Indian Institute of Technology, Madras
CS5691: Pattern recognition and Machine Learning
PRML Assignment-I Report
Group - 22

Ashutosh Kakadiya - CS18S013,
Divyanshu Kakwani - CS18S005,
Nikhil Baba - CS15B008

September 2018

Contents

1 PCA **2**
- 1.1 Eigenanalysis . . . 2
- 1.2 Results . . . 2
- 1.3 Conclusion . . . 6

2 Linear Regression Models **7**
- 2.1 Overview . . . 7
- 2.2 Models . . . 7
  - 2.2.1 Model 1: Polynomial Curve Fitting . . . 7
  - 2.2.2 Model 2: Linear model with Polynomial Basis . . . 8
  - 2.2.3 Model 3: Linear model with Gaussian Basis . . . 8
- 2.3 Results on Dataset 1 . . . 8
  - 2.3.1 Model 1 . . . 8
- 2.4 Results on Dataset 2 . . . 14
  - 2.4.1 Model 2 . . . 14
  - 2.4.2 Model 3 . . . 19
- 2.5 Results on Dataset 3 . . . 22
  - 2.5.1 Model 2 . . . 22
  - 2.5.2 Model 3 . . . 23
- 2.6 Conclusion . . . 24

Chapter 1

PCA

1.1 Eigenanalysis

Dimensionality Reduction by Principal Component Analysis:

The task is to identify the important dimensionalities of a 64 x 64 image and reduce it from 4096 to something significantly lower, say 10 or so. This is done by eliminating all the less informative eigen values and its corresponding vectors.

The training set for this task is a set of 150 gray-scale images of a person's face in various angles. Each image is in a 64 x 64 pixel representation. Each image is captured and converted into a row vector of dimension 1 x 4096 and all the images are captured into a 150 x 4096 matrix.

Co-variance matrix **C** is computed from the above matrix, after it is normalized.It becomes easier to find co-variance matrix of normalized data.

Principal component matrix(**V**) is computed in the order of significant eigen values and following formula is applied.

Reduced Image = Original Image x $\mathbf{V}_{4096*k}$ **x** $\mathbf{V}^T_{k*4096}$
where k is number of significant eigen values.

1.2 Results

For the results, we put up a set of 5 images using **N** significant eigen values, The comparison is shown below,

Original Image	k = 10	k = 20	k = 40

Table 1.1: PCA Reconstruction for k = 10, 20, 40

Original Image **k = 80** **k = 160** **k = 320**

Table 1.2: PCA Reconstruction for k = 80, 160, 320

Original Image **k = 640**

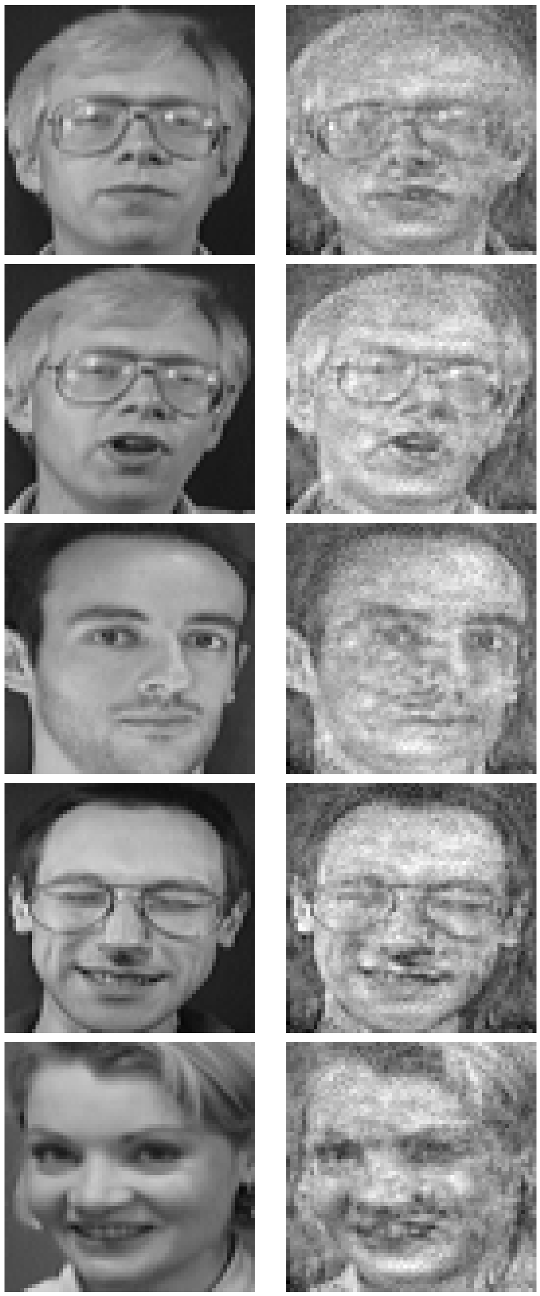

Table 1.3: PCA Reconstruction for k = 640

1.3 Conclusion

As we can observe, the amount of detail that keeps increasing by the number of significant eigen vectors(significant features) is differential. All the important features are covered in the top 1-10 percent of the eigen vectors in most of the cases. This shows that PCA is very much doing it's task and data reduction is very efficient.

