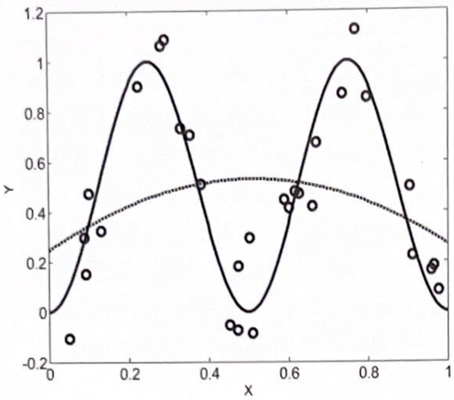
irrelevant inputs cannot be detected and they contribute to noise during RBF training.

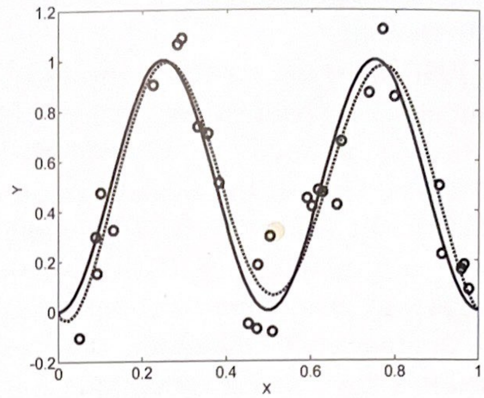
RBF network model complexity is controlled by two factors: the number of basis functions, or clusters, and their width parameters. There is usually a trade-off between these two factors for optimal model selection. That is, the model complexity grows with a larger number of clusters and also with a smaller width. This ambiguity can be resolved by using the number of basis functions m as the only user-defined complexity parameter, and automatically selecting the width parameter via internal cross-validation. This version of RBF training algorithm is used in a toy example presented next.

Example 6.3: Application of RBF network for univariate regression. This example illustrates estimation of a sine-squared target function, using synthetic data set previously used in Example 6.1. The number of RBF basis functions m is the only user-defined complexity parameter. Several RBF models for m = 2, 5, 10, 20 estimated from the same training data set of 30 samples are shown in Fig. 6.11. Clearly, the model with 2 basis functions underfits the data, the model with 5 basis functions yields the best estimate, and the model with 20 basis functions shows significant overfitting.

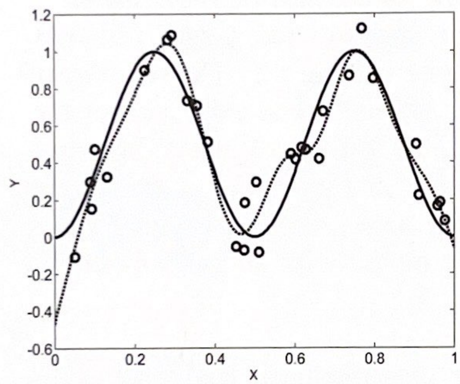
RBF networks can be also used for classification. RBF classifiers usually minimize the squared loss, so they are computationally equivalent to RBF networks for regression. These RBF classifiers effectively implement multiple-response regression, using *1-of-J* encoding for output class labels, as explained in Chapter 5. RBF training for classification follows the same two-step procedure described above for regression, where the center and width parameters are estimated first (via unsupervised learning) and then linear coefficients are estimated via least squares fitting. As in the case of MLP classifiers, the classification error (rather than squared loss) should be used for model selection. RBF classifiers become particularly attractive when class distributions can be well approximated by a small number of clusters. In this case, representative clusters can be identified during the unsupervised learning stage of RBF network training.



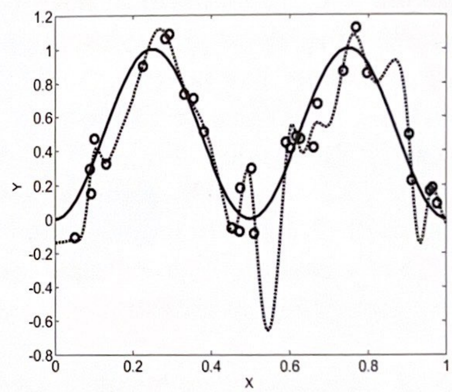
RBF Model with 2 Units



RBF Model with 5 Units



RBF Model with 10 units



RBF Model with 20 units

FIGURE 6.11 RBF network for univariate regression. The true target function is shown in solid line and its estimate in dashed line.

