Regression

Contents

List of Figures

1 Introduction

Regression is the process of fitting a curve for a given set of data points **x, t** [1] and then predicting the value **y(x)** for a new **x** where **t** is the output of given data point **x** and **y(x)** is the function learnt by regression to predict value of unknown **x**. Broadly, it does this by finding **y(x)** by minimizing a cost function L(**y(x), t**)). In any regression task we assume a set of basis functions ψ_i and fit a curve of nature below,

$$a_0 + \sum_{i=1}^{m} a_i \cdot \psi_i$$

where a_i are parameters that need to be estimated. Basis functions can be polynomial, sigmoidal, fourier etc. depending on the type of regression problem we are solving.

2 Experimentation

We have assumed Polynomial Basis functions of form,

$$a_0 + \sum_{i=1}^{m} a_i \cdot x^i$$

where x is the feature and a are the coefficients. Above equation uses only one feature to generate the polynomial. For more than one feature we can assume all feature interaction terms to construct the polynomial,

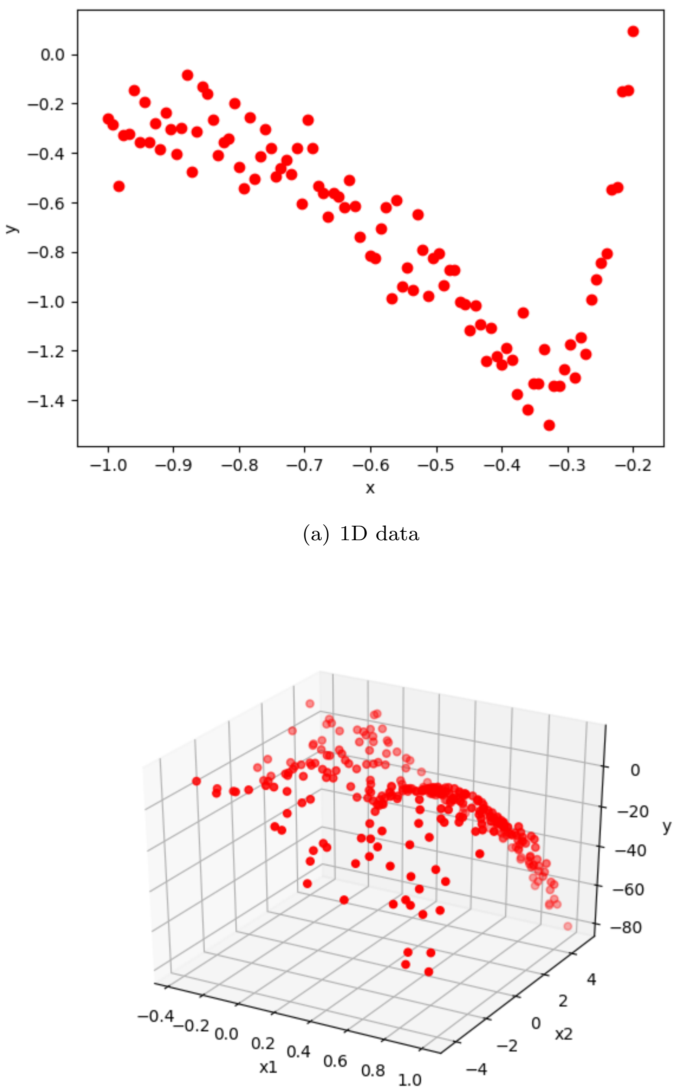
$$a_0 + \sum_{j=1}^{k} \sum_{b_1 + b_2 \ldots b_n = j} a_{ji} \cdot x_1^{b_1} \cdot x_2^{b_2} \cdots x_n^{b_n}$$

where k corresponds to maximum degree of polynomial, b_i is the degree of i^{th} feature of a monomial.

2.1 Data Source

All data used for experimentation are from q19_1.txt, q19_2.txt, 19_3.txt which comprises of 1D, 2D and multi-dimensional data points. Each feature corresponds to a dimension. Plots of the entire data set is shown below for 1D and 2D.

[1] All vectors in the report are in bold



(a) 1D data

(b) 2D data

Figure 1: Scatter Plot of 1D and 2D data

2.2 1D Curve Fitting

2.2.1 Curve Fitting

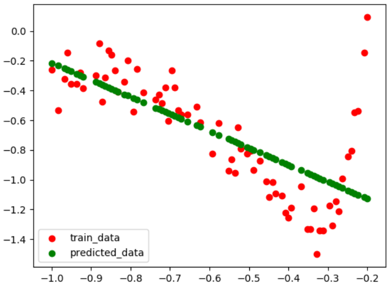
In the experiment below we are using sum of squared error averaged over number of data points where N is number of data points.

$$\frac{1}{N} \cdot \sum_{n=1}^{N} (y(\mathbf{x}) - t)^2$$

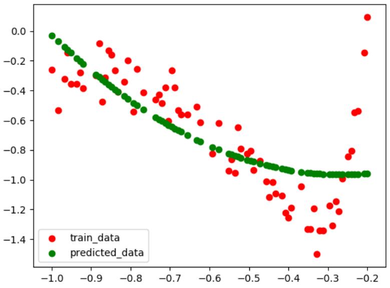
Training data of 1D data shown in scatter plots above was fit using polynomials of nature,

$$a_0 + \sum_{i=1}^{m} a_i \cdot x^i$$

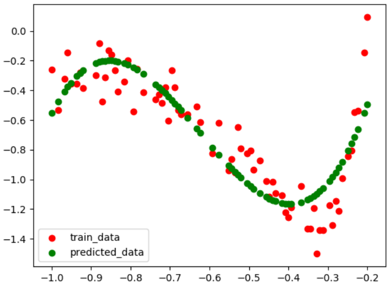
with m varying from 1 to 6. Training ratio of 0.7, validation ratio of 0.2 and testing ratio of 0.1 is used. A separate validation data is set aside for model selection so that model selected is not biased towards testing data.