Chapter 2

Linear Regression Models

2.1 Overview

The broad perspective of what we do to design a predicting model with large amount of data is as follows:

1.Divide the data in the hand into training, validation and testing data.

2.Create models with varying hyper parameters using the training data.

3.Find the best model out of these by finding the model with least error
on the validation data.

4.Finally, test the accuracy of your model on the test data.

The difference between many procedures lie in the way of training and method of choosing hyper parameters.

2.2 Models

2.2.1 Model 1: Polynomial Curve Fitting

The approximated function has the following representation:

$y(x, \mathbf{w}) = \Sigma_{i=0}^{M-1} w_i \Phi_i(x)$, where $\Phi_i(x)$ is polynomial basis function set to x^i for$i = 1, 2, 3, ..., P$. Here, P is maximum degree of given polynomial function.

Where object is to find optimal values of $\bar{w^*}$ with respect to minimizing sum of squared error(SSE) function with regularization:
$\varepsilon(\bar{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, \bar{w}) - t_n)^2 + \frac{\lambda}{2} \sum_{i=1}^{M} w_i^2$

2.2.2 Model 2: Linear model with Polynomial Basis

The approximated function has the following representation:

$y(x, \mathbf{w}) = \Sigma_{i=0}^{M-1} w_i \Phi_i(x)$, where $\Phi_i(x)$ is polynomial basis function set to x^i for $i = 1, 2, 3, ..., P$. Here, P is maximum degree of given polynomial function.But here i will become (i_1, i_2) because of bivariate data.
For, example if $P = 2$ then, $y(x, w) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2$
Where object is to find optimal values of $\bar{w^*}$ with respect to minimizing sum of squared error(SSE) function with regularization:
$\varepsilon(\bar{w}) = \dfrac{1}{2} \sum_{n=1}^{N} (y(x_n, \bar{w}) - t_n)^2 + \dfrac{\lambda}{2} \sum_{i=1}^{M} w_i^2$

2.2.3 Model 3: Linear model with Gaussian Basis

The approximated function has the following representation:

$y(x, \mathbf{w}) = \Sigma_{i=0}^{M-1} w_i \Phi_i(x)$, where $\Phi_i(x) = e^{-(x-\mu_i)^2/2\sigma^2}$
Here μ andσ are pre-set parameters. To find the values of μ, first we apply K-means clustering on given training dataset with gaussian basis function. Value of σ will be assigned arbitrarily. The minimization(error) function with regularization is :
$\varepsilon(\bar{w}) = \dfrac{1}{2} \sum_{n=1}^{N} (y(x_n, \bar{w}) - t_n)^2 + \dfrac{\lambda}{2} \bar{w}^t \tilde{\Phi} \bar{w}$, where $\tilde{\phi_{ij}} = e^{-\frac{||\bar{\mu}_i - \bar{\mu}_j||^2}{2\sigma^2}}$

2.3 Results on Dataset 1

2.3.1 Model 1

Plots

Here are the different plots of the approximated polynomial functions obtained using univariate training datasets of different sizes, for different model complexities and for different values of λ.