Example 6.4: RBF network for binary classification.

This example illustrates application of RBF network to binary classification, using Ripley's data set, introduced earlier in Section 6.3. The decision boundary formed by an RBF classifier with 4, 9 and 25 units is shown in Fig. 6.12. The RBF software used in this example requires specification of a single parameter (the number of RBF basis functions), and the width is determined automatically via 5-fold cross-validation on the training data. Note that during unsupervised learning stage, the RBF units are placed in the region of the input space where the density of input data samples is high. For example, in Fig. 6.12(a), four units are placed near the centers of 4 Gaussians used to generate Ripley's data set. RBF classifier with 9 units seems to have better decision boundary than the other two models, which exhibit underfitting (with 4 units) and overfitting (with 25 units). Evaluation of these models using 1,000 test samples confirms this interpretation (see Problem 6.11). Note that there is not much overfitting, even when the number of RBF units is large (~25).

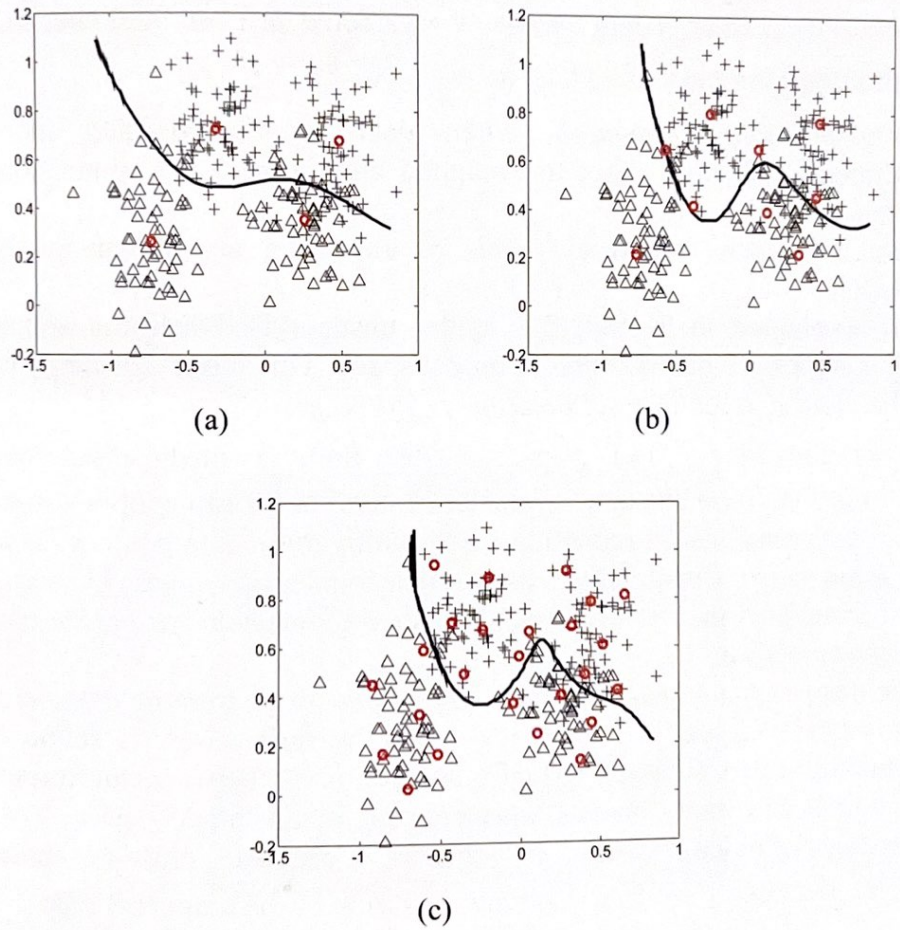


FIGURE 6.12 Decision boundary obtained by RBF classifier for Ripley's training data. RBF unit centers are shown in red.
(a) RBF model with 4 units.
(b) RBF model with 9 units.
(c) RBF model with 25 units.