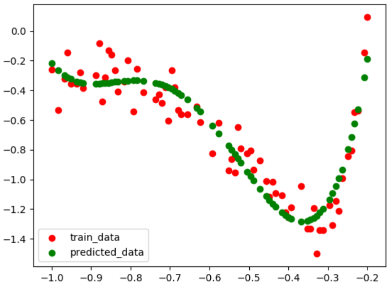
(a) m=1



(b) m=2

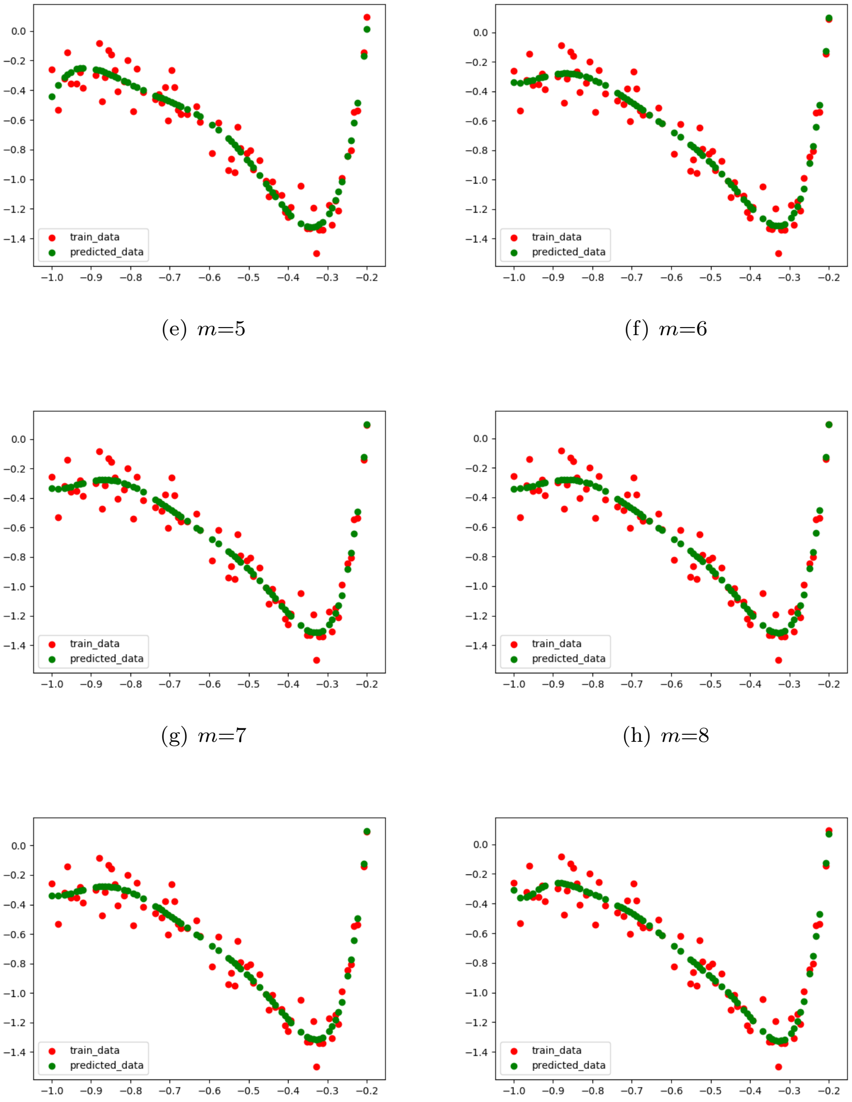


(c) m=3



(d) m=4

From $m = 6$ onwards one can observe plots are not changing much and polynomial is trying to fit noise too.



(e) m=5

(f) m=6

(g) m=7

(h) m=8

(i) m=9

(j) m=10

Figure 1: Scatter Plots for 1D training data

2.2.2 Repeated Cross Validation

k-fold Cross Validation splits data into k parts and assigns (k-1) parts to training and 1 part into validation. This process is repeated k times such that validation and training gets different parts of data set. In experiment, we are giving $0.7 \cdot k$ parts to training, $0.2 \cdot k$ parts to validation and $0.1 \cdot k$ parts to testing. Experiment is repeated k times so that training occurs across entire data set.

Model is trained by using repeated 10-fold cross validation since the data size is small(100) for 1D, thereby requiring training across the entire data set for correct model identification. Cross validation is repeated many times by random shuffling of data to reduce variation in its result.

Cross validation is better than repeated random sampling as the training happens across the entire data set without falling pray to misrepresentation of data by training data which might occur in repeated random sampling.