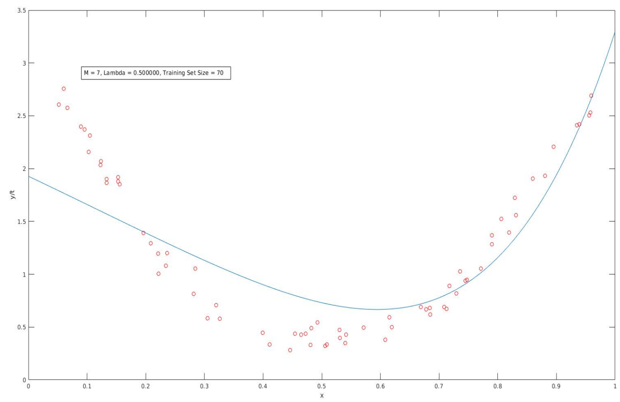

Figure 2.1: $M = 1, \lambda = 0.5, \text{Training set size} = 70$

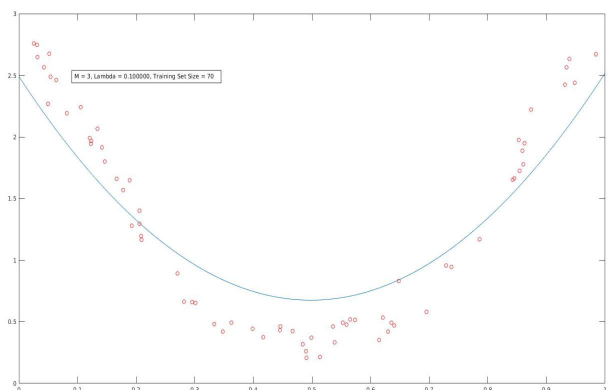

Figure 2.2: $M = 3, \lambda = 0.1, \text{Training set size} = 70$

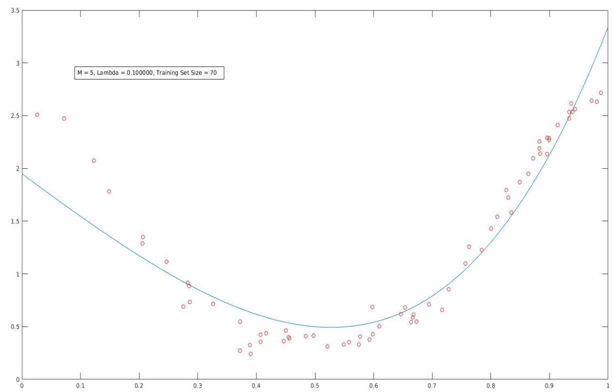

Figure 2.3: $M = 5, \lambda = 0.1, \text{Training set size} = 70$

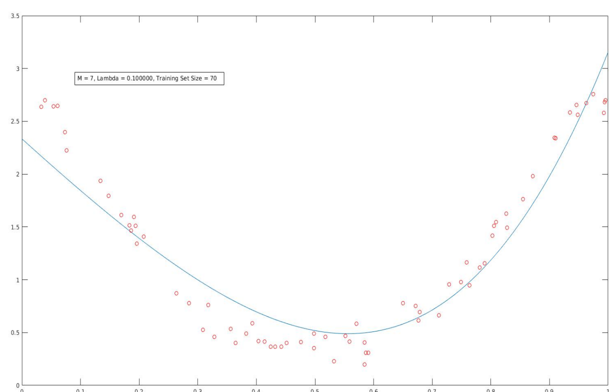

Figure 2.4: $M = 7, \lambda = 0.1, \text{Training set size} = 70$

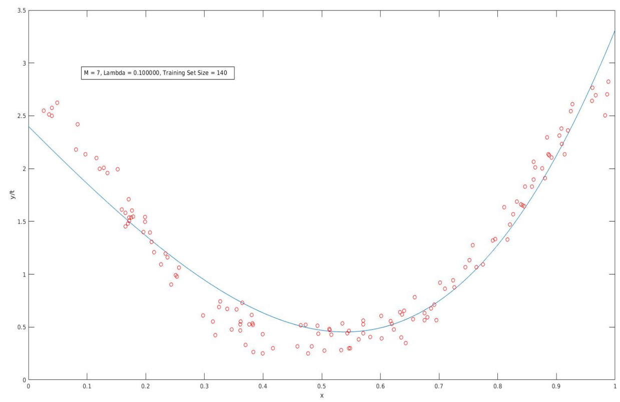

Figure 2.5: $M = 7, \lambda = 0.1, \text{Training set size} = 140$

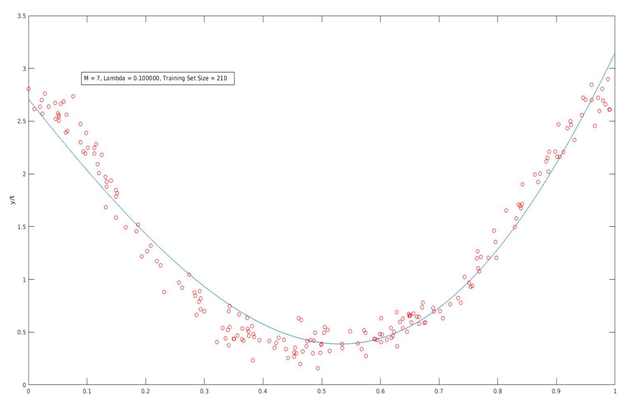

Figure 2.6: $M = 7, \lambda = 0.1, \text{Training set size} = 210$

