Kernel Methods and Pattern Analysis
Assignment 1

Parth Joshi (CS09B051) & Sudharshan GK (CS13M050) Umesh Kanoja (CS13M024)

February 20, 2014

1 Methodology

1. **Generating Univariate Data :** For the case of univariate data, we used the given function $f(x) = \exp(\sin(2\pi x)) + \ln(x)$, adding random gaussian noise with mean 0 and standard deviation 0.3 to generate target output values for a given x. Rather than generating fresh data wherever required, we generated a single large dataset initially and saved it to file. To obtain datasets of different sizes, we then use a *bootstrapping* mechanism and sample the required number of points with repetition from the same dataset. The complete dataset generated consists of 10000, 1000 and 1000 samples for training, testing and validation respectively.

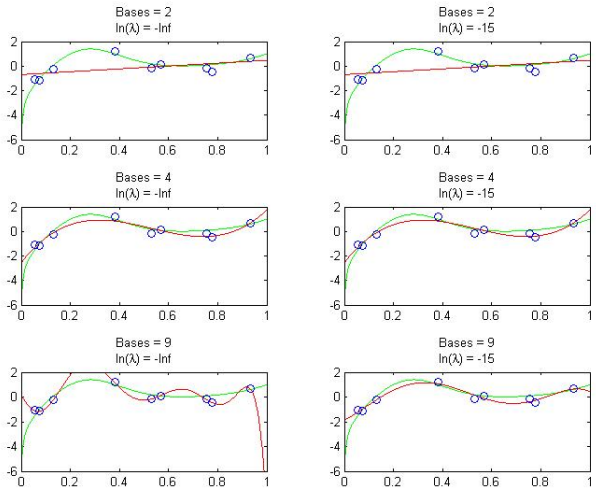
2. **Computing Variance for Gaussian Basis Functions :** Our implementation allows the setting of the width for the Gaussian basis functions empirically or using the data to compute a measure of the variance. For the latter case, we use the average variance of the data as computed from the trace of the covariance matrix.

3. **Normalization of Data :** The datasets were all normalized to the range $[0, 1]$ as part of pre-processing since we observed a lot of variance in the range of values the different fields of the dataset were taking (particularly for the multivariate data) which made the matrices containing the datasets badly conditioned.

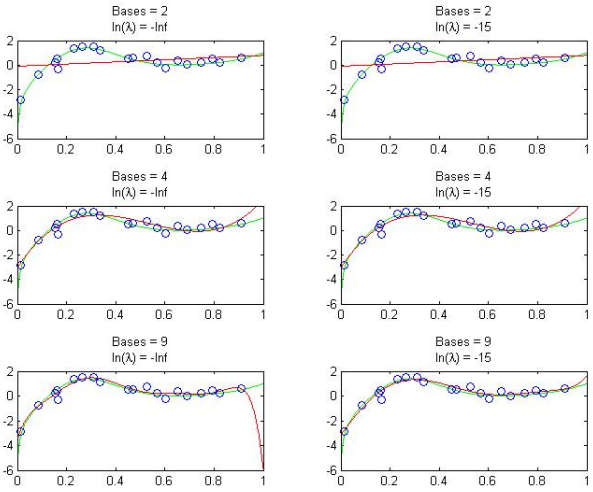
4. **Use of Root Mean Square Error :** Instead of directly using the mean squared error (MSE) for our plots, we have made use of its square root since it gives more meaningful ranges for the error value with the error being expressed in the same units as the original data.

2 Univariate Dataset

2.1 Plot 1a : Comparison of Plot of Aproximated Function vs Model output for Different Complexity and Regularization



(a) Plot for N=9



(b) Plot for N=20

Observations

- For the smaller number of data points in Fig 1a with no regularization, the 1 degree polynomial gives a poor fit, the 3 degree polynomial gives a good fit while the degree 8 polynomial gives a perfect fit to the training data. However, it is evident that the degree 3 polynomial fits the function $f(x)$ to be approximated much better than the higher degree one.
- After we add a regularization term, the higher degree polynomial gives a better approximation to $f(x)$.

- When we increase the size of the training dataset in Fig 1b, the higher degree polynomial produces an excellent approximation to $f(x)$ as compared to those of lower degree.
- Again, regularization helps improve the approximation but the effect is not as pronounced as in the earlier case.

Inferences

- As we increase the complexity of the model (the degree of the polynomial), the capacity of the model to describe the features of $f(x)$ improves but in the absence of sufficient data, highly complex models may get finely tuned to the random noise in the data, resulting in overfitting.
- Ridge regression using a regularization term helps to overcome the overfitting problem if the number of data points available is low. It can be observed that the regularization parameter is a means to control the effective model complexity.
- The overfitting problem does not arise if we have sufficient number of training examples.
- The regularization parameter is a tool to control model complexity. However, when the model itself is very simple (very few basis functions) the effect of the regularization parameter is negligible.

2.2 Plot 1b : Comparison Plot of Bias and Variance for multiple training sets for Different Complexity and Regularization Parameter values