2.2.3 Model Selection

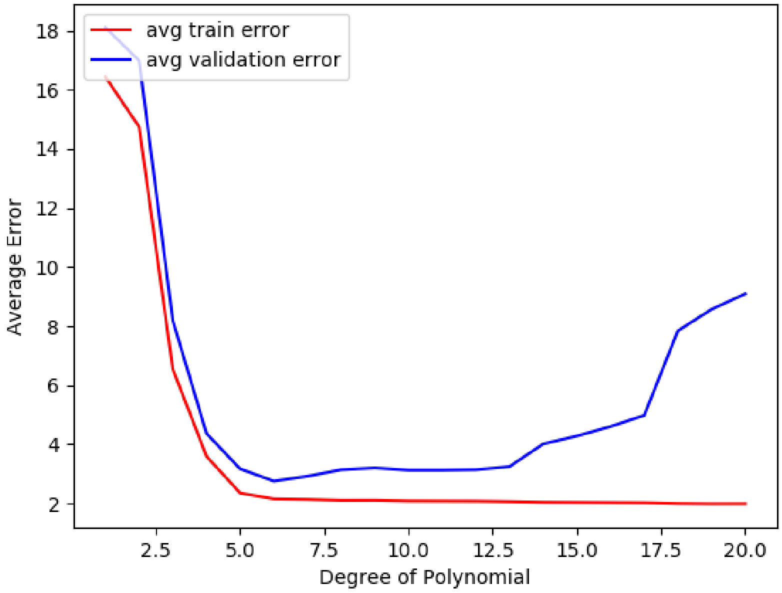


Figure 2: Average Error vs Degree of polynomial

Training error and validation error is high for degree 1 polynomial. This is due to high bias towards a lower degree model. As the model complexity is increased, traning and validation error decreases to minimum at degree 6. Training error decreases as model is more flexible to fit data points. Validation error decreases as model is able to better understand nature of data. But validation error increases after a certain point as model tends to overfit noise thereby failing to generalize to new points. This is called a high variance problem. When the experiment is run multiple times, degree 6 is chosen every time. Therefore, degree 6 polynomial is selected as the model.

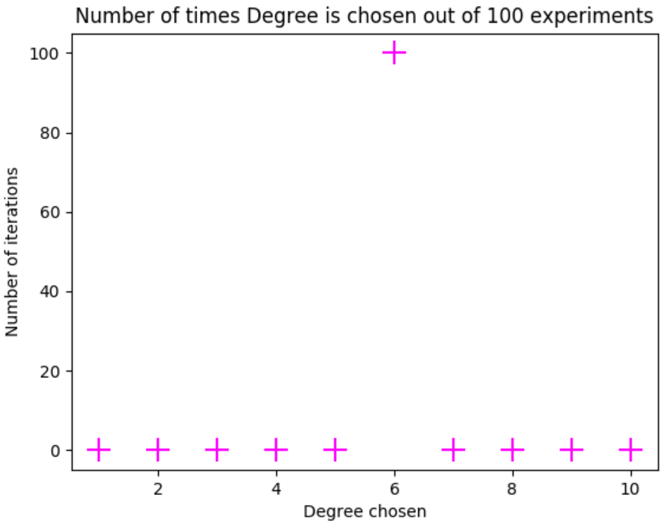
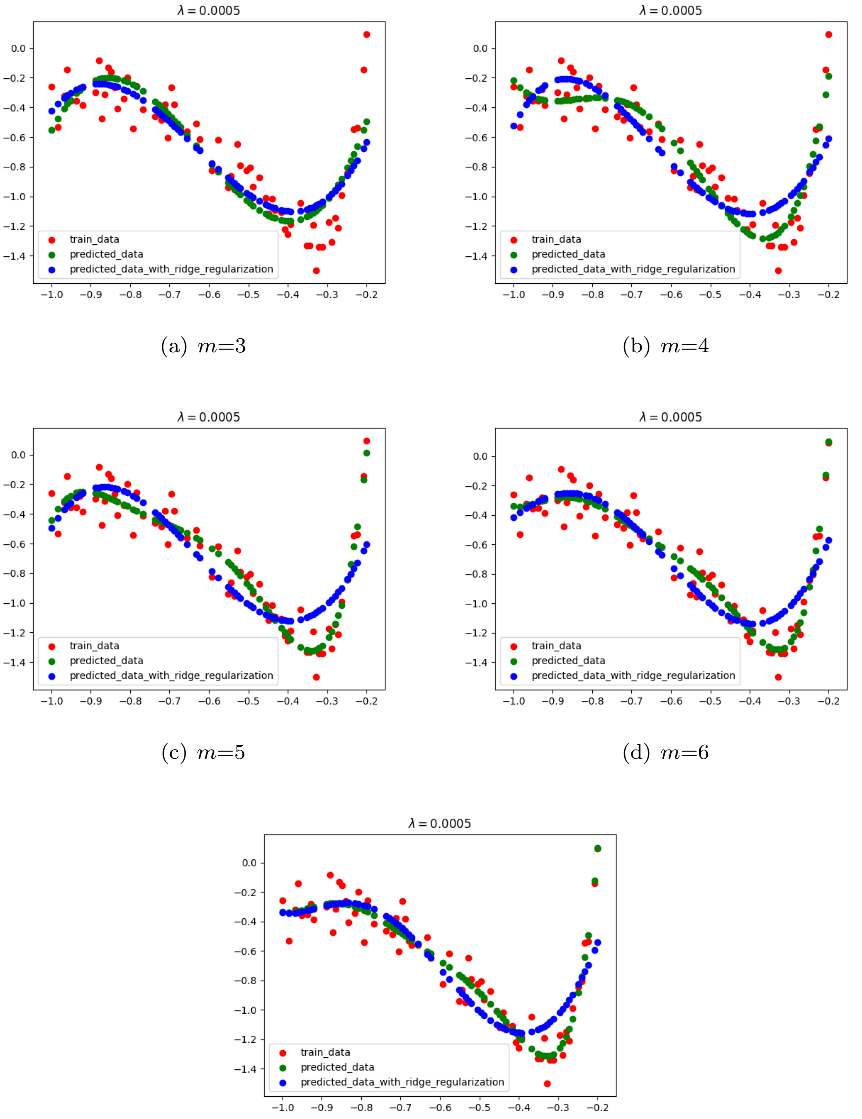


Figure 3: Number of times degree is chosen

2.2.4 Ridge Regression

Goal of ridge regression is to come with smoother models by penalizing coefficients terms, thereby avoiding overfitting problems. Degree of the model selected by ridge can be same or lesser than that of ordinary least squares but it ensures that model is smooth. In the experiment below we are using sum of squared error averaged over number of data points N along with regularizaton as the cost function.

$$\frac{1}{N} \cdot \sum_{n=1}^{N} (\boldsymbol{\theta} \cdot \mathbf{x} - t)^2 + \lambda \cdot \boldsymbol{\theta}^T \boldsymbol{\theta}$$



(a) m=3

(b) m=4

(c) m=5

(d) m=6

(e) m=7

Figure 4: Ridge Scatter Plots for 1D training data

Smooth plots obtained due to ridge regression are shown above. As degree of polynomial is increased ordinary least square model tend to fluctuate to fit the points but ridge regression model tends to remain smooth.

