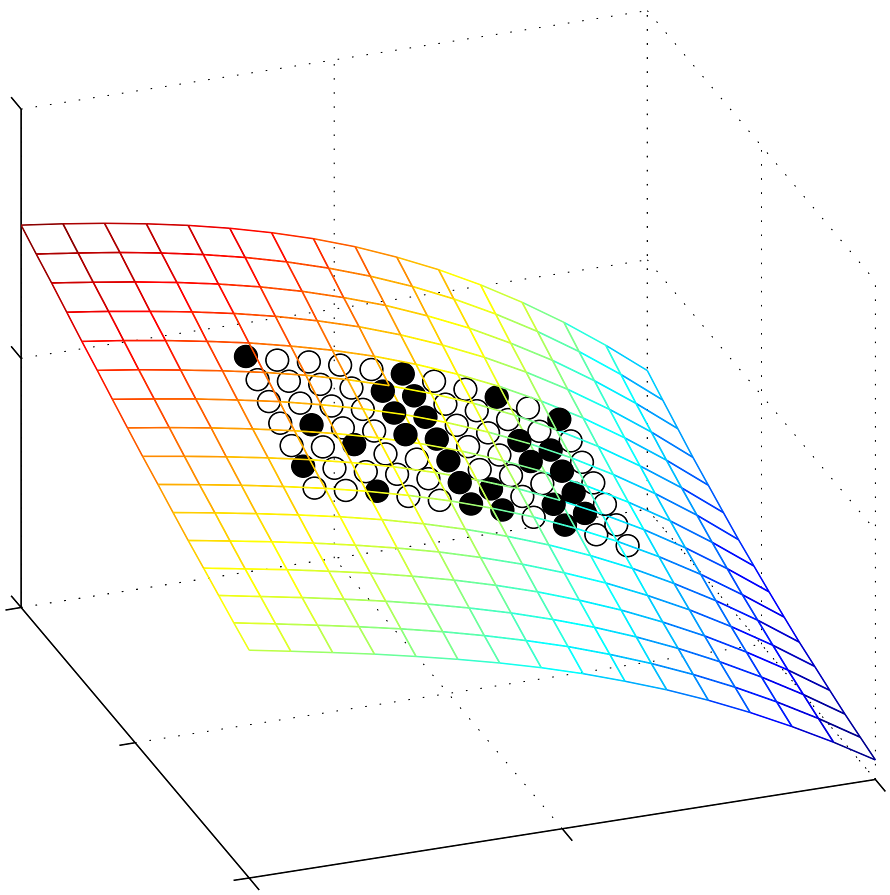
The *refmix* Data Set

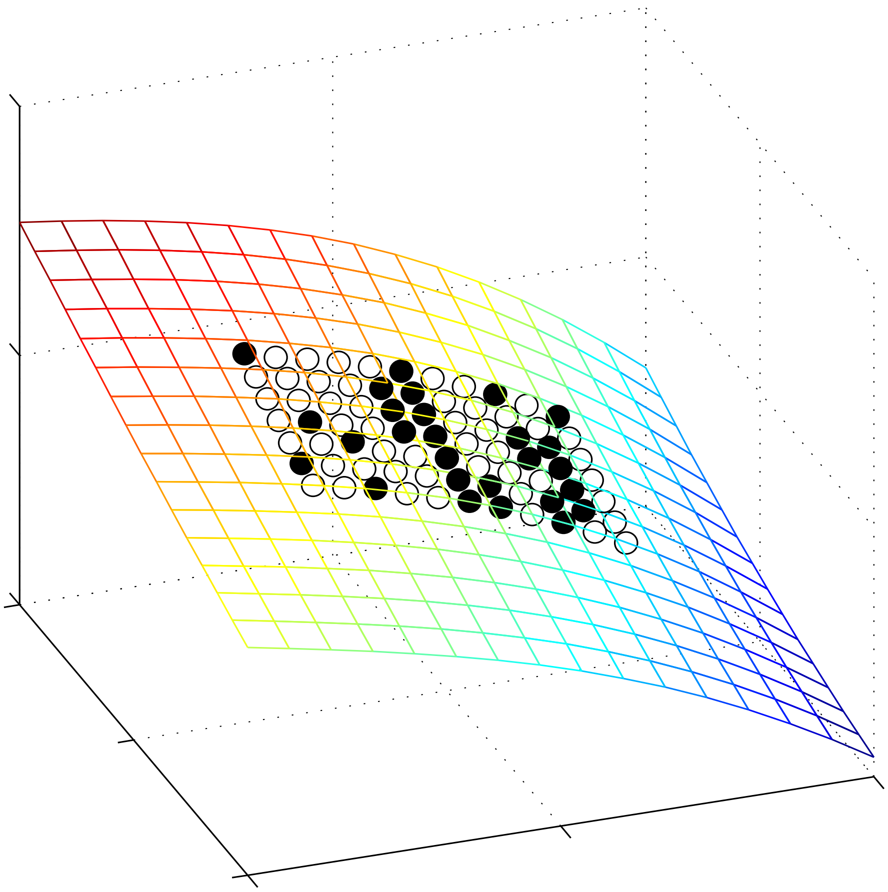


Mark J L Orr

Centre for Cognitive Science
Edinburgh University

November 1997

The *refmix* Data Set



Mark J L Orr[†]

Centre for Cognitive Science
Edinburgh University

November 1997

[†] mark@cns.ed.ac.uk, http://www.cns.ed.ac.uk/people/mark.html

Contents

Acknowledgements

This report was produced for Predictive Control Limited by Edinburgh University as part of a collaborative project, *Adaptive Radial Basis Function Neural Networks*, sponsored by the Engineering and Physical Science Research Council (research grant GR/L15388). The work was directed by Dr John Hallam (Artificial Intelligence), Prof Tom Leonard (Mathematics and Statistics) and Prof Alan Murray (Electrical Engineering) and supported by Prof David Willshaw (Centre for Cognitive Science). Thanks to David Sandoz and Harry Zhang of PCL for their cooperation.

1 Introduction

In this document we examine a data modelling problem and compare polynomial models with radial basis function neural networks [2] using a variety of training algorithms. The data set appears simple at first (figure 1) but in fact is more interesting than it looks.

1.1 The data

The data was supplied by Predictive Control Ltd. and known as the *refmix* data set. It consists of 77 samples of two independent variables (x_1 and x_2) and one dependent variable (y). The problem is to find the model which most accurately maps the relation between the independent and dependent variables, i.e. find the function $f : \mathcal{R}^2 \to \mathcal{R}$ which will likely give the least error when applied to new data sampled in the same way as the training set. Figure 1 shows a number of different views of the data.