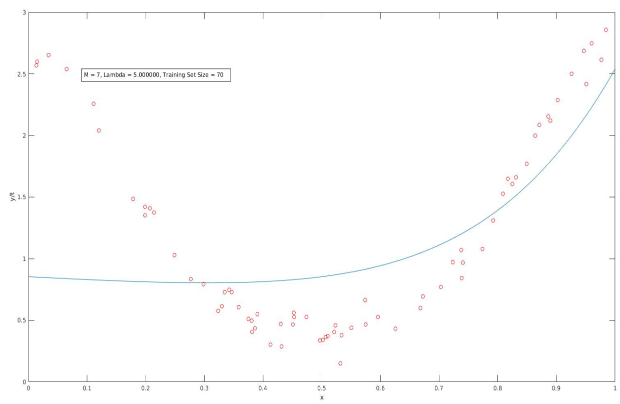

Figure 2.7: $M = 7, \lambda = 1.00, \text{Training set size} = 70$

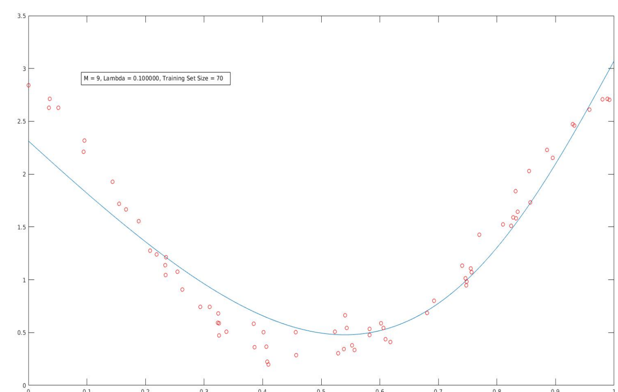

Figure 2.8: $M = 7, \lambda = 1.00, \text{Training set size} = 70$

Validation RMS Errors

Train Size	Model Complexity	Regularizer	Reg. Coefficient	RMS Error
70	7	Quadratic	0.5	0.2880
70	3	Quadratic	0.1	0.2676
70	5	Quadratic	0.1	0.2772
70	7	Quadratic	0.1	0.2153
140	7	Quadratic	0.1	0.1489
210	7	Quadratic	0.1	0.1771
210	7	Quadratic	0.0	0.0656

Table 2.1: Validation RMS Errors

Optimal Hyperparameters

Training Size = 210, lambda = 0.0, and M = 7 lead to the model with the least validation error of 0.0656