2.2.5 λ Selection

From ordinary least squares we deduced that degree 6 polynomial fits data well. One can fit data using degree 6 with reduced variance and greater smoothness using ridge regression. Varying λ from low to high value and fitting degree 6 model, gives the plot below. When λ is small, training error is small as the model tries to perfectly fit the training data. As λ is increased to large values, magnitude of coefficients are reduced, thereby increasing training error. Validation error also increases with large λ due to same reason. From the plots below, one can infer that value of e^{-20} has to be selected for λ as it is the value at which both validation and training error are low. This can be observed from the plots and table below.

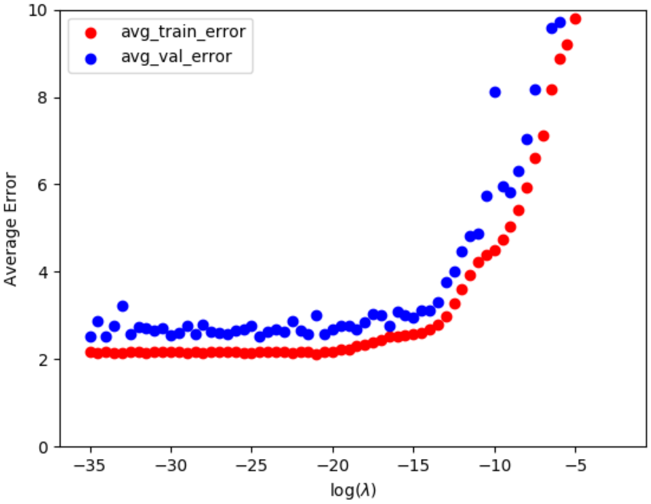


Figure 5: 1D Average Error vs λ

log(λ)=-35	log(λ)=-2
24.76967516	-0.80562741
276.84787831	0.80062943
1169.18356076	0.81113372
2532.28631709	-1.04418124
2999.30932852	0.53283648
1845.70202003	0.17102316
461.2532338	-0.8295211

When log(λ) is -2, magnitude of coefficients is less than 1, contrary to the case when log(λ) is -35, minimum magnitude of coefficient is 24.

2.2.6 Importance of Training data

Training data is extremely important for model selection. Model selection may go wrong because higher degree models might overfit training data leading to high variance problem. Lower degree model will be selected as it gives low validation error. This is illustrated in the following plot where training ratio is set to 0.2, validation ratio to 0.4 and testing ratio to 0.4. Lower degree model 5 was chosen eventhough degree 6 was a better model with larger training data set.

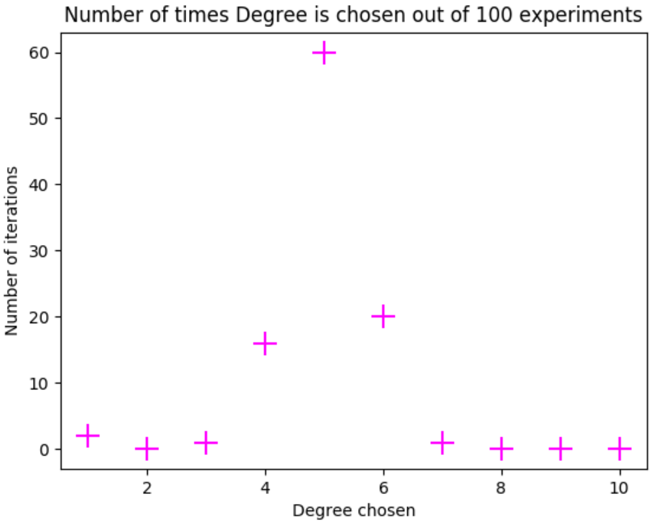
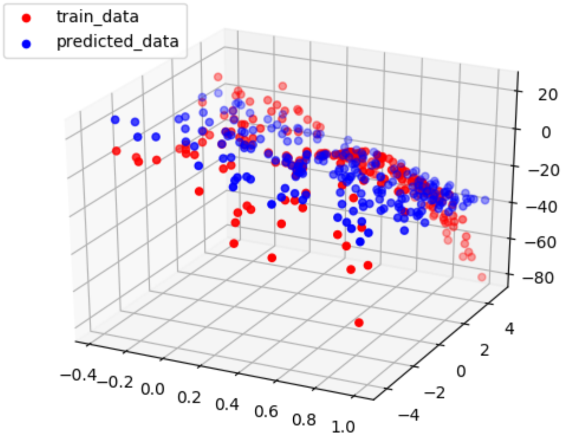


Figure 6: Less Training data

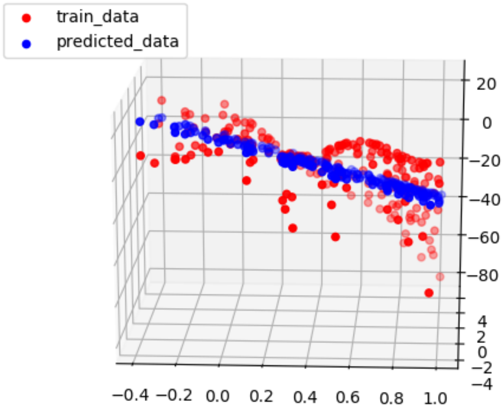
2.3 2D and Multidimensional Curve Fitting