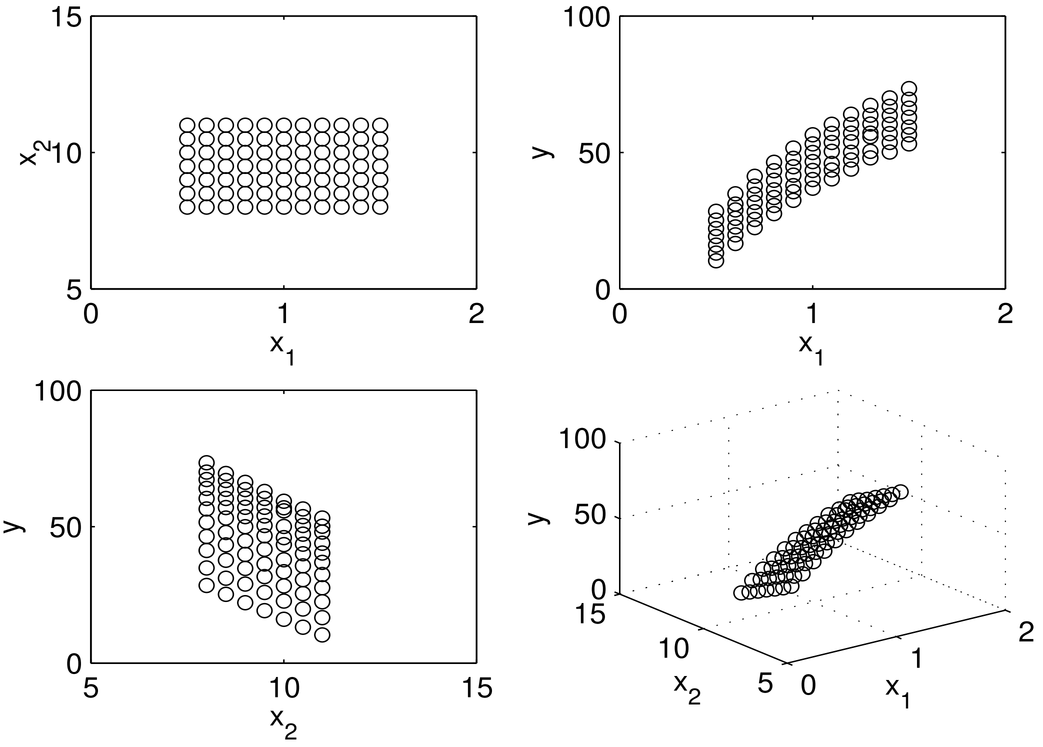


Figure 1: The *refmix* data set.

The data set is relatively small in terms of numbers of samples and dimensions of the input and output spaces (the number of independent and dependent variables, respectively). It also appears relatively uncorrupted by noise and looks as if it should be fit by a smooth function such as a polynomial.

We will use the notation ${}^i\mathbf{x}$ for the i-th input data vector (${}^i\mathbf{x} = [{}^ix_1 \; {}^ix_2]^\top$) where the index runs from $i = 1$ to $i = p$ (p is the number of samples, i.e. 77). Similarly, ${}^iy$ is the i-th output value. The variances of the input values (which will be useful later for deciding the shape of the radial basis functions) will be denoted

$$\sigma_1^2 \;=\; \frac{1}{(p-1)} \sum_{i=1}^{p} (\bar{x}_1 - {}^ix_1)^2 \tag{1}$$

$$\sigma_2^2 \;=\; \frac{1}{(p-1)} \sum_{i=1}^{p} (\bar{x}_2 - {}^ix_2)^2 \tag{2}$$

where $\bar{x}_1 = \frac{1}{p} \sum_{i=1}^{p} {}^ix_1$ and $\bar{x}_2 = \frac{1}{p} \sum_{i=1}^{p} {}^ix_2$ are the means. The actual figures come out at about $\sigma_1 \approx 0.3$ and $\sigma_1 \approx 1.0$.

1.2 The models

We will look at primarily linear-in-the-parameter models of the form

$$f(\mathbf{x}, \mathbf{w}) \;=\; \sum_{j=1}^{m} w_j \, h_j(\mathbf{x}) \tag{3}$$

where the basis functions $\{h_j\}_{j=1}^m$ are possibly nonlinear functions of the input vector $\mathbf{x}$ but are independent of the parameters $\mathbf{w} = [w_1, \ldots, w_m]^\top$. However, in the case of radial basis functions we will use an extra parameter, namely the size or width, r, on which the $\{h_j\}_{j=1}^m$ depend in a nonlinear way.

Two types of models are considered: polynomials and radial basis functions. In the former the basis functions consist of products of various powers of x_1 and x_2:

$$\{h_j(\mathbf{x})\} \;=\; \left\{1,\; x_1,\; x_2,\; x_1^2,\; x_1 x_2,\; x_2^2,\; x_1^3,\; x_1^2 x_2,\; \ldots\right\} \tag{4}$$

For radial basis functions, we use (mainly) elliptical Gaussian parameterised by a scaling parameter r and centred on some subset (or possibly all) of the input points:

$$h_j(\mathbf{x}, r) \;=\; \exp\left(-\frac{(x_1 - {}^jx_1)^2}{r^2\,\sigma_1^2} - \frac{(x_2 - {}^jx_2)^2}{r^2\,\sigma_2^2}\right) \tag{5}$$

where σ_1^2 and σ_2^2 are the variances along the coordinate axes in input space (1,2). Figure 2 illustrates the difference between polynomial and radial basis functions.

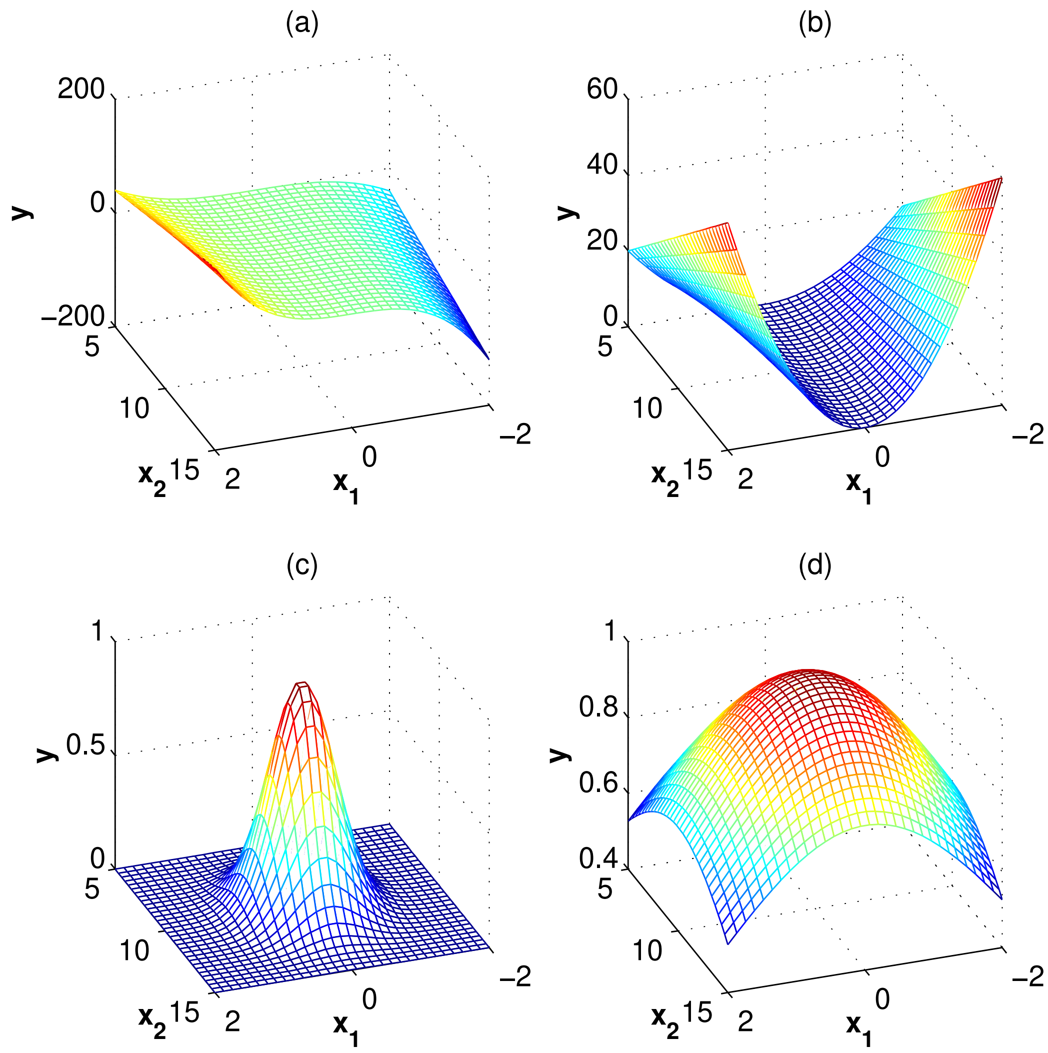


Figure 2: Typical polynomial and radial basis functions: (a) polynomial, $x_1^3 x_2$, (b) polynomial, $x_1^2 x_2$, (c) radial, centred on $(0, 10)$, size $r = 2$, (d) radial, centred on $(0, 10)$, size $r = 10$.