Train and Test RMS Error of the model

Test $E_{rms} = 0.0854$
Train $E_{rms} = 0.0972$

Scatter Plots

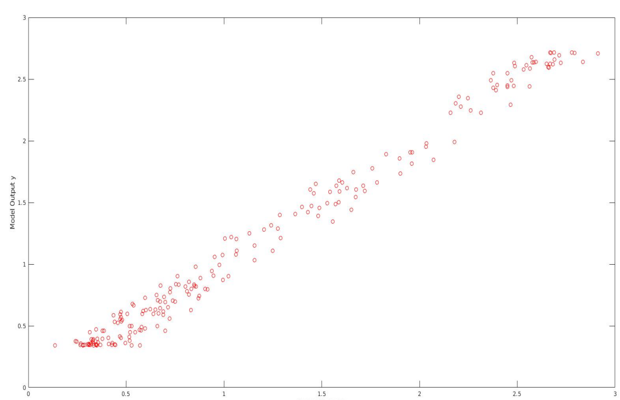

Figure 2.9: Target vs predicted scatter plot on training set

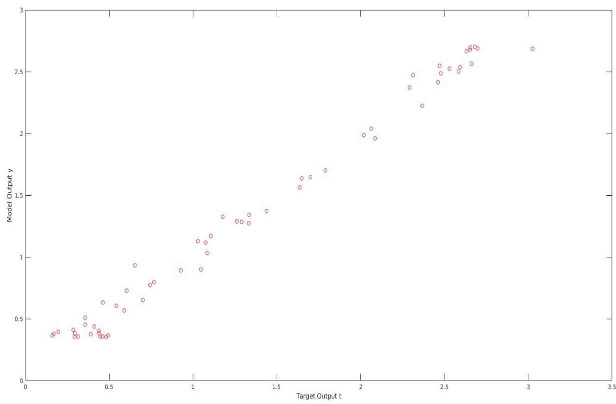

Figure 2.10: Target vs predicted scatter plot on test set

2.4 Results on Dataset 2

2.4.1 Model 2

Plots

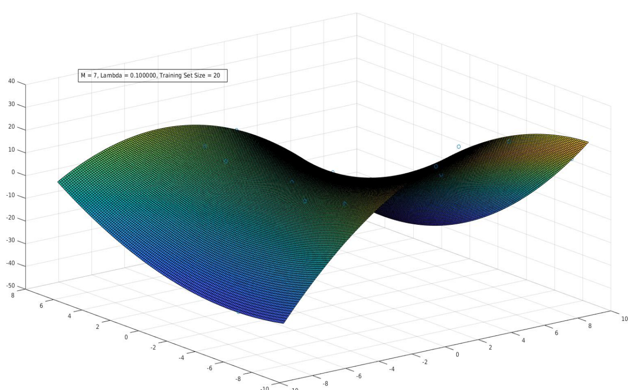

Figure 2.11: $M = 7, \lambda = 0.1$, Training set size $= 20$

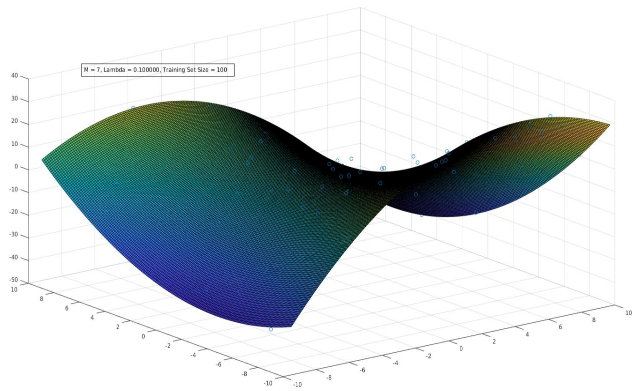

Figure 2.12: $M = 7, \lambda = 0.1, \text{Training set size} = 100$

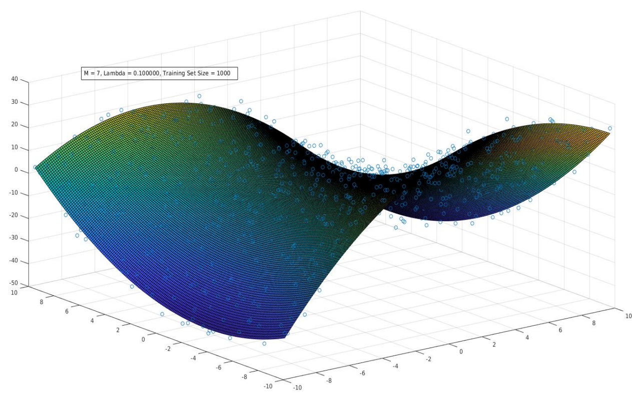

Figure 2.13: $M = 7, \lambda = 0.1, \text{Training set size} = 1000$

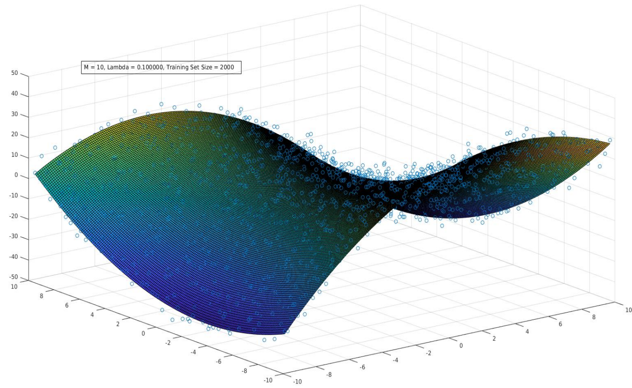