2.3.1 Curve Fitting

In the experiment below we are using Ordinary least squares is used as error function. Training ratio is 0.7, validation ratio is 0.2 and testing ratio is 0.1. All polynomials till degree m=7 are considered for model selection. Since more than 1 feature is present, we need to incorporate interaction terms in the features. Interaction terms for degree 2 with two features will be x_1^2, $x_1 \cdot x_2$, x_2^2. Interaction terms for degree 3 with two features will be x_1^3, $x_1^2 \cdot x_2$, $x_1 \cdot x_2^2$, x_2^3. Degree 3 polynomial will contain bias, degree 1, degree 2 and degree 3 interaction terms. One can observe a plane fit when we consider degree 1 and a curve when we consider degree 2.



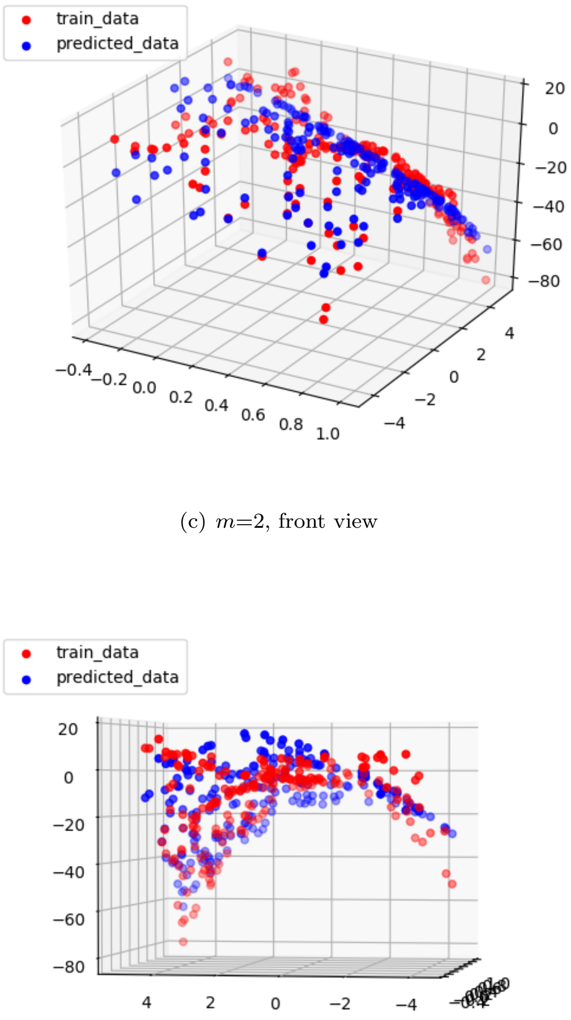
(a) m=1, front view



(b) m=1, side view

2.3.2 Interaction Features Generation

We were not able to find Python package that readily generates interaction terms. So we have written a simple for loop for its generation. Algorithm used is similar to monomial generation for a degree n with k terms. Step wise illus-

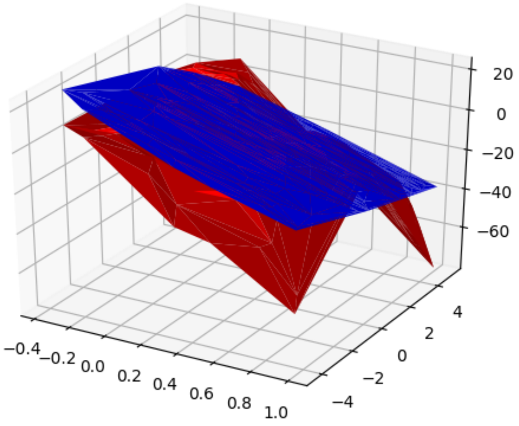


(c) m=2, front view

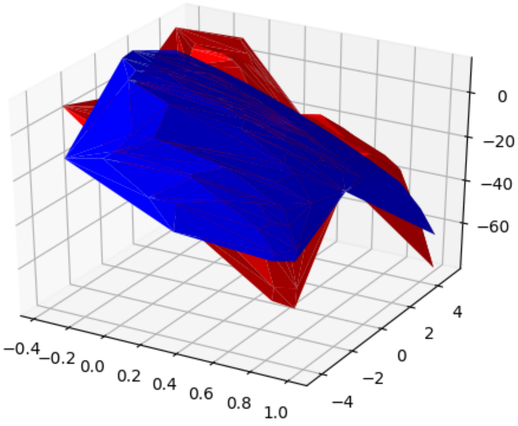
(d) m=2, side view

Figure 6: Scatter plots for 2D training data

tration for degree 3 with 2 terms is given below. Algorithm can generate it for any degree n and feature k.