The radial functions are local: they die away sufficiently far from their centres, whereas the polynomials increase without limit. However, if the input data is confined to a region of input space smaller than the size of the radial functions then locality is effectively lost and each input point reacts to each basis function – see figure 2(d).

1.3 The cost function

We will use the cost function

$$C(\mathbf{w}, r, \lambda) = \sum_{i=1}^{p} \left(f({}^i\mathbf{x}, \mathbf{w}, r) - {}^i y \right)^2 + \lambda \sum_{j=1}^{m} w_j^2 \tag{6}$$

where for polynomials the dependence on r disappears. This cost consists of the usual squared error term to fit the model to the data plus a penalty term to suppress the size of the weights (called weight decay in the neural network literature [1] or ridge regression (RR) in statistics [3]). The latter is used to avoid over-fitting, where an unnecessarily complex model tries to fit the noise in the data. The trade-off between fitting the data and fitting the noise is controlled by the regularisation parameter λ. In some situations we will drop the weight-penalty term if we know the model is too simple to overfit the data.

By differentiating the cost (6) with respect to $\mathbf{w}$ and setting the result to zero the global minimum can be found. This occurs when $\mathbf{w} = \hat{\mathbf{w}}$ where

$$\hat{\mathbf{w}} = \left(\mathbf{H}^\top \mathbf{H} + \lambda\, \mathbf{I}_m \right)^{-1} \mathbf{H}^\top \mathbf{y} \tag{7}$$

and where $\mathbf{H} \in \mathcal{R}^{p \times m}$ is the design matrix,

$$H_{ij} = h_j({}^i\mathbf{x}) \tag{8}$$

$\mathbf{y} \in \mathcal{R}^p$ is the vector of output values

$$\mathbf{y} = [{}^1y \; {}^2y \; {}^3y \; \ldots \; {}^py]^\top \tag{9}$$

and $\mathbf{I}_m$ is the m-dimensional identity matrix.

Forward selection (FS), where some of the polynomial terms or radial basis functions are discarded (or more accurately, not selected), is an alternative technique to reduce the complexity of models likely to overfit the data [3]. The basic idea is to build up the model term by term (or basis function by basis function). At each step the term which most reduces the mean squared error is chosen to be added while a model selection criterion (e.g. GCV – see next section) is continually monitored. As soon as the criterion stops decreasing (the onset of over-fitting) the selection process halts and any remaining unselected terms are discarded. It's also feasible to have a regularisation term as well with λ continually adjusting to minimise GCV as the model builds up [2]. This is called regularised forward selection (RFS).

1.4 Model comparison criteria

The performance of any model in fitting the data would ideally be measured by the average squared error on a separate and uncorrupted data set. In the absence of such test data a common technique is to split the available data into a test set and a training set, train the model on the latter and measure its average squared error on the former. A more refined version of this is to split the available data into test and train sets in a number of different ways and to average over these partitions of the data as well. The partitions can be restricted to the case where there are always $p-1$ samples in the training set and only 1 sample in the test set (and therefore p possible partitions). This is called leave-one-out (LOO) cross-validation. The linear mathematics underlying the types of model we are considering means that it is unnecessary to train p different networks to calculate the LOO average error. Indeed it can be shown [2] that

$$\sigma^2_{\mathrm{LOO}} \;=\; \frac{1}{p}\,\mathbf{y}^\top\mathbf{P}\,\left(\mathrm{diag}(\mathbf{P})\right)^{-2}\mathbf{P}\,\mathbf{y} \tag{10}$$

where $\mathbf{P}$ is the matrix

$$\mathbf{P} \;=\; \mathbf{I}_p - \mathbf{H}\left(\mathbf{H}^\top\mathbf{H} + \lambda\,\mathbf{I}_m\right)^{-1}\,\mathbf{H}^\top \tag{11}$$

and $\mathbf{I}_p$ ($\mathbf{I}_p$) is the p-dimensional (p-dimensional) identity matrix. A related error measure is generalised cross-validation (GCV), given by

$$\sigma^2_{\mathrm{GCV}} \;=\; \frac{\mathbf{y}^\top\mathbf{P}^2\,\mathbf{y}}{\left(\mathrm{trace}(\mathbf{P})\right)^2} \tag{12}$$

We will use these estimated errors to compare different models (for example RBF networks with different basis widths, polynomials of different order, or polynomials and RBF networks).

Finally the number of free parameters in the model is

$$\gamma \;=\; p - \mathrm{trace}(\mathbf{P}) \;=\; m - \lambda\,\mathrm{trace}\left((\mathbf{H}^\top\mathbf{H})^{-1}\right) \tag{13}$$

Without regularisation, i.e. when $\lambda = 0$, γ is just equal to m, otherwise $\gamma < m$. This is intuitively reasonable as the purpose of regularisation is to reduce the flexibility of the model (i.e. decrease its degrees of freedom).

2 Model fits

In the following subsections various models are tried out on the data set. At the end of this section is a summary of the results.

2.1 Zero-order polynomial

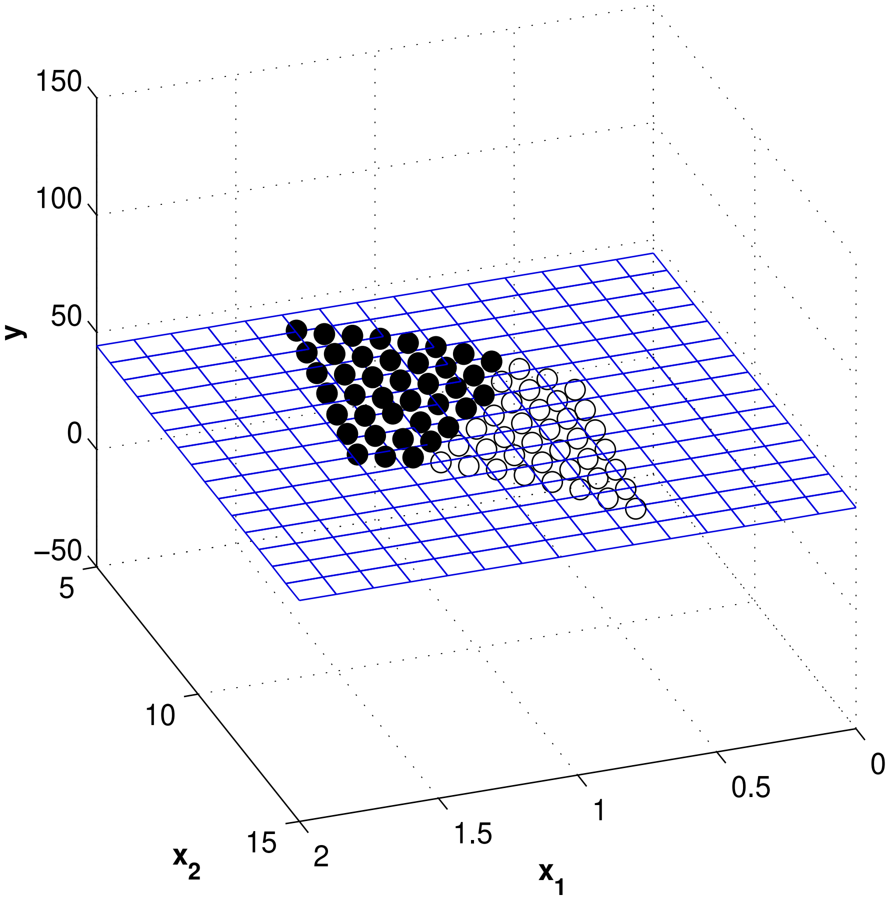


Figure 3: Here the model is just a constant, independent of x_1 or x_2. As in subsequent plots, data points which lie above the model surface are drawn filled, those below are open.