Observations

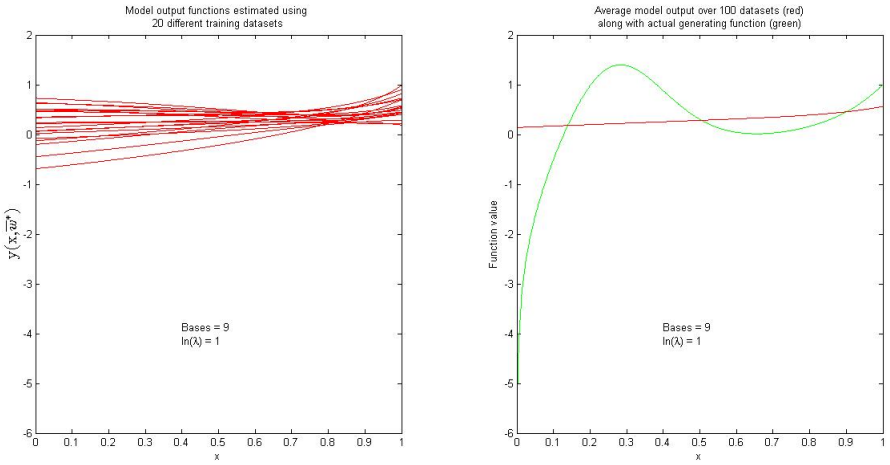
- The left hand plots show the estimated output functions for 20 different datasets. The plots serve to illustrate the variance in the model and its sensitivity to the choice of dataset.
- The right hand plots show the average of the estimated output obtained from 100 different datasets along with the true function $f(x)$. The deviation of the average from $f(x)$ depicts the bias in the model.
- It is evident from Fig 1 that as we increase the value of λ from $\exp(-20)$ through $\exp(-10)$ to 1 for a fixed number of bases, the variance steadily decreases but the bias increases.
- Conversely, we can see from Fig 2 that as the number of basis functions is increased for a given λ, the variance increases while the bias reduces.

Inference

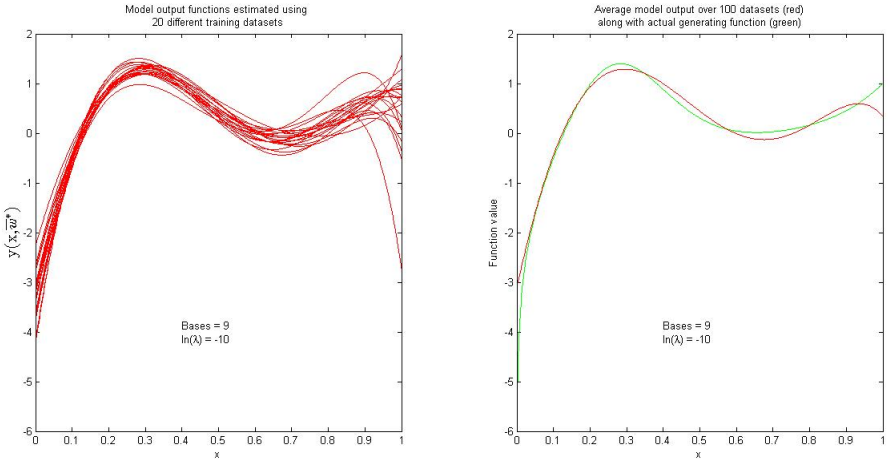
- There is a trade-off between the bias and the variance as the effective model complexity varies.
- If the regularization parameter λ is made too large as in the case of Fig 1a, it becomes the dominating factor and even a high model complexity of a degree 9 polynomial is unable to prevent severe underfitting of the data.

Figure 1: Varying $\ln(\lambda)$ keeping basis dimension fixed

(a)



(b)



(c)

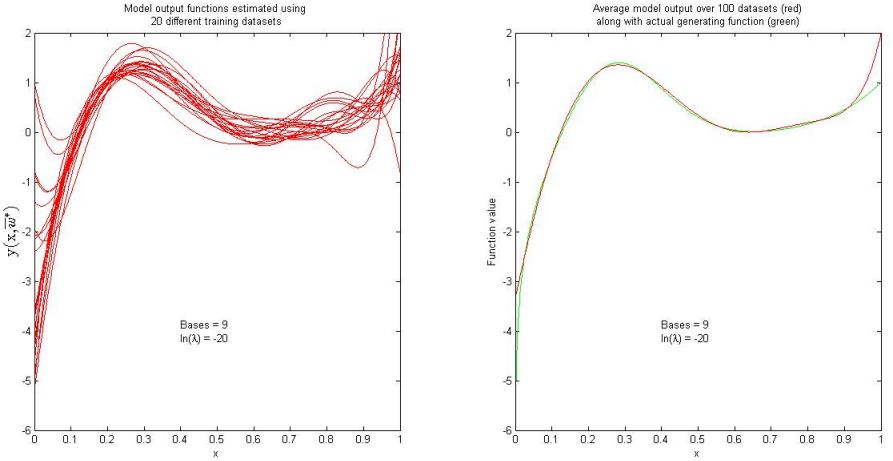
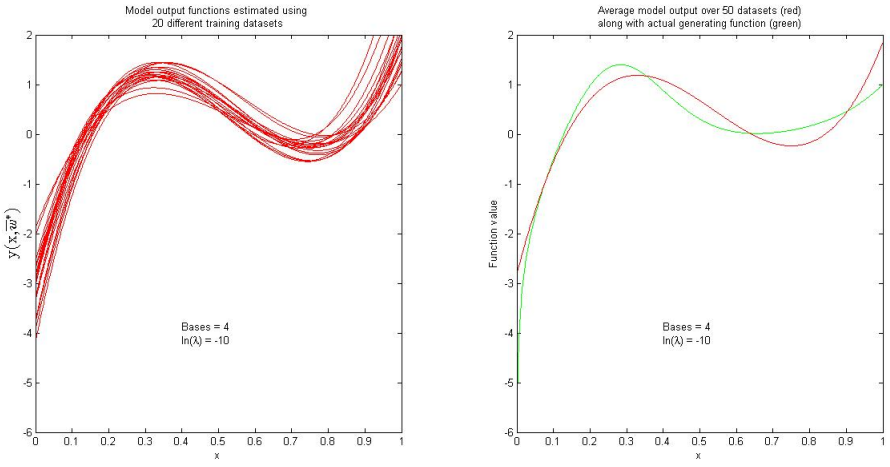
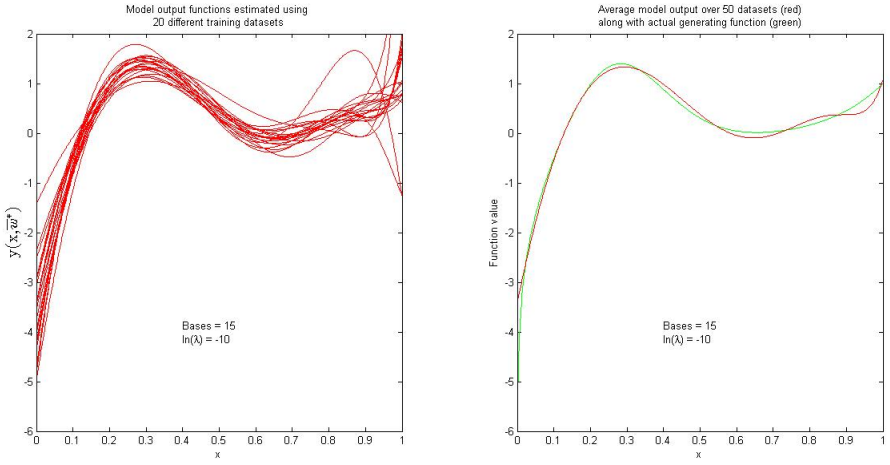