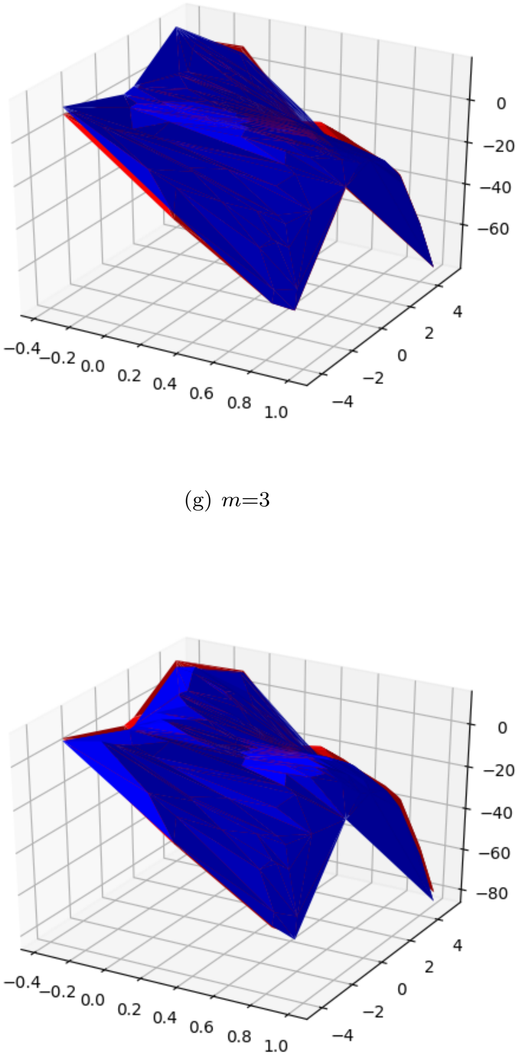
(e) m=1



(f) m=2

Step 1: multiplier = $[x_1, x_2]$, currentDegree = $[[x_1], [x_2]]$

Step 2: Multiply multiplier and currentDegree from i_{th} list to n_{th} list of currentDegree for i_{th} element of multiplier. currentDegree = $[[x_1^2, x_1 \cdot x_2], [x_2^2]]$ Flatten currentDegree and add it to training features.



(g) m=3

(h) m=4

Figure 6: Curve fitting for 2D training data

Step 3: Multiply multiplier and updated currentDegree from step 2 from i^{th} list to n^{th} list of currentDegree for i^{th} element of multiplier. currentDegree =

$[[x_1^3, x_1^2 \cdot x_2, x_1 \cdot x_2^2], [x_2^3]]$ Flatten currentDegree and add it to training features.

2.3.3 Closed Form Solution

Finding $(X^TX)^{-1}$ requires O(n^3) operations where n is the number of features in X. Since multidimensional data set contains only 9 features, we will be using closed form solution in 2D and multidimensional data. For 1D, it goes without being said that closed form solution can be used.

2.3.4 Feature Standardization

Features in 2D and multi-D data are not standard. Some are of 1×10^{-2} order , while some are of 1×10^2 order . Since we are using closed form solution, feature standardization is used to avoid X^TX to become an ill conditioned system. [2]

2.3.5 Repeated Cross Validation

Since data size is small(300) in 2D, repeated 10-fold cross validation is used. For multidimensional, we have large data set(10000), hence cross validation is not used.

2.3.6 Model Selection

As discussed in 1D case, point where validation error increases is the point of overfitting. This leads to choosing degree 3 polynomial as the model. One can also run experiment multiple times to confirm that degree 3 is indeed the model for 2D case.

[2] Feature Standardization in closed form solution

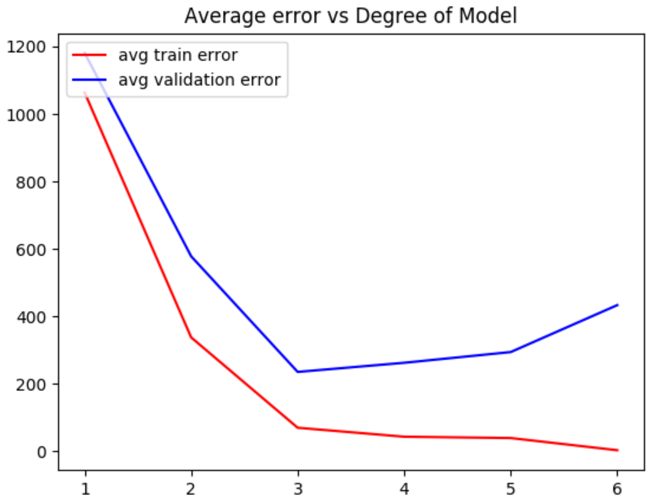


Figure 7: Average Error vs Degree of polynomial for 2D data

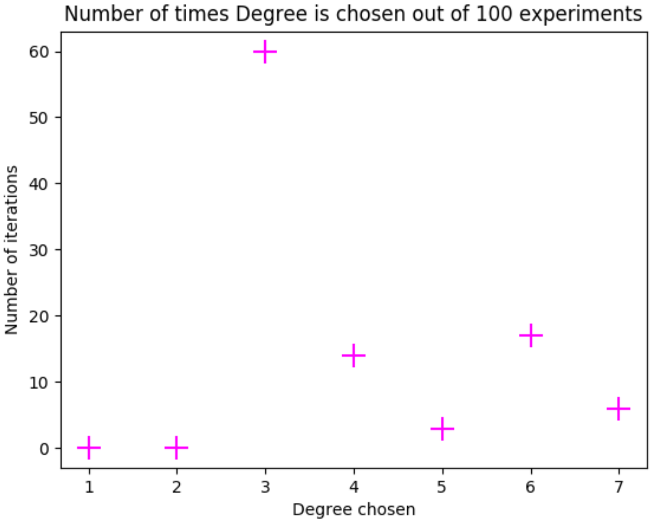


Figure 8: Degree chosen for 2D data

2.3.7 Correlated Feature Removal

Removing correlated features in multiple dimension speeds by interaction term generation. Suppose if x_1 and x_2 are correlated, x_2 can be expressed as (mx_1+c) and m and c gets absorbed in bias and weights in regression leaving out feature x_1. Using this, we can express any power of x_2 in terms of x_1 with the constants getting absorbed in weights and bias.