The simplest model we consider is one which involves just one free parameter and which is independent of either x_1 and x_2.

$$f(x_1, x_2, w_1) \;=\; w_1 \tag{14}$$

Since this is likely to underfit rather than overfit, we didn't try regularisation or forward selection. Besides forward selection from a set with only one member doesn't make much sense. The optimal value of the free parameter, $\hat{w}_1$, is, of course, just the mean of the data outputs (see figure 3). Unsurprisingly, the predicted accuracy for this model was very poor: $\sigma^2_{\mathrm{GCV}} = \sigma^2_{\mathrm{LOO}} = 239$.

2.2 First-order polynomial

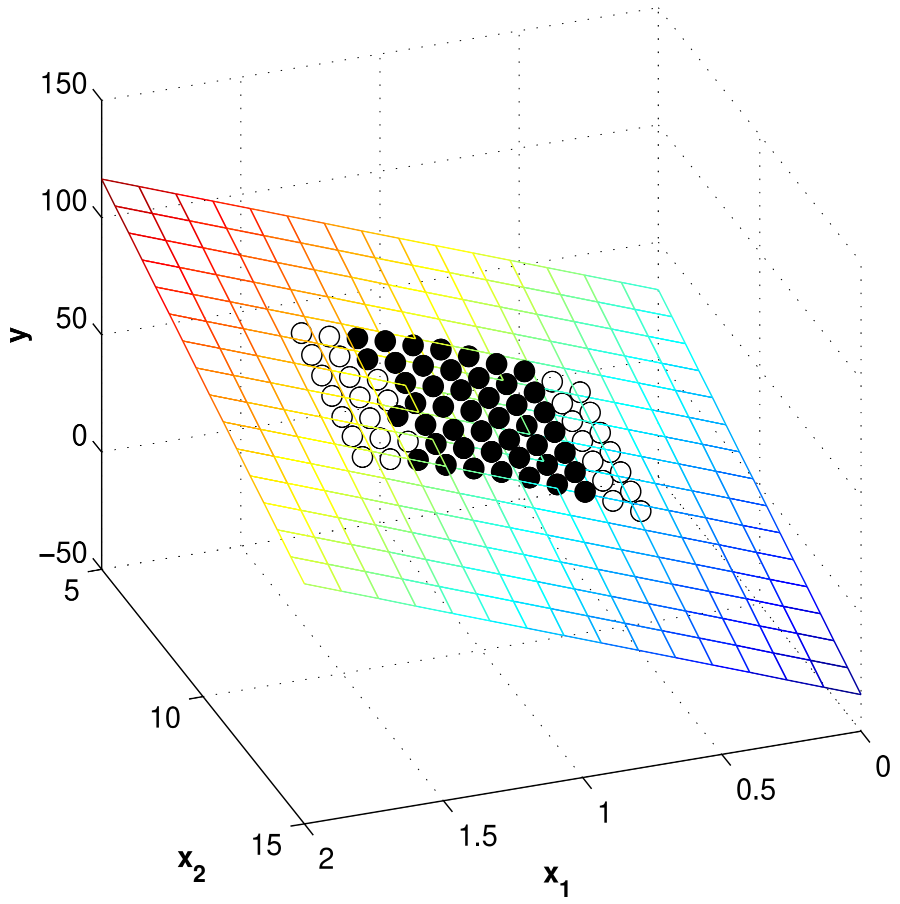


Figure 4: This time the model is linear in the dependent variables.

This next model is made up of a constant term and two linear terms and is now, unlike (14), dependent on x_1 and x_2 (see figure 4).

$$f(x_1, x_2, \mathbf{w}) \;=\; w_1 + w_2\, x_1 + w_3\, x_2 \tag{15}$$

The predicted errors of this model were hugely better than the previous, zero-order model: $\sigma^2_{\mathrm{GCV}} = 3.82$ and $\sigma^2_{\mathrm{LOO}} = 3.88$. However, the model is still too simple to need regularisation or forward selection.

2.3 Second-order polynomial

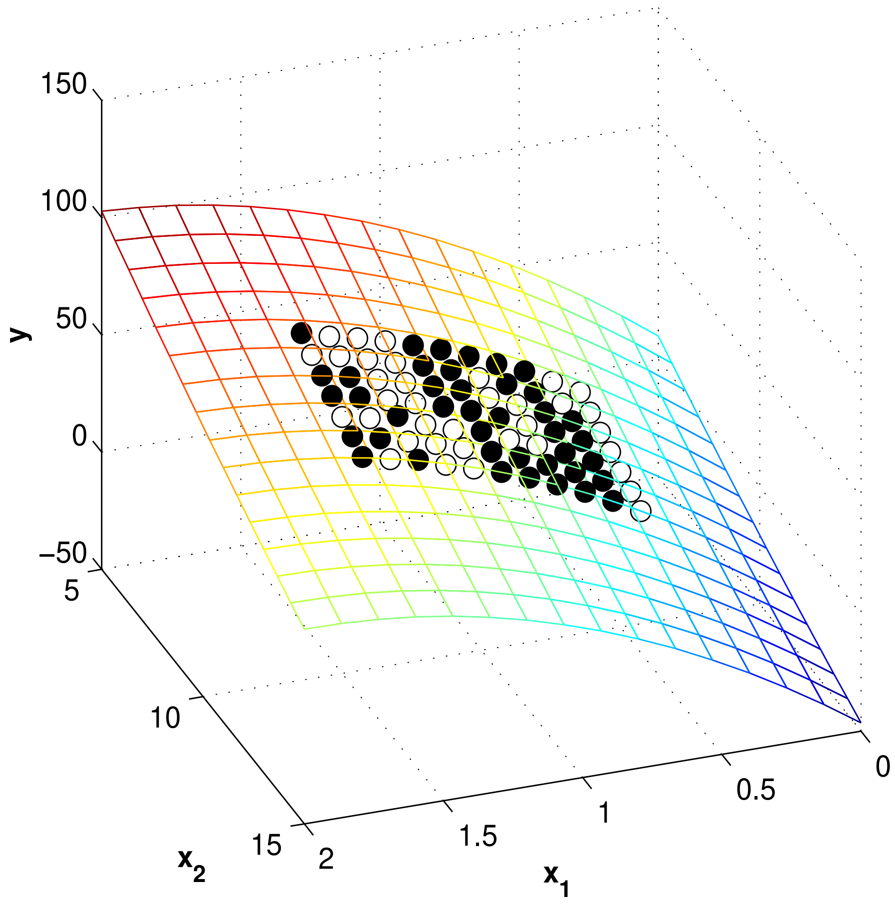


Figure 5: This model is quadratic in the dependent variables.

Second order terms in x_1 and x_2 are added to the previous linear model.

$$f(x_1, x_2, \mathbf{w}) \; = \; w_1 + w_2\, x_1 + w_3\, x_2 + w_4\, x_1^2 + w_5\, x_1\, x_2 + w_6\, x_2^2 \qquad (16)$$

This was a big improvement over the previous one, dropping the prediction errors down to $\sigma^2_{\mathrm{GCV}} = 0.168$ and $\sigma^2_{\mathrm{LOO}} = 0.163$. The patterns of points lying above and below the model surface (figure 5) are beginning to look more random, suggesting that the optimal model complexity (and possible over-fitting) is not far off if we continue to add yet more terms to the model.