Figure 2.14: $M = 10, \lambda = 0.1$, Training set size $= 2000$

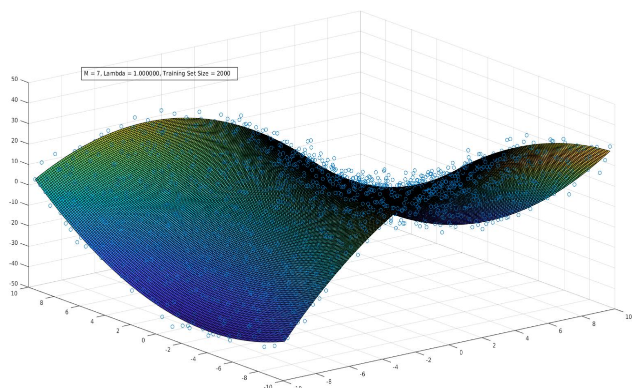

Figure 2.15: $M = 7, \lambda = 1.0$, Training set size $= 2000$

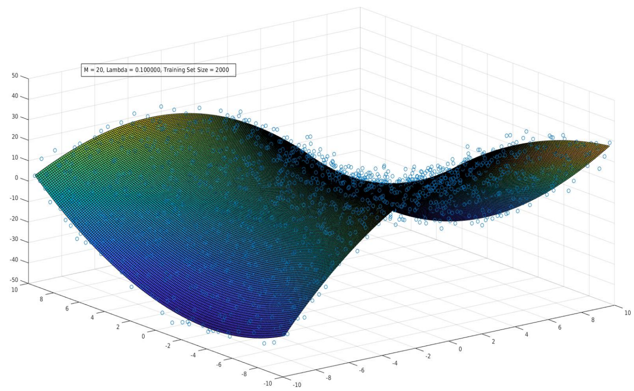

Figure 2.16: $M = 20, \lambda = 0.1$, Training set size = 2000

Validation RMS Errors

Train Size	Model Complexity	Regularizer	Reg. Coefficient	RMS Error
20	7	Quadratic	0.1	3.9847
2000	20	Quadratic	0.1	3.7177
100	7	Quadratic	0.1	3.8176
1000	7	Quadratic	0.1	3.7155
2000	10	Quadratic	0.1	3.6865
2000	7	Quadratic	1.0	3.6873
2000	20	Quadratic	0.1	3.7177

Table 2.2: Validation RMS Errors

Optimal Hyperparameters

Training Size = 2000, lambda = 1.0, and M = 7 lead to the model with the least validation error of 3.6873