Model Selection in multidimensional data takes time if all 8 features are incorporated. This is because interaction term generation is time consuming. To generate interaction terms till degree 5 with 8 features took close to 20 minutes without completion!

This problem can be overcome by removing features that are highly correlated with other features. We are setting threshold for correlation detection as 0.7. Correlation matrix for multidimensional dataset is shown below,

$$\begin{bmatrix}
1 & -0.93 & 0.11 & -0.75 & -0.75 & -0.20 & 0.13 & -0.16 \\
-0.93 & 1 & -0.14 & 0.93 & 0.93 & 0.20 & -0.12 & 0.15 \\
0.11 & -0.14 & 1 & -0.27 & -0.27 & 0.02 & -0.03 & -0.03 \\
-0.75 & 0.93 & -0.27 & 1 & 1 & 0.14 & -0.03 & 0.13 \\
-0.75 & 0.93 & -0.27 & 1 & 1 & 0.14 & -0.03 & 0.13 \\
-0.20 & 0.20 & 0.02 & 0.14 & 0.14 & 1 & -0.65 & 0.49 \\
0.13 & -0.12 & -0.03 & -0.03 & -0.03 & -0.65 & 1 & 0.02 \\
-0.16 & 0.15 & -0.03 & 0.13 & 0.13 & 0.49 & 0.02 & 1
\end{bmatrix}$$

Observing the matrix we get to know that only features 1, 3, 6, 7 and 8 are uncorrelated. Feature 2 correlated with 1, Feature 4 correlated with 1 and Feature 5 correlated with 1. This reduces features from 8 to 5 and speeds up interaction terms generation 10 times(in just 2 min!) for 25 repetitions 10-fold cross validation.

Observing from the figure above, model chosen for multidimensional data is degree 4. One can confirm by running the experiment multiple times.

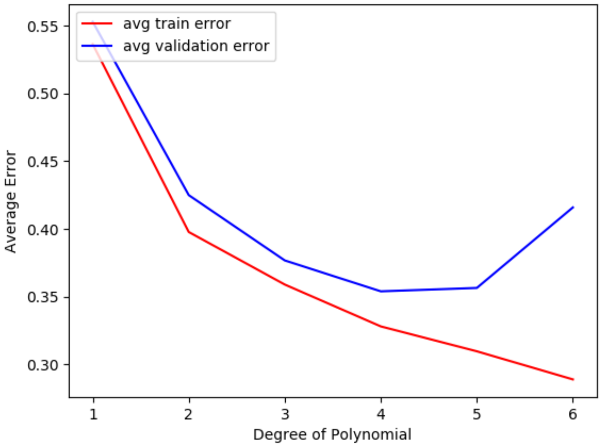


Figure 9: Average Error vs Degree of polynomial for multidimensional data

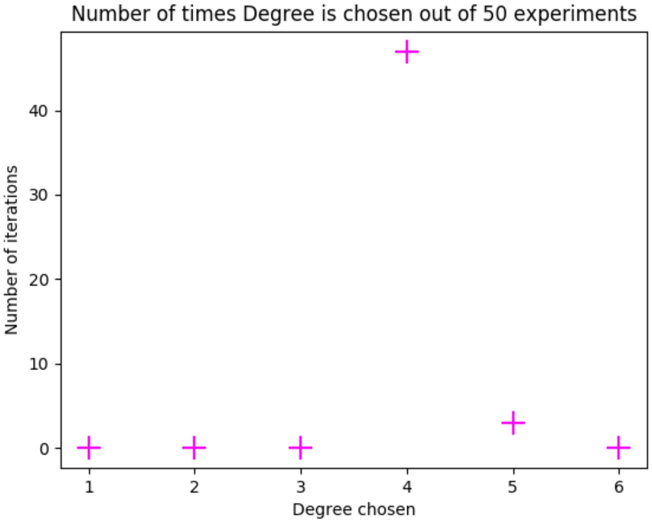
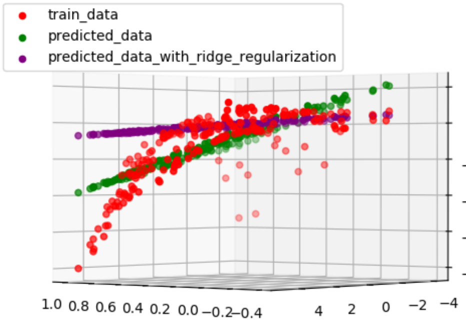


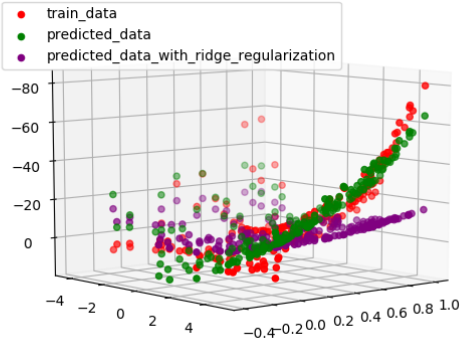
Figure 10: Average Error vs Degree of polynomial for multidimensional data

2.3.8 Ridge Regression

Ridge regression generates smooth or broad curves. For illustration purposes we have set λ value high to 1000 that forces magnitude of coefficients to be small. Look at the ridge regularized curves in figures below. Ridge regularized curves are much broader and smoother compared to ordinary least square curves.