Figure 2: Varying basis keeping $\ln(\lambda)$ fixed

(a)



(b)



2.3 Plot 2: Plot of the squared bias, variance, average loss and the error on validation data with varying effective model complexity

Figure 3

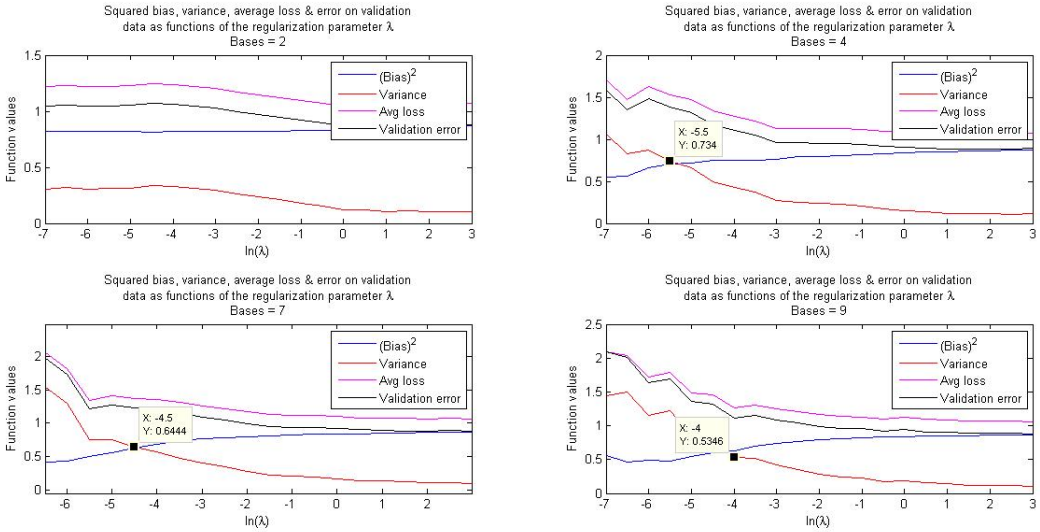
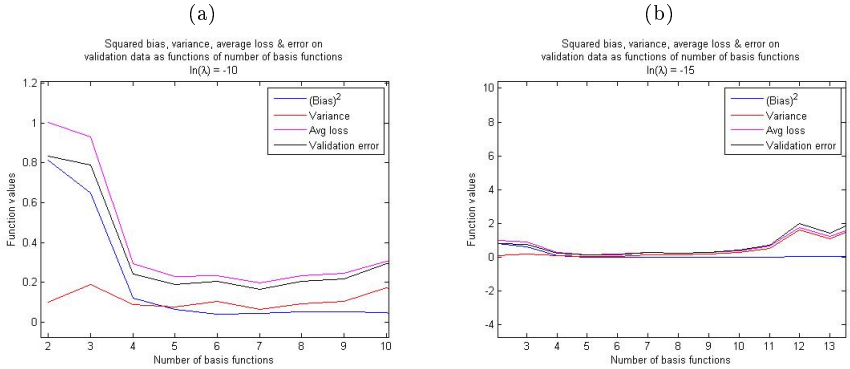


Figure 4



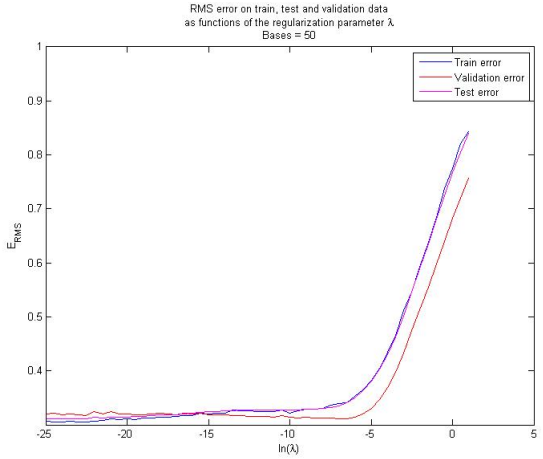
Observations

- For a given number of basis functions, the variance decreases with increasing λ while the bias increases.
- The error on validation data and the average loss always lie above the bias and variance curves.
- As the number of basis functions increases, the value of $\ln(\lambda)$ at which the bias and variance balance each other i.e. the best model complexity tends to increase.
- Keeping the value of λ fixed, we can see that the bias decreases while the variance increases with increase in the dimensionality of the basis.
- The point where the bias and variance curves intersect in this case also tends to move towards the right as the value of λ increases.