Train and Test RMS Error of the model

Test $E_{rms} = 3.4748$
Train $E_{rms} = 3.2929$

Scatter Plots

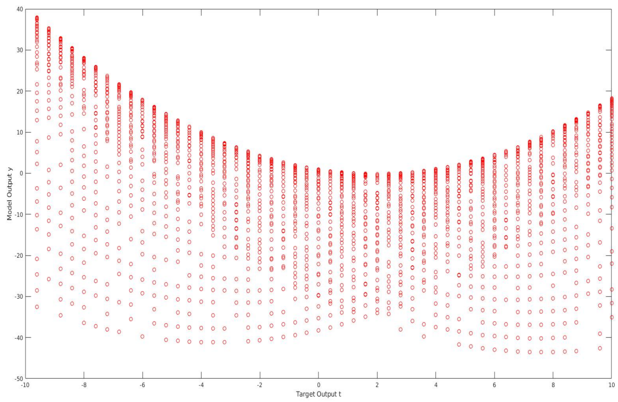

Figure 2.17: Target vs predicted scatter plot on training set

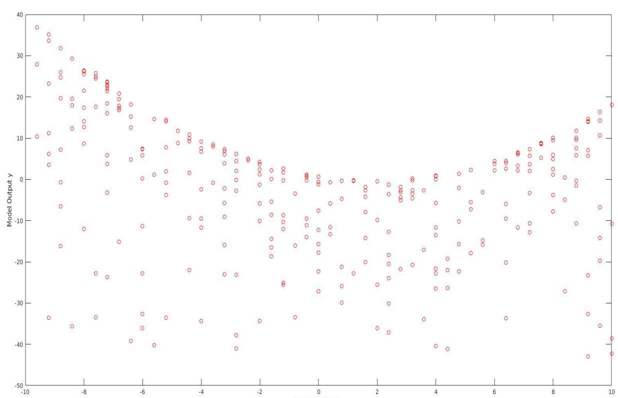

Figure 2.18: Target vs predicted scatter plot on test set

2.4.2 Model 3

Plots

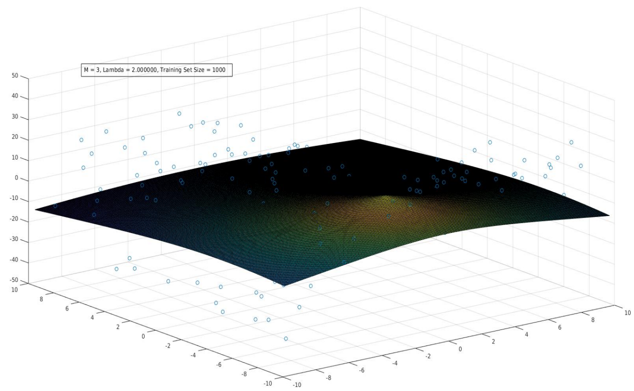

Figure 2.19: $M = 3, \lambda = 2$, Training set size $= 1000$

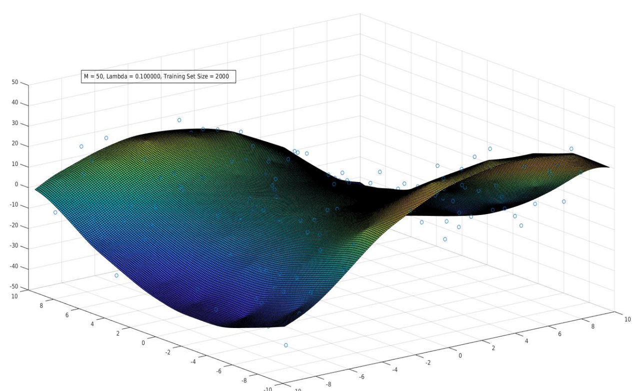

Figure 2.20: $M = 50, \lambda = 0.1$, Training set size $= 2000$

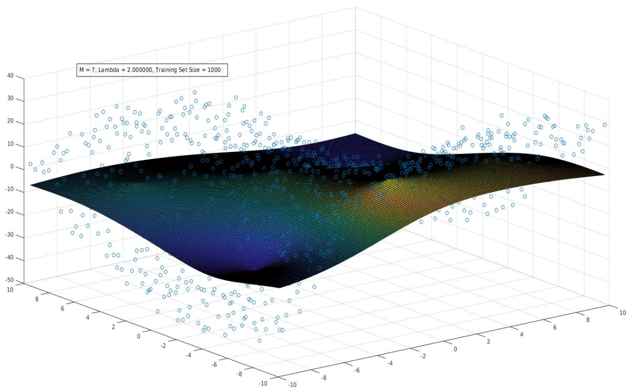

Figure 2.21: $M = 7, \lambda = 2, \text{Training set size} = 1000$

Validation RMS Errors

Train Size	Model Complexity	Regularizer	Reg. Coefficient	RMS Error
1000	20	Tikhonov	0.1	4.9183
1000	15	Tikhonov	0.1	5.5222
1000	50	Tikhonov	0.0	4.0556
1000	100	Quadratic	0.0	3.7345
2000	50	Quadratic	0.1	4.1558

Table 2.3: Validation RMS Errors

Optimal Hyperparameters

Training Size = 1000, lambda = 0.0, and M = 100 lead to the model with the least validation error of 3.7345

Test and Train RMS Error of the model

Test $E_{rms} = 5.5674$
Train $E_{rms} = 3.6038$

Scatter Plots

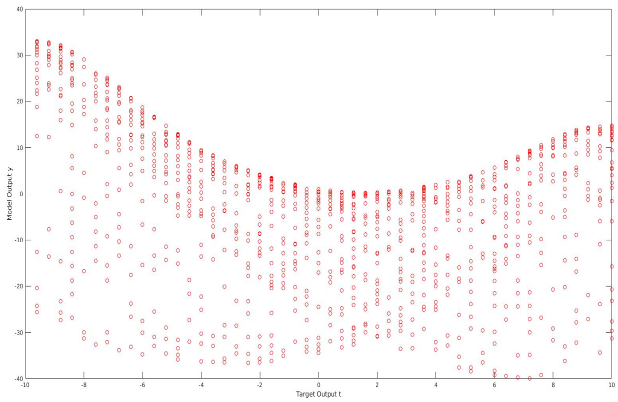

Figure 2.22: Target vs predicted scatter plot on training set

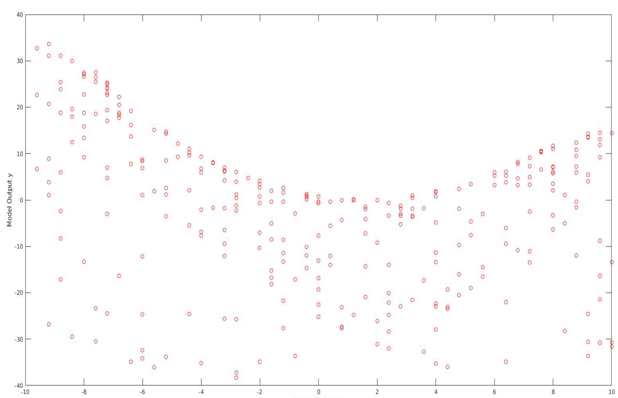

Figure 2.23: Target vs predicted scatter plot on test set

2.5 Results on Dataset 3

2.5.1 Model 2

Validation RMS Errors

Train Size	Model Complexity	Regularizer	Reg. Coefficient	RMS Error
721	10	Quadratic	1	9.4435
721	5	Quadratic	0.1	12.2426
721	7	Quadratic	0.0	13.2950
721	30	Quadratic	5	9.8378
721	40	Quadratic	0.1	8.3934
721	60	Quadratic	0.2	6.0645

Table 2.4: Validation RMS Errors

Optimal Hyperparameters

Training Size = 721, lambda = 0.5, and M = 60 lead to the model with the least validation error of 6.0645