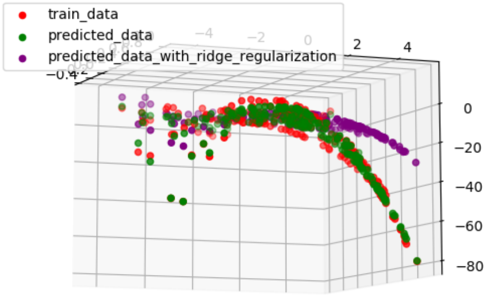
(a) m=1



(b) m=2

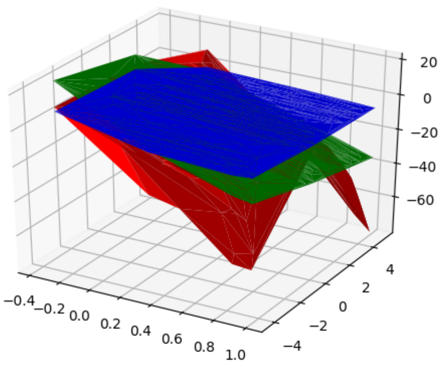
2.3.9 λ Selection

From ordinary least squares we deduced that degree 3 polynomial fits 2D data well and degree 4 polynomial fits multidimensional data well. As discussed in 1D case, to smoothen the polynomials we use ridge regression and find the coefficients of the degree 3(2D) and degree 4(multidimensional). From the figures below, e^2 is the best choice of λ for 2D case and multidimensional case.

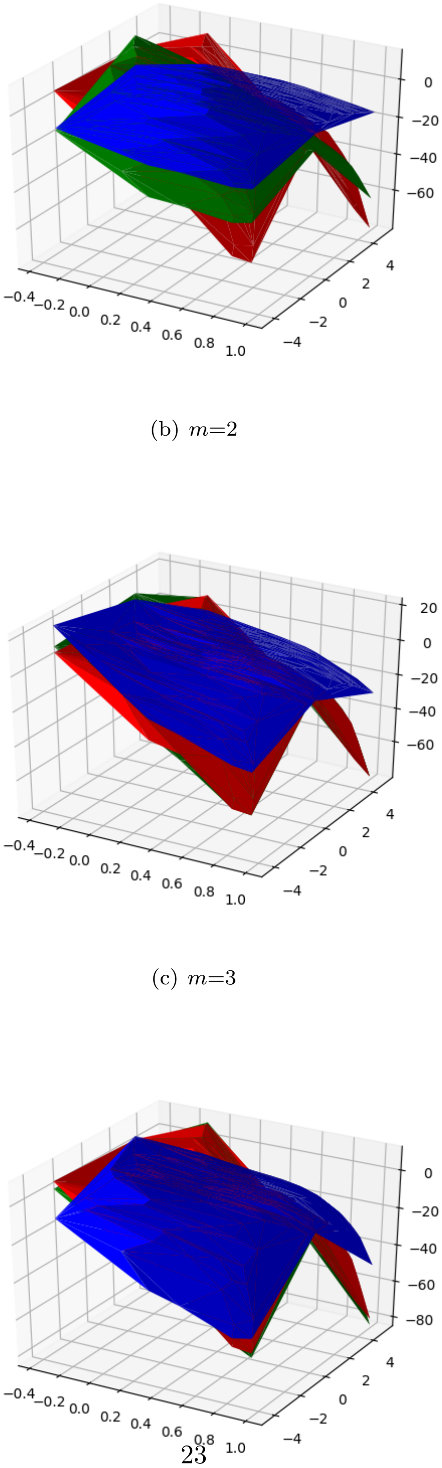


(c) *m*=3

Figure 11: Ridge Scatter Plots for 2D training data



(a) *m*=1



(b) m=2

(c) m=3

(d) m=4

Figure 11: Ridge Curve fitting for 2D training data

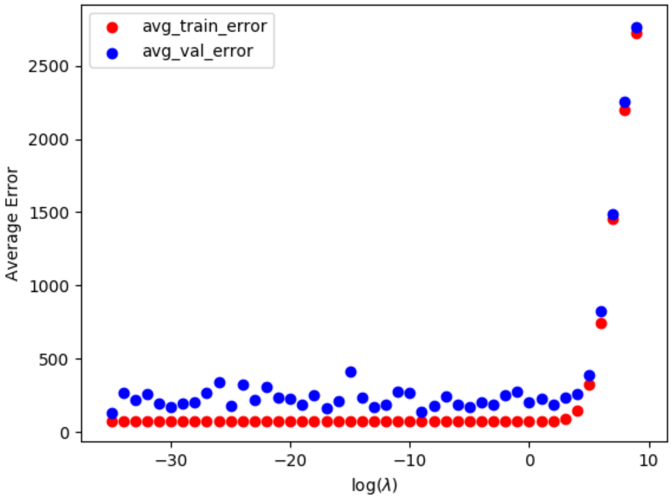


Figure 12: 2D Average Error vs λ

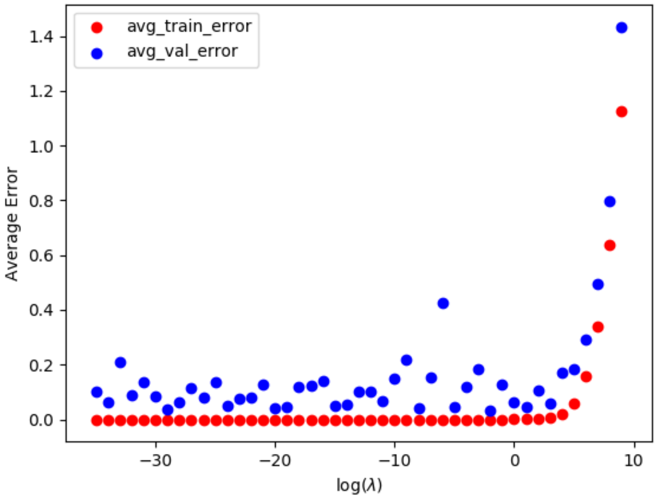


Figure 13: Multidimensional Average Error vs λ