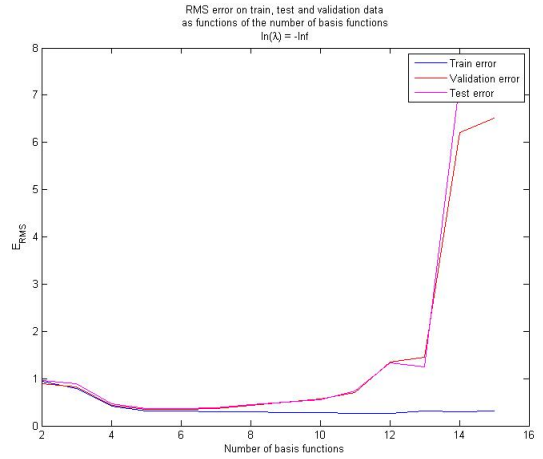
Inferences

- It is worth noting that the number of basis functions and the regularization parameter work in tandem to determine the effective complexity of the model. The size of the training dataset being the same, higher dimensionality of basis increases complexity while higher λ tends to reduce complexity.
- Thus if we increase the number of basis functions, in order to avoid increasing the variance of the model and making it sensitive to the choice of training examples, we must increase the value of λ as reflected in the shift of bias-variance intersection point.
- Similarly if we increase the value of λ, then, all other things being equal, we should also increase the number of basis functions in our model to avoid over-regularization which will increase the bias inherent in the model.

2.4 Plot 3 : Plot of root mean squared error (E_{RMS}) on training, validation and test data for different model complexities and λ values



(a) E_{RMS} vs $\ln(\lambda)$ for number of bases $= 50$



(b) E_{RMS} vs number of bases for $\ln(\lambda) = -Inf$

Observations

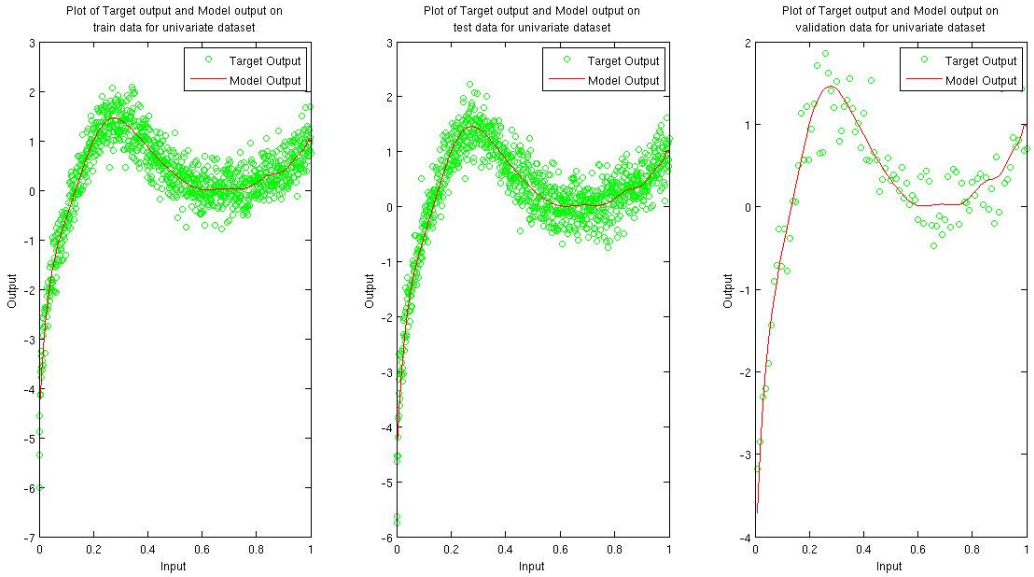
- As we increase the number of basis functions, the error on training data decreases while that on test and validation data increases. The minimum test / validation error occurs with ≈ 5 bases.
- The initial training error for the second plot is very small since a large number of basis functions is chosen to describe the trend with $\ln(\lambda)$. This results in overfitting. The error then shows an increasing trend.
- The validation error initially decreases gradually till the value of $\ln(\lambda) \approx -8$. Beyond this, it sharply increases.
- The number of data points used for the second plot is much higher than the first because we use a larger number of basis functions to get a meaningful plot.

Inferences

- The trend of training, test and validation error is as expected with change in the number of bases. The training error keeps declining as the model becomes more tuned to the data while the test and validation error increases after a point.
- For very large values of λ, the error increases since the model complexity effectively becomes very low.
- The validation error displays a minima at $\ln(\lambda) \approx -8$. However, the test data at this point is much higher. Hence a lower value of the regularizaton parameter would be a better choice.

2.5 Plot 4 : Plot of model and target outputs for the training, validation and test data

Figure 5: Number of basis functions = 50, $\ln(\lambda) = -25$



Observation

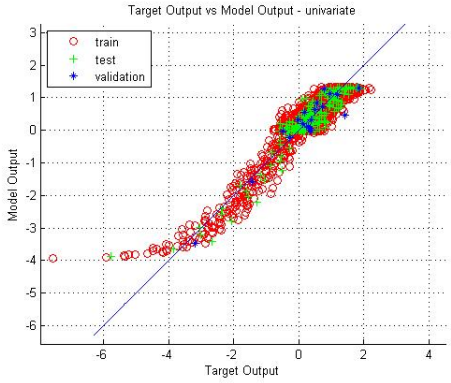
- All three datasets training, test and validation are well-explained by the model, even for points close to 0 and 1, the limits of the range of values.

Inferences

- Since we have chosen a large number of points for training, we have the luxury of setting our polynomial approximator to be of high degree (50 in this case) even without the need for regularization.
- The high model complexity ensures that even the points at the edges of the range are well-explained by the model (which will usually not be the case).

2.6 Plot 5 : Scatter plot of target output vs model output for training, validation and test data

Figure 6



Observation

- The model output gives a very good correlation with the target output with nearly all the plot points lying along the line $y = x$.
- It can be seen that the density of points in the the range $[0, 2]$ is much greater than that elsewhere on the plot.
- Only in the south-west corner of the graph does the correlation deviate from the ideal position.