2.4 Third-order polynomial

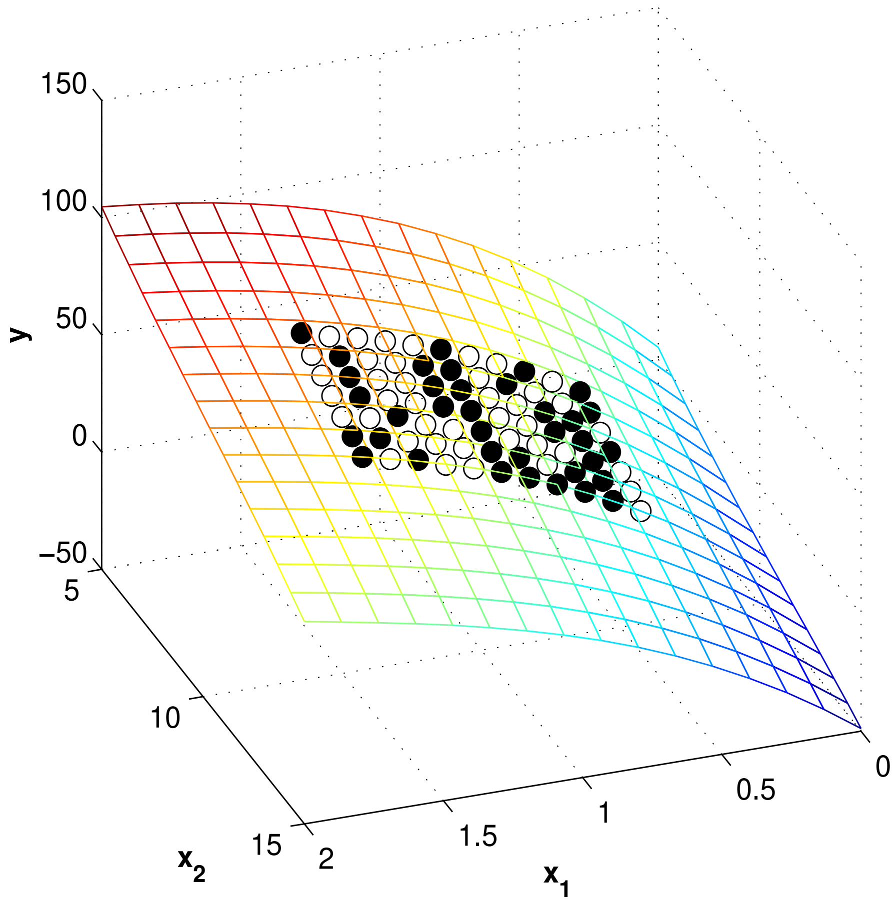


Figure 6: The best fit cubic model.

Next we add third order (cubic) terms in x_1 and x_2.

$$\begin{aligned} f(x_1, x_2, \mathbf{w}) &= w_1 + w_2\, x_1 + w_3\, x_2 + w_4\, x_1^2 + w_5\, x_1\, x_2 + w_6\, x_2^2 + \\ & \quad w_7\, x_1^3 + w_8\, x_1^2\, x_2 + w_9\, x_1\, x_2^2 + w_{10}\, x_2^3 \end{aligned}$$

The improvement in prediction error over the previous model is now quite small. The two estimates are down by about 6% to 0.159 (GCV) and 0.150 (LOO). The models are now getting close to over-fitting so it's appropriate to investigate regularisation and/or forward selection. The results are summarised in table 1, but essentially, while they result in a decrease in both GCV and LOO, the improvement is marginal. Forward selection chooses to discard only one term (in table 1 you can see γ differs by exactly 1 between cubic and cubic-FS models). Interestingly, however, the discarded one is the linear term $w_3\, x_2$ and not one of the higher order terms.

2.5 Fourth-order polynomial

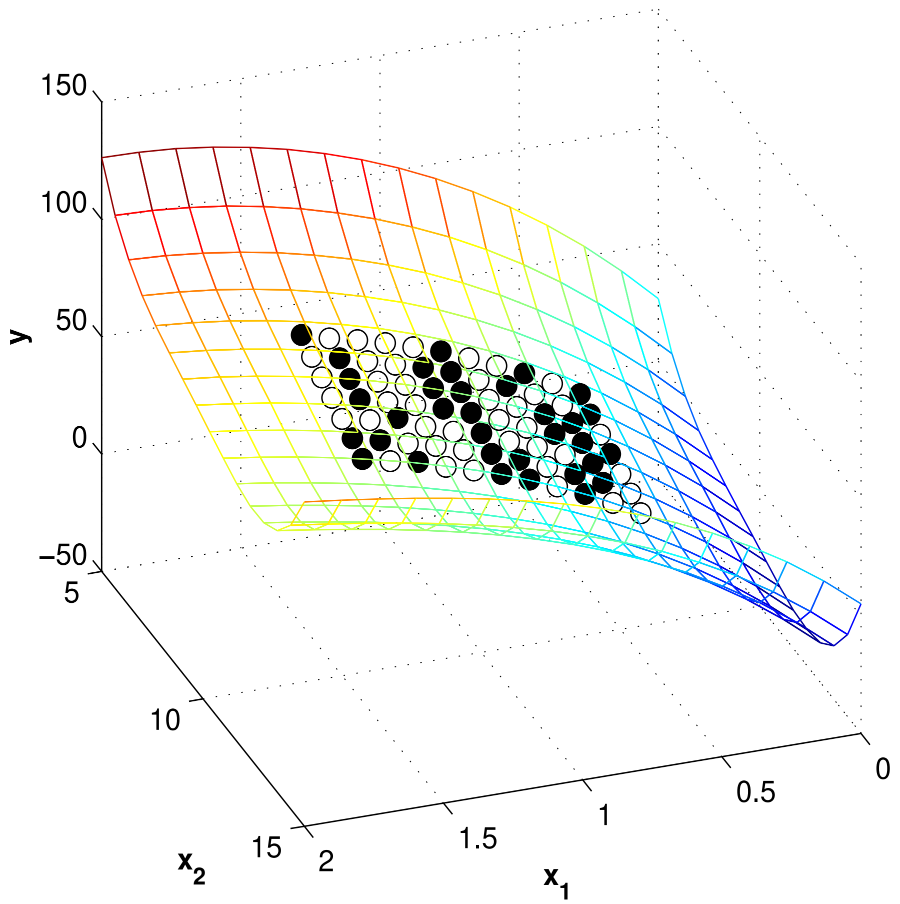


Figure 7: This model is quartic, i.e. has all terms up to fourth power in x_1 and x_2 and their products.

Finally fourth order terms are added to make the model quartic.

$$
\begin{aligned}
f(x_1, x_2, \mathbf{w}) \;\; = \;\; & w_1 + w_2\, x_1 + w_3\, x_2 + w_4\, x_1^2 + w_5\, x_1\, x_2 + w_6\, x_2^2 + \\
& w_7\, x_1^3 + w_8\, x_1^2\, x_2 + w_9\, x_1\, x_2^2 + w_{10}\, x_2^3 + \\
& w_{11}\, x_1^4 + w_{12}\, x_1^3\, x_2 + w_{13}\, x_1^2\, x_2^2 + w_{14}\, x_1\, x_2^3 + w_{15}\, x_2^4
\end{aligned}
$$

This time the predicted errors are higher, back up to 0.184 (GCV) and 0.173 (LOO) which is worse than both the cubic and quadratic models and indicates that the model is too complex. The results of applying forward selection (FS), regularisation (RR) and their combination (RFS) are given in table 1. The pure quartic clearly over-fits significantly and the complexity reduction makes the model competitive (though still just marginally worse) than the cubic models. Forward selection discards five terms all together: three low order terms, w_1, $w_3\, x_2$ and $w_6\, x_2^2$ and two high order terms, $w_{12}\, x_1^3\, x_2$ and $w_{13}\, x_1^2\, x_2^2$.