Test and Train RMS Error of the model

Test $E_{rms} = 7.1788$
Train $E_{rms} = 6.9197$

Scatter Plots

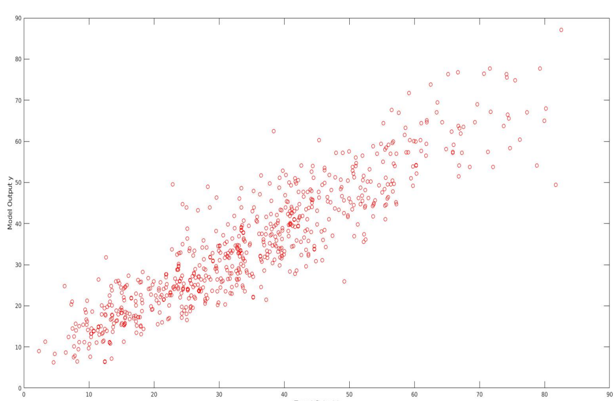

Figure 2.24: Target vs predicted scatter plot on training set

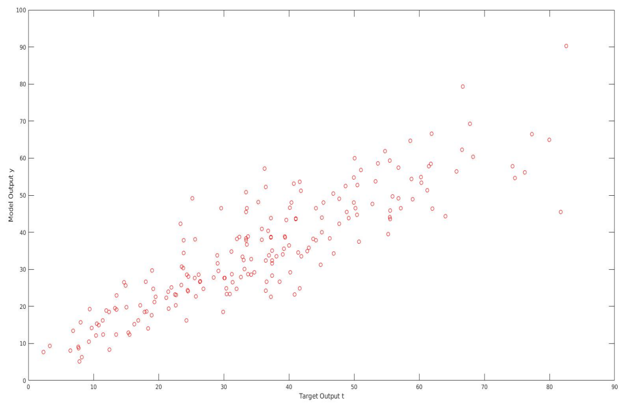

Figure 2.25: Target vs predicted scatter plot on test set

2.5.2 Model 3

Validation RMS Errors

Train Size	Model Complexity	Regularizer	Reg. Coefficient	RMS Error
721	30	Quadratic	1	12.8942
721	40	Tikhonov	2	14.1106
721	100	Quadratic	0.1	11.5520
721	70	Tikhonov	1	13.5096
721	50	Quadratic	5	14.6333

Table 2.5: Validation RMS Errors

Optimal Hyperparameters

Training Size = 721, lambda = 0.1, and M = 100 lead to the model with the least validation error of 11.5520

Test and Train RMS Error of the model

Test E_{rms} = 12.2834
Train E_{rms} = 10.3887

Scatter Plots

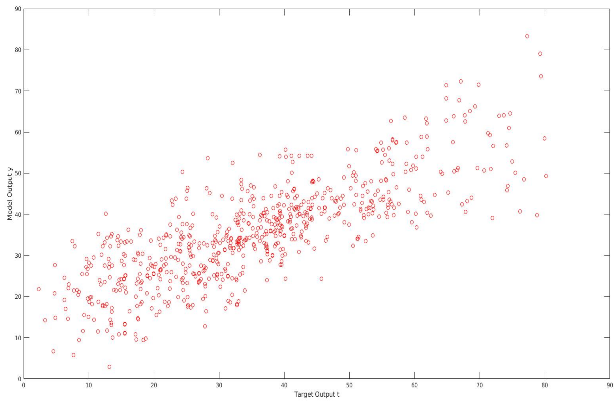

Figure 2.26: Target vs predicted scatter plot on training set

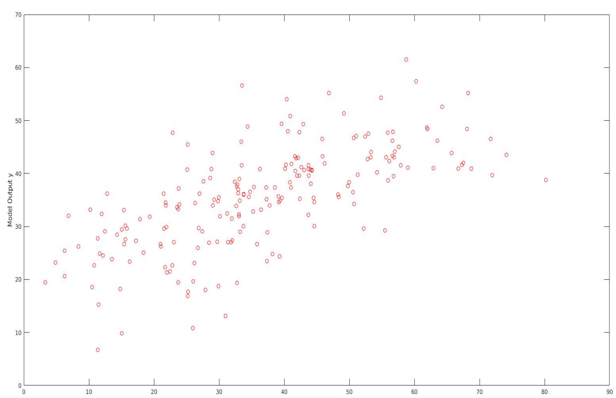

Figure 2.27: Target vs predicted scatter plot on test set

2.6 Conclusion

- Based on above experiments and results are shown, polynomial basis regression is performing well than gaussian basis function on both bivariate and multivariate dataset.
- Time complexity of gaussian basis function is higher than polynomial basis function regression.

- Models with regularization terms are less complex than simple regression models, because regularization term avoid to learn more complex tasks so it prevents overfitting on training dataset.
- Sometimes, After certain model complexity, and value of λ with increasing complexity there is no significant change in results!