Inferences

- For the univariate dataset, polynomial basis functions can be used to obtain a very close approximation to the original function $f(x)$.
- The points of low correlation are located at the edge of the graph. This can be attributed to the sparsity of the training data in that region and the finiteness of the range ($x \in [0, 1]$) under consideration. Therefore, the model finds it difficult to obtain accurate estimates of the target value particularly at the edges of the range of values. This can also be seen from the earlier plots where the variance in the model output is particularly high at the edges of the range (Fig 1 and 2)

3 Bivariate Data

3.1 Plot 3 : Plot of root mean squared error (E_{RMS}) on training, validation and test data for different model complexities and λ values

Observations

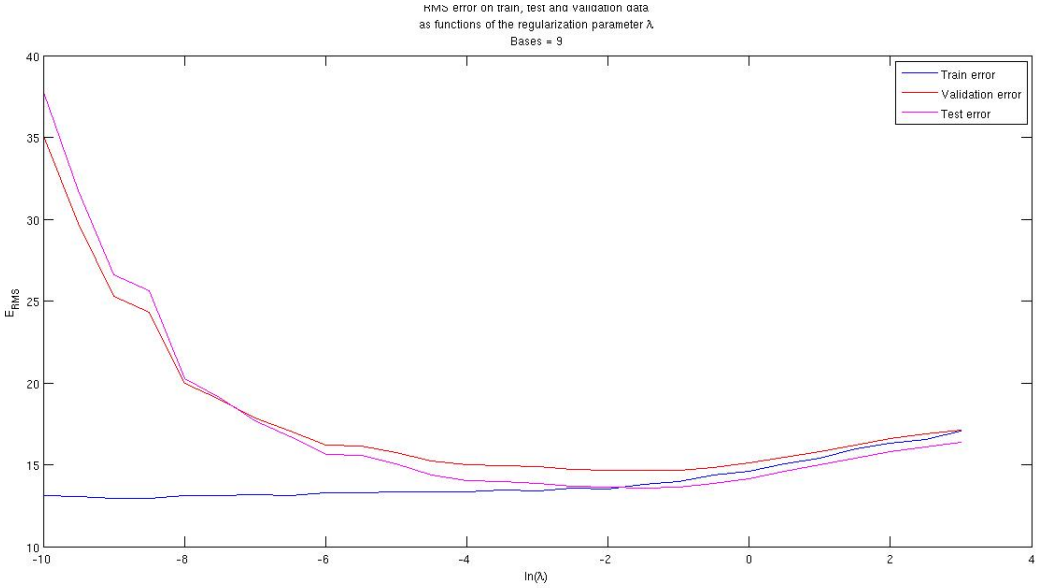
- For a given number of basis functions, the root mean square error on the test and validation data decreases with increase in the regularization parameter. With bases = 9, the test and validation error achieve a minimum value at $\approx \ln(\lambda) = -2$.
- Beyond $\ln(\lambda) = -2$, the test and validation error increase once again.
- There is a gradual increase in the error on the training data as the value of λ is increased.
- With a fixed $\ln(\lambda) = -5$, increasing the number of basis functions causes the error on the test and validation data to decrease initially until a minimum is reached when the number of basis functions is approximately 5.
- Beyond this point, the test and validation error begin a gradual ascent.
- The train error decreases with increasing number of basis functions.

Inferences

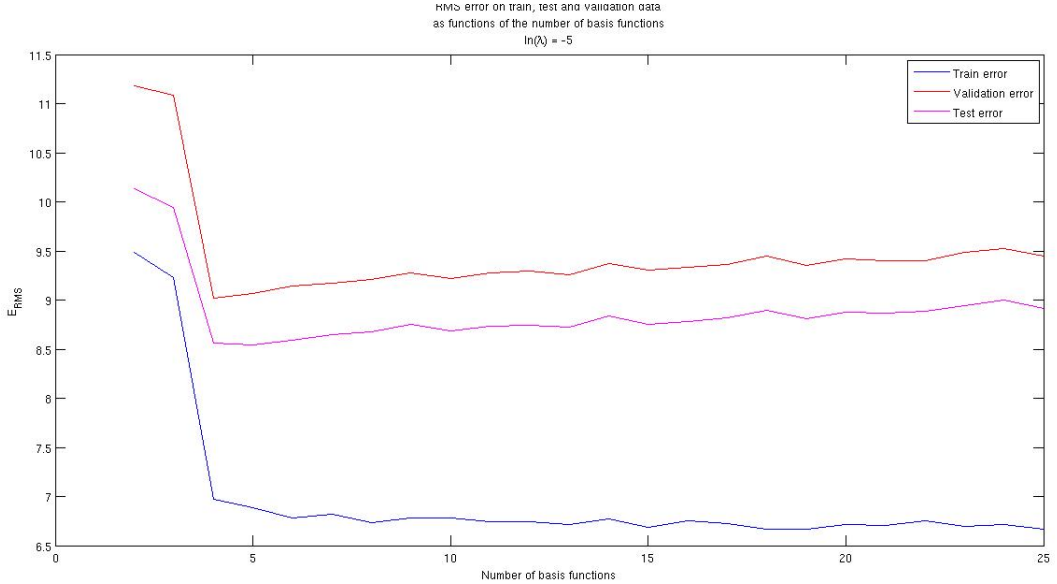
- The point $\ln(\lambda) = 2$ represents an optimal trade-off between bias and variance where the generalization ability of the model over unknown data is ideal.
- With increasing λ, over-regularization compromises the ability of the model to explain the data. This also explains the gradual rise in the error on the train data.
- Similarly, the error minimum when number of basis functions is 5 corresponds to an ideal balance between bias and variance of the model.
- Increasing the number of basis functions keeping the number of training examples, λ and other parameters constant causes the model to overfit the data. Thus, the training error keeps decreasing while the test and validation error keep increasing.