2.6 Regularised RBF network

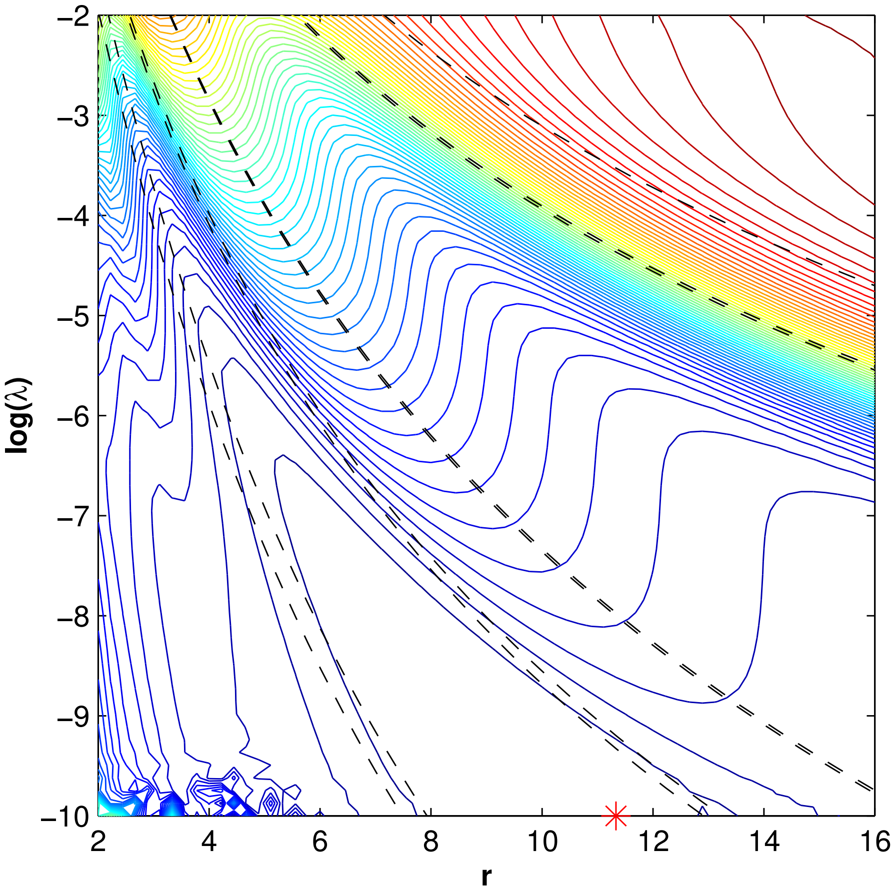


Figure 8: A contour plot of $\log_{10}(\sigma^2_{\mathrm{GCV}})$ as a function of $\log_{10}(\lambda)$ and r in a regularised RBF network. The minimum is marked by a small red star and the model corresponding to it is shown in figure 9.

.

We now investigate radial basis function models and initially follow the simple scheme whereby each data point contributes one basis function to the network ($m = p$), the basis function sizes are parameterised by a single parameter r and regularisation is used. Clearly, we do need some form of complexity reduction because otherwise the model will have $p = 77$ degrees of freedom but from the results for the polynomials models we already know that the optimal number of degrees of freedom is likely to be somewhere around 10.

Since the problem at hand is small in terms of the number of data points it is feasible to search for the optimal values of λ (the regularisation parameter) and r (the basis functions' size) by stepping through a two-dimensional grid in (λ, r)-space looking for the minimum GCV. Figure 8 shows the results of such a search.

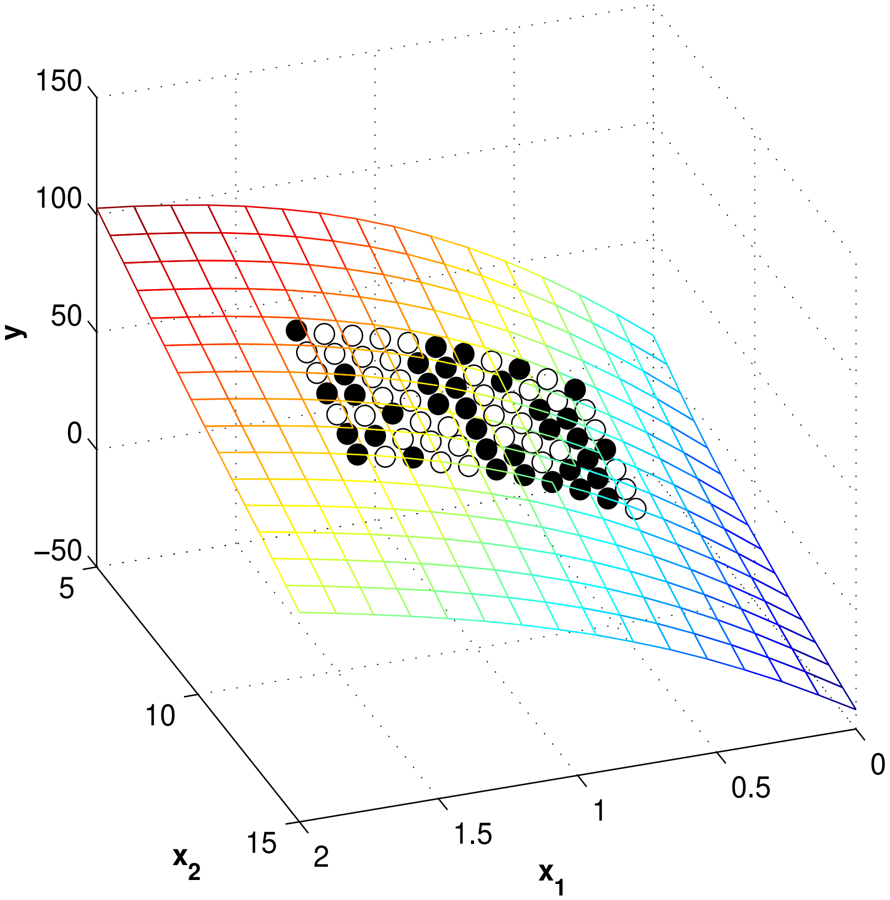


Figure 9: The RBF model with the best values for λ and r found from figure 8.

The minimum predicted error in the grid of $(\lambda,\, r)$ values is at $\lambda = 1 \times 10^{-10}$ and $r = 11.3$ where $\sigma^2_{\mathrm{GCV}} = 0.160$, $\sigma^2_{\mathrm{LOO}} = 0.152$ and $\gamma = 9.68$. These figures are very similar to those for the cubic model (see table 1). Figure 9 depicts the RBF model using the minimum error values of λ and r. It's quite hard to tell this apart from figure 6 for the cubic model.

The dashed lines in figure 8 depict the variation with r of some of the eigenvalues of the matrix $\mathbf{H}^\top\mathbf{H}$ (μ_4 to μ_{12} in descending order, going from top-right to bottom-left). Clearly there is a close relationship between the eigen-structure of this matrix and the optimal value of λ. This sort of behaviour is quite common (I've seen the same phenomenon in other applications of RBF networks) and it would be interesting to investigate it further.