Figure 7

(a) E_{RMS} vs $\ln(\lambda)$ for number of bases $= 9$



(b) E_{RMS} vs number of bases for $\ln(\lambda) = -5$



3.2 Plot 4 : Plot of model and target outputs for the training, validation and test data

Observation

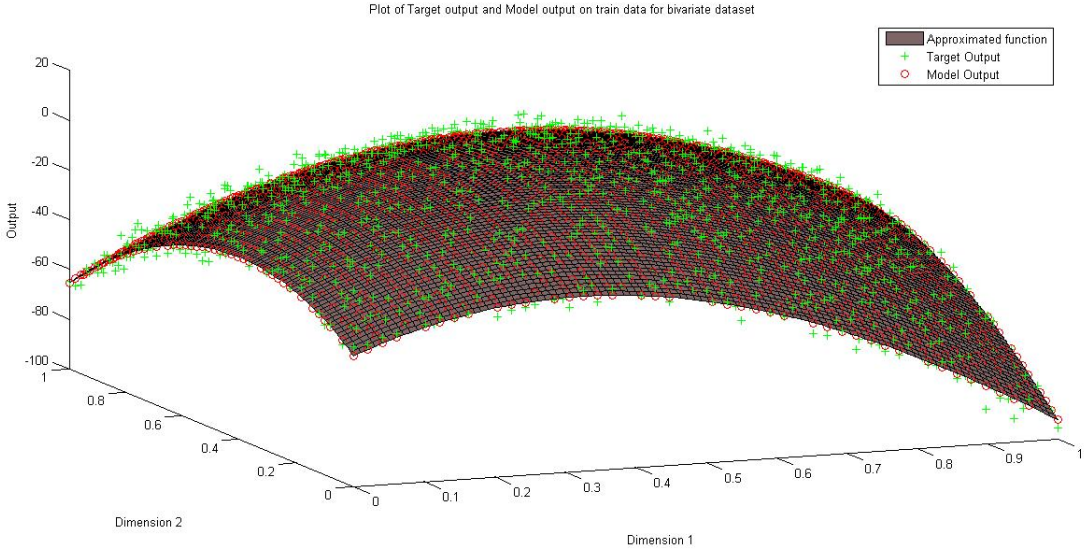
- The target outputs (green) seem to indicate that the underlying function to be approximated is a Gaussian with random noise added. As such the use of Gaussian radial basis functions seems to give a reasonably good fit to the data except under certain parameter configurations (see Fig 9 and 10).
- It can also be observed that increasing the variance hardly has any effect on the error once a minimum threshold has been crossed.

Inference

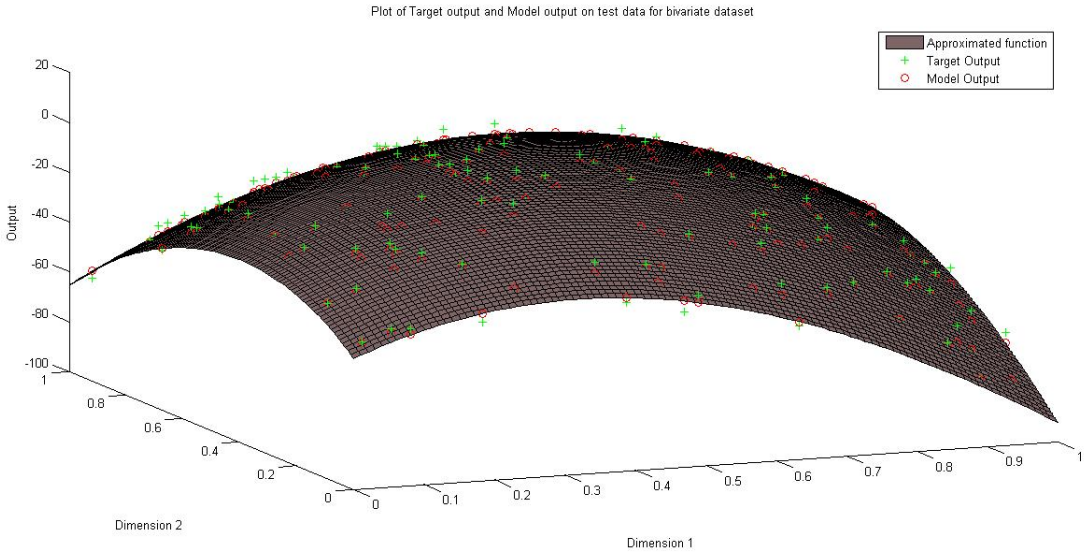
- Since the target values come from a bell-shaped curve over the feature vector space, the underlying function to be approximated does not have many complex topological features that may be missed by the model if there is lack of sufficient data. In essence, the simplicity of the model makes up for some data insufficiency in the sense that the data is reducible to its basic statistics - the mean and variance. If we have sufficient data to estimate the mean well enough, we can vary the empirical variance parameter to get a good enough fit to the data. Thus, overfitting to the training examples can in general be avoided relatively easily.
- The simplicity of the model also allows the weight vectors to adjust easily in case the variance is increased. Therefore, increasing the variance does not really change the error much.
- At very low variance, one can observe the different Gaussians which are fit to the data points, where centroid corresponds to the means identified by the k-means algorithm.

Figure 8: Model Output vs Target Output

(a)



(b)



(c)

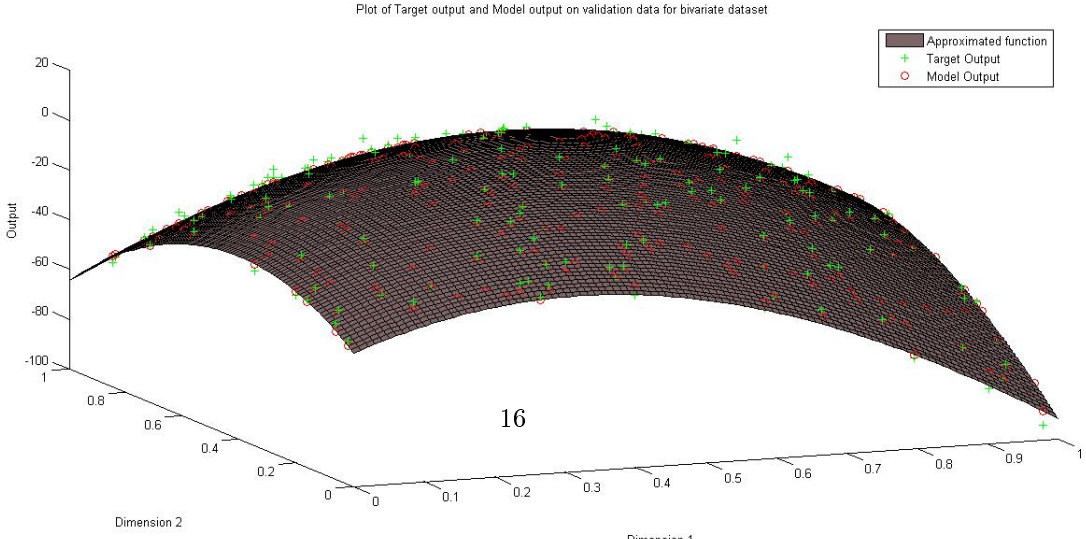


Figure 9: Model Output vs Target Output for Very Small Variance

(a)

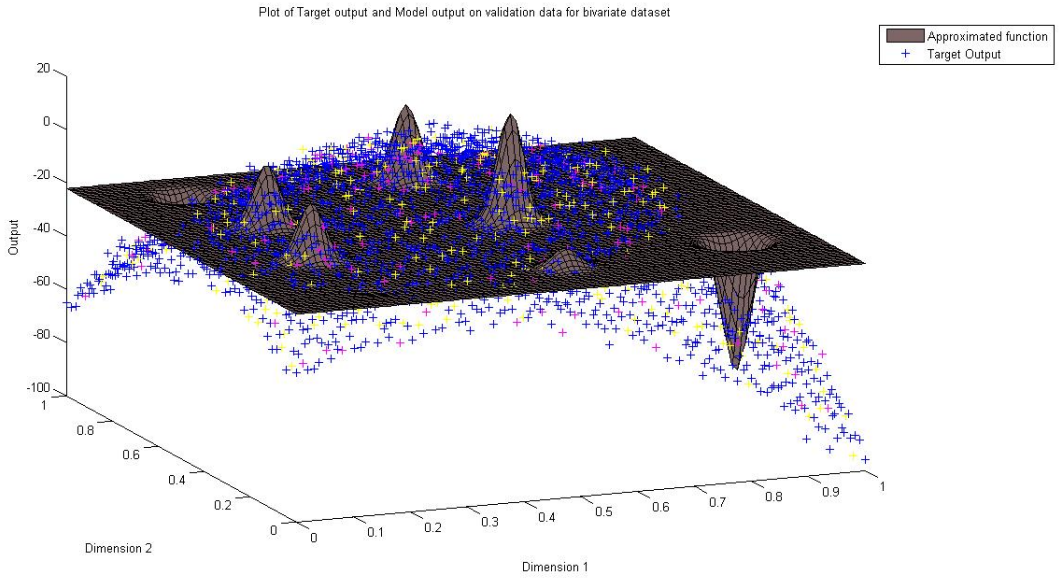
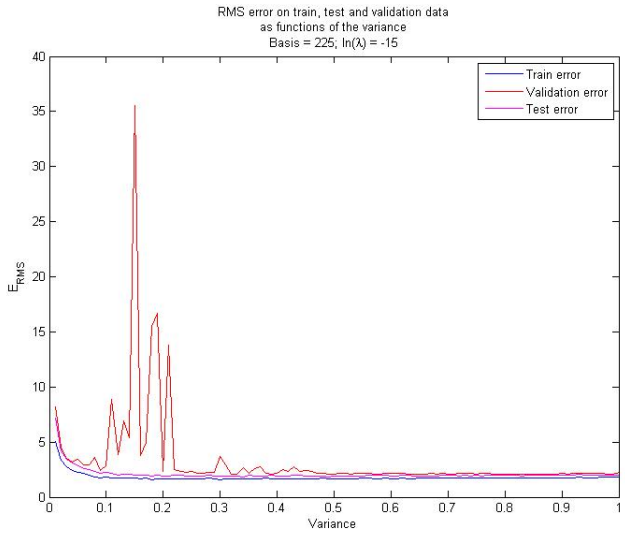
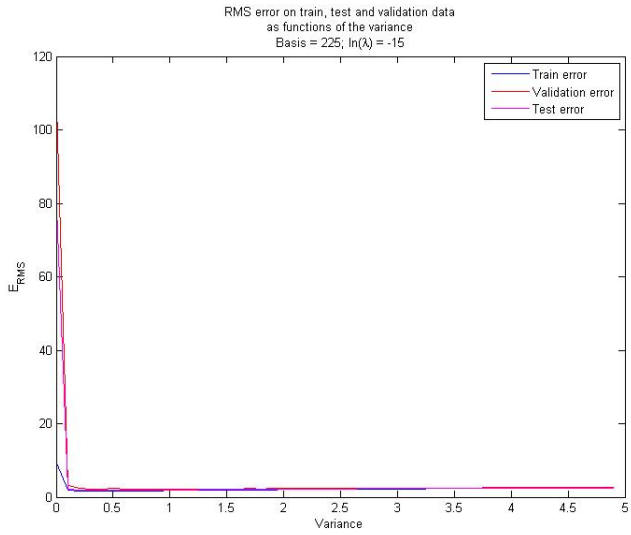


Figure 10: E_{RMS} vs Variance

(a)