2.7 Forwardly selected RBF network

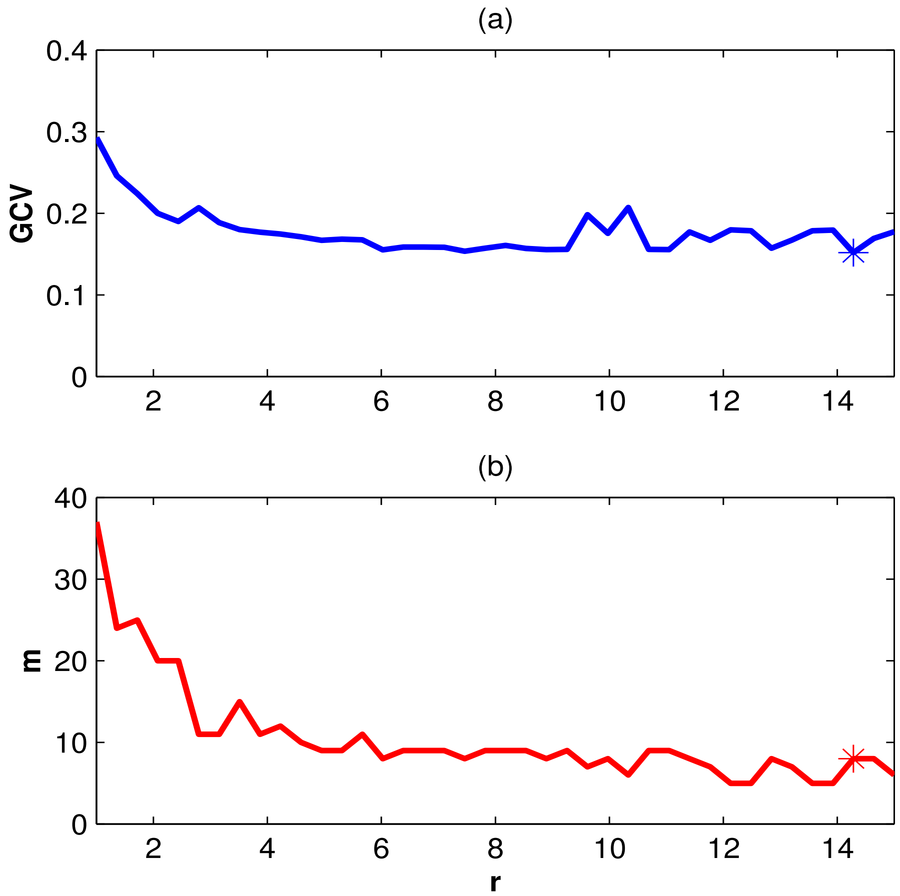


Figure 10: (a) GCV and (b) m (the number of selected centres) as a function of the radius r for RBF networks constructed by forward selection. The best (minimum GCV) is shown starred and the corresponding model is shown in figure 11.

The alternative to regularisation is forward selection which explicitly reduces the number of parameters in the network by pruning (or, more accurately, by not selecting) some of the basis functions. The two plots of figure 10 show how GCV and the number of selected centres varies as r increases from 1 up to 15, with the position of minimum GCV marked by a star. As for the regularised network, a large radius is preferred and the optimal number of free parameters ($\gamma = m$ in this case) is just under 10, close to the number of terms in the best polynomial models.

Regularised forward selection is marginally worse (see table 1). However, these two models, RBF-FS and RBF-RFS, are the top two performing models on the data set although it should be emphasised the winning margin is very small. As can be seen from the table, there is only about 5% or 6% improvement from the ninth ranked model (quartic-RFS) to the best model (RBF-FS).

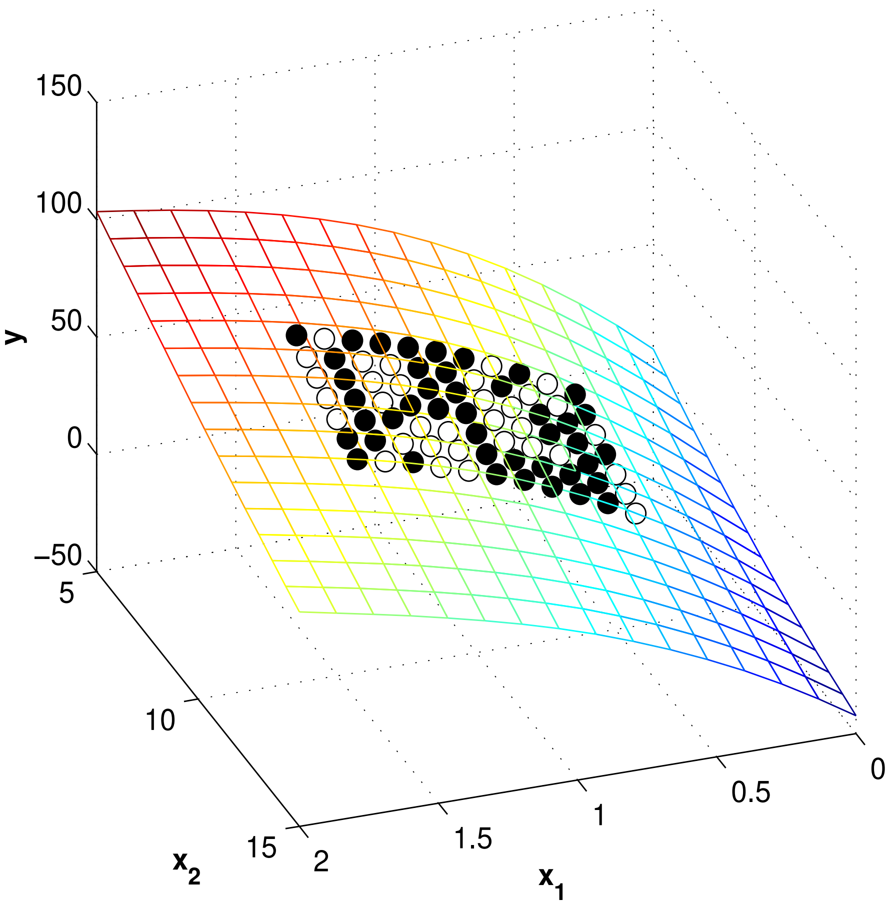


Figure 11: The RBF model constructed by forward selection with the best value of r, as indicated in figure 10.

Figure 11 shows the model with (marginally) the best estimated prediction performance on the data set, a forwardly selected RBF using the optimal size (a whopping great $r = 14.3$) from figure 10. This network contained just 8 of the original 77 radial basis functions which the forward selection algorithm was free to choose from.

2.8 Non-local basis functions

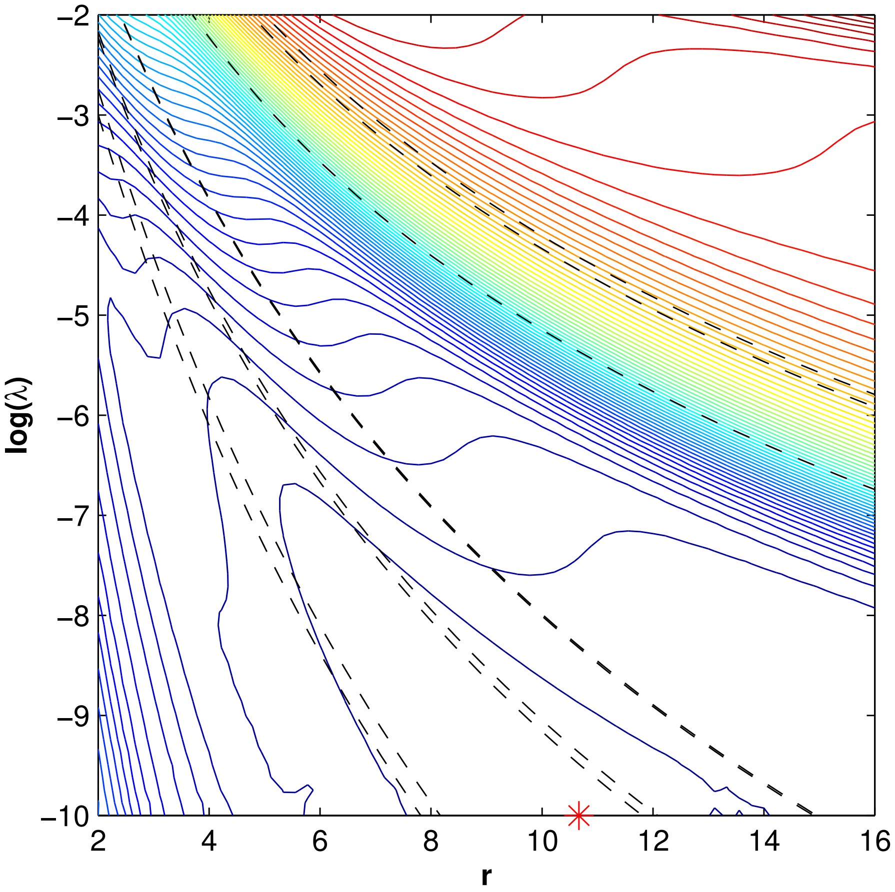


Figure 12: Similar to figure 8 except that the RBFs are multiquadric, rather than Gaussian.

Some practitioners of RBF networks prefer to use non-local basis functions. Such functions increase monotonically away from their centres, instead of decreasing. An example is the *multiquadric*,

$$h_j(\mathbf{x},\, r) \;=\; \sqrt{1 + \frac{(x_1 - {}^j x_1)^2}{r^2\,\sigma_1^2} + \frac{(x_2 - {}^j x_2)^2}{r^2\,\sigma_2^2}} \tag{17}$$

(compare with equation (5)). A search in $(\lambda,\, r)$-space (figure 12), similar to figure 8 but using multiquadric basis functions produced almost identical values of σ^2_{GCV} and σ^2_{LOO} at the optimum. Similarly, no improvement was detected in the forward selection algorithm when non-local basis functions were used.

2.9 Summary

The following table summarises the results of fitting the various models to the *refmix* data set.

model	γ	MSE	GCV	LOO	λ	r
RBF FS	8	0.122	0.152	0.145	n/a	14.3
RBF RFS	7.9	0.124	0.154	0.147	7.7×10^{-11}	8.5
cubic RFS	9.0	0.122	0.156	0.147	1.2×10^{-5}	n/a
cubic FS	9	0.122	0.156	0.148	n/a	n/a
cubic RR	9.5	0.121	0.158	0.149	3.2×10^{-5}	n/a
cubic	10	0.121	0.159	0.150	n/a	n/a
RBF RR	9.7	0.122	0.160	0.152	1.0×10^{-10}	11.3
quartic FS	10	0.121	0.160	0.154	n/a	n/a
quartic RFS	9.7	0.123	0.161	0.153	6.0×10^{-5}	n/a
quartic RR	10.1	0.123	0.167	0.162	1.1×10^{-3}	n/a
quadratic	6	0.143	0.168	0.163	n/a	n/a
quartic	15	0.119	0.184	0.173	n/a	n/a
linear	3	3.53	3.82	3.88	n/a	n/a
constant	1	233	239	2.39	n/a	n/a

Table 1: Performances of the various models applied to the data set. The rows are ordered by increasing GCV, then increasing LOO. The best performers all have $\gamma \approx 8$ and (if they're RBFs) large values for r.

3 Conclusion

The smoothness of the data set and the performance of the cubic model, especially with forward selection, suggest the underlying function may indeed be a third-order polynomial, with perhaps a missing-term ($w_3\, x_2$ – see subsection 2.4). However, in that case why does the RBF model (with forward selection) seem to perform better, in the sense of a marginally lower GCV predicted error? Is it possible that, if the data really is sampled from a polynomial, an RBF model could still appear to be the best performer as judged by GCV alone?

To try to answer this question a simple experiment was performed. First, the original data set was fit by a cubic model (missing the $w_3\, x_2$ term) and the residual mean square error noted. New raining sets were then generated, using the same inputs as the original data set, but employing the cubic model generate the outputs and the size of the residual error for adding random Gaussian noise. 1000 such training sets were produced.

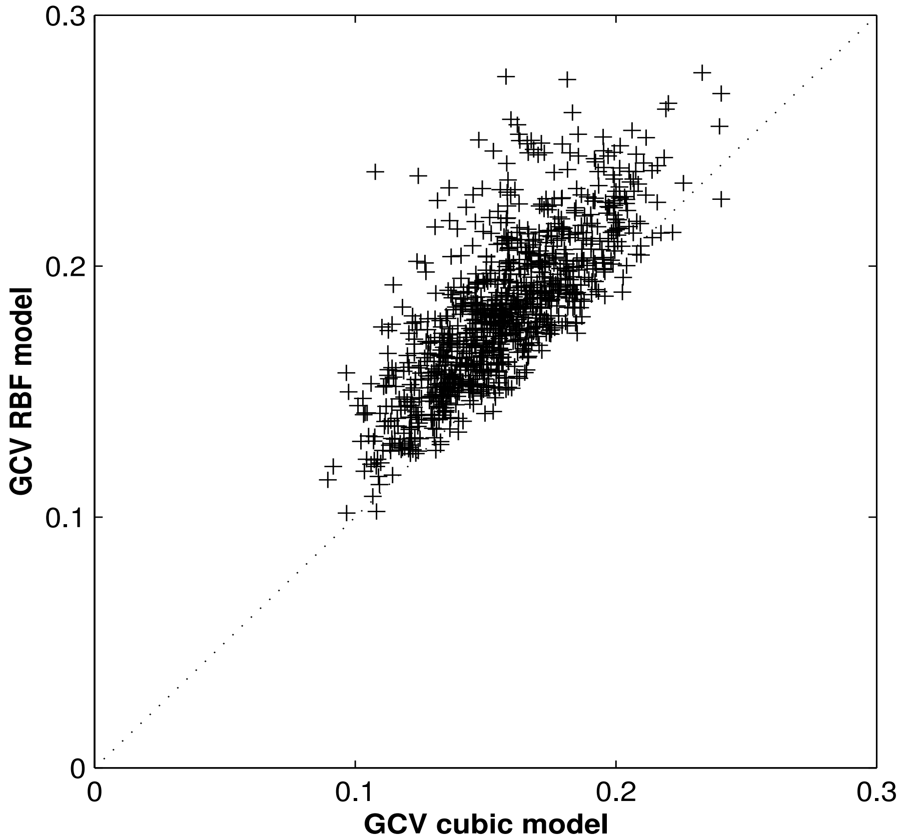


Figure 13: GCV scores from cubic and RBF models compared on simulated data sets where the underlying function really is a cubic.

Each training set was then fit by the correct model (a third-order polynomial with a missing term) and also by an RBF network using forward selection and a radius equal to the optimal value found in subsection 2.7 ($r = 14.3$). The GCV predicted errors from each model are plotted in figure 13, with each point representing an individual training set.

The idea of the experiment was as follows: if the GCV error from the cubic fit was consistently less than that from the RBF fit then, since in table 1 there isn't much difference between cubic and RBF models, this would be evidence that the underlying function is not a polynomial. Unfortunately, as can be seen from figure 13, there is only a trend for the average GCV error to be higher for the RBF model, with a significant probability that the two models will have approximately equal GCV errors, as in our results (table 1). Thus we can't say with any confidence that the underlying function is not a simple polynomial.

So the main conclusions are as follows:

- RBF networks and polynomial models with around 8 or 9 degrees of freedom give a good fit.
- The RBF networks prefer a large scaling parameter (which produces a very smooth function).
- The model selection criteria we have used (GCV, LOO) are not quite able to distinguish which model is best (RBF or cubic polynomial).
- If the true function is a polynomial it may be missing (or have a small coefficient on) the term linear in x_2.

References

[1] C. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

[2] M.J.L. Orr. Introduction to radial basis function networks. Technical report, Centre for Cognitive Science, Edinburgh University, 1996. Download from `http://www.cns.ed.ac.uk/people/mark/papers/rbf.html`.

[3] J.O. Rawlings. *Applied Regression Analysis.* Wadsworth & Brooks/Cole, Pacific Grove, CA, 1988.