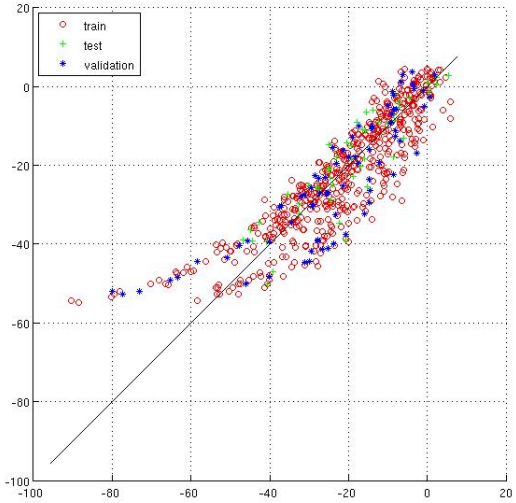


(b)



3.3 Plot 5 : Scatter plot of target output vs model output for training, validation and test data

Figure 11: Number of basis functions = 25, $\ln(\lambda) = -5$, variance = 0.1



Observations

- Like the univariate case, the output of our model seems to give a good enough correlation to the target output with most of the points lying in a small bounded region about the line $y = x$.
- Again the correlation worsens at the edges of the graph.

Inferences

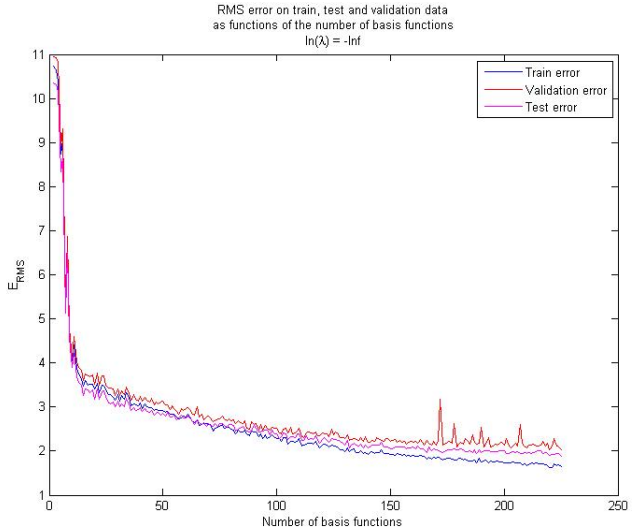
- The above results are very similar to those obtained for the univariate data case.
- As was evident earlier as well, Gaussian basis functions approximate the data very well as the function to be approximated itself seems to be a Gaussian.
- The reason for degradation of model performance are the same as for the univariate case - sparsity of the data at the edges and finite range of function to be approximated.

4 Multivariate Data

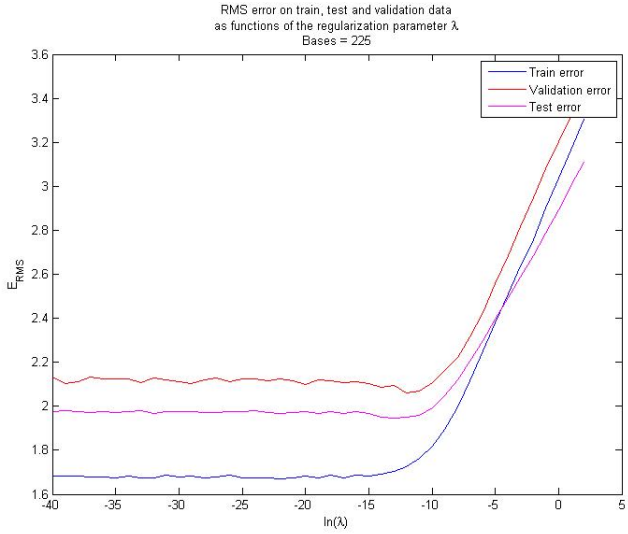
4.1 Plot 3 : Plot of root mean squared error (E_{RMS}) on training, validation and test data for different model complexities and λ values

Figure 12

(a) E_{RMS} vs number of bases for $\ln(\lambda) = -Inf$



(b) E_{RMS} vs $\ln(\lambda)$ for number of bases = 225



Observations

- While increasing the number of basis functions causes the error on the test and validation data to decrease initially.
- Beyond certain point the train error goes lower but the test and validation error starts increasing.

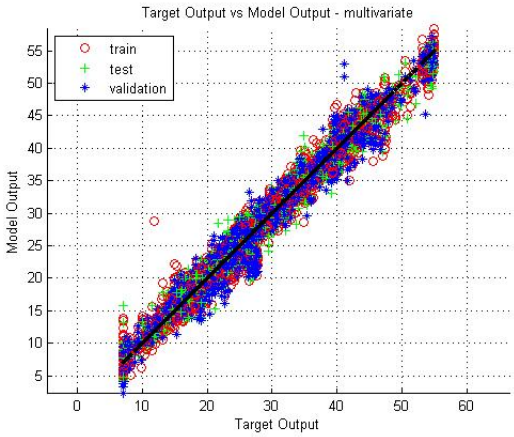
- For a given number of basis functions, the root mean square error on the test and validation data decreases with increase in the regularization parameter. With bases = 225, the test and validation error achieve a minimum value at $\approx \ln(\lambda) = -12$.
- Beyond $\ln(\lambda) = -12$, the test and validation error increase once again.
- There is a gradual decrease in the error on the testing and validation data for $\ln(\lambda) < -12$.

Inferences

- By Increasing the number of basis functions keeping the number of training examples, and other parameters constant causes the model to overfit the data. Since the number of data is very high we observe a gradual increase in testing and validation error.
- Since the degree polynomial is high we observe very gradual decrease in root mean squared error as the regularization parameter is increased.

4.2 Plot 5 : Scatter plot of target output vs model output for training, validation and test data

Figure 13: Number of basis functions = 225, $\ln(\lambda) = -15$, variance = 0.5



Observations

- Like the other cases the output of our model seems to give a good enough correlation to the target output with most of the points lying in a small bounded region about the line $y = x$.
- There are few outlier points which are not properly fit by our regression model.

Inferences

- The above results are very similar to those obtained for the univariate data case.

References

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning*