CS F320: FODS Assignment 1

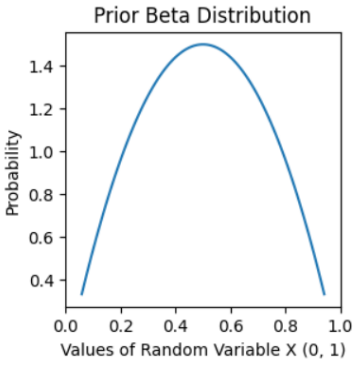
Group Members:

Arjun Muthiah	2019B3A70374H
Sanket Bhatt	2019A7PS0147H
Hitaishi Desai	2019B3A70602H

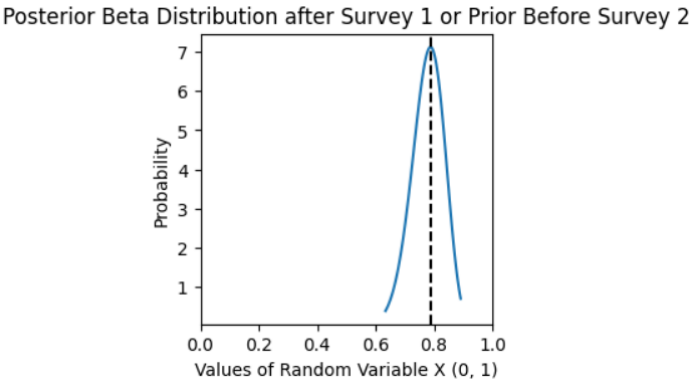
Question 1A

Prior and Posterior Distribution

The prior distribution of a probability is defined as the probability distribution before any evidence is taken into account. The initial prior can be estimated as the initial beta distribution with parameters α,β = (2,2).
Let us define a random variable X which takes on value 0 if a person does not like the update and 1 if they do.
The graph for this prior distribution is



The peak of this survey is at X = 0.5, inferring that there is equal probability that a person may like or dislike the update.
Now, post the first survey - there is some evidence which can manipulate the distribution. In formal terms, post the factoring of evidence, the prior distribution is updated to a posterior distribution which takes into account the likelihood of that evidence.
After the first survey (which has results of 40 favorable responses and 10 unfavorable responses) - the α,β values will get updated to α = 2+40 = 42 and β = 2+10 = 12 (by the definition of the beta distribution).
This will subsequently give the posterior probability distribution of

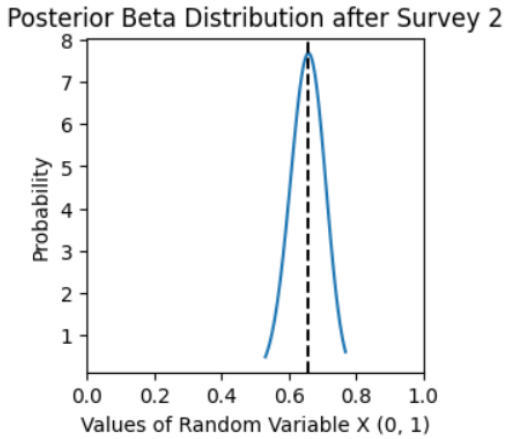


The peak of this distribution is at 0.7884 - inferring that 41/52 persons may like this update (taking into account both the prior and the evidence)

Intuitively, the posterior of survey 1 can be interpreted as the prior of survey 2 as it takes into account all the likelihoods before the evidence of survey 2.

After survey 2 (13 favored and 17 unfavored responses), the posterior distribution becomes one of $\alpha,\beta = 42+13, 12+17 = 55,29$.

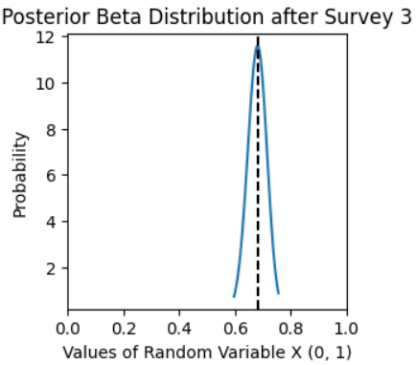
The posterior distribution is



The peak of this distribution is at 0.658.

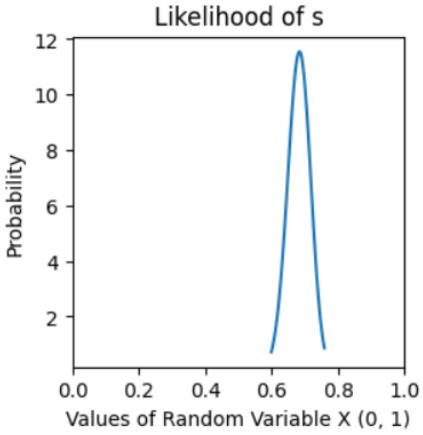
Finally, the posterior distribution post survey 3 (70 favored, 30 unfavored) is one of $\alpha,\beta = 55+70,29+30 = 125, 59$.

The final posterior distribution is



Now, the likelihood of a given probability distribution is calculated only taking into account the evidence provided. Hence, the likelihood is

P(favored opinion of update) = 40+13+70/(40+13+70+10+17+30) = 123/123+57 = 123/180 = 0.6833
P(unfavored opinion of update) = 57/180 = 0.3166
This can be described as a beta distribution



Hence, the likelihood and distribution of the survey can be defined thusly.

Question 1B

Gradient Descent: Algorithm Explanation

In this segment, we have attempted to develop a multivariate regression model to explain the effect of MLOGP and GATS1i on aquatic toxicity. To do so, we have run models varying from degrees 0 to 9 to find the one that minimizes the error.
To estimate the coefficients of the regression model, we have used both batch and stochastic gradient descent algorithms, and have also attempted to regularize the former.

Gradient descent algorithms are iterative optimization algorithms, commonly used to find the coefficients for which the cost function is minimum. Since our cost function

$$E = \frac{1}{2n}\sum(y_n - wx_n)^2$$

is convex, we start with a weight vector initialized to zero, updating it in each iteration by subtracting the product of our gradient at that point and our learning rate. Thus, with every iteration, we get closer and closer to our optimal weights, and, consequently our minimum possible cost.

$$w^{k+1} = w^k - \alpha\left(\nabla E(w)\right)_{w=w^k}$$

Stochastic gradient descent differs from the previously detailed by considering one training example per iteration, making it faster and more efficient.

In both types of gradient descents, we observe similar results.

$$w^{k+1} = w^{\kappa} - \alpha E\left(w^{k}, x_i, y_i\right)$$

We have taken care to keep our learning rate high enough to not run into a local minima trap, and can hence afford to cap the number of iterations at a sufficiently large number.

Batch Gradient Descent: Results

We used a gradient descent model with a learning rate of 10^-5 and 100 iterations to determine the degree of the polynomial that would best predict the LC50 value.
Upon running the model for polynomials of degrees 0 to 9, we found the following training and testing errors for each:

Degree	Training Error	Testing Error
0	2583.351559007729	617.9065606699786
1	513.5095304014449	118.46249344755215
2	685.3755313897041	166.97464868292244
3	8.229603025233518e+307	1.29603472083114e+307
4	nan	nan
5	nan	nan
6	nan	nan
7	nan	nan
8	nan	nan
9	nan	nan

For polynomials of degrees >3, we found the errors to be too large to compute.
Evidently, the polynomial of degree 1,

$$y = w_0 + w_1 x_1 + w_2 x_2$$

best fits the given data, with a training error of 513 and a testing error of 118, being the lowest of all degrees for both.

Stochastic Gradient Descent: Results

Our Stochastic Gradient Descent model uses a learning rate of 10^-3, cycling through all examples at random to compute both the gradient as well as the cost.
Since our initial values of the weight vector are randomized, we get different training and testing errors each time.
The errors from one execution are tabulated below but we find them to be similar each time.

Degree	Training Error(needs to be updated)	Testing Error
0	4929.633199203019	1273.77240577518
1	576.5443901494828	136.95029788996268
2	65094.94624114905	16372.87032651886
3	1065331.6480220456	214969.04893759685
4	61551687.62832603	7573105.359955335
5	8767962500.57954	633575138.1069396
6	1194513758805.3298	42672407381.9446
7	13549266358176.215	32726605235.1993
8	5.907547286027005e+64	3.5523113185111346e+62
9	2.272424455315455e+211	7.684243516626818e+208

Here too, we find the polynomial of degree 1 to be the one that explains the given data best.

Batch Gradient Descent with Regularization

With every gradient descent algorithm comes the threat of overfitting with polynomials of higher degree, i.e. the models explain the training data with close to zero error, but fail to fit the testing data, at the cost of the coefficients of the regression growing to exceptionally large values. To counter this, we introduce a regularization term into our optimization problem, and set it to the sum of some transformation on the positive values of w, usually either a modulus or a square transformation.

With this term we multiply a measure of the strictness of this constriction (lambda), a value between 0 and 1, 0 being the most lenient. Below are our results for one execution:

1. Q = 0.5

Lambda	Training Error	Testing Error
0.1	1942.3431328538463	501.8297379952431
0.2	1942.3433739284374	501.82980104088983
0.3	1942.3436150030698	501.82986408654705
0.4	1942.343856077736	501.8299271322131
0.5	1942.344097152441	501.8299901778888
0.6	1942.344338227188	501.83005322357565
0.7	1942.344579301968	501.83011626927095
0.8	1942.3448203767896	501.83017931497653
0.9	1942.3450614516457	501.8302423606913

2. Q = 1

Lambda	Training Error	Testing Error
0.1	637.9372954397261	146.68883047055505
0.2	637.9372462183574	146.68882050789992
0.3	637.9371969970178	146.68881054525218
0.4	637.9371477757089	146.6888005826121
0.5	637.9370985544292	146.68879061997941
0.6	637.9370493331802	146.68878065735444
0.7	637.9370001119606	146.6887706947369
0.8	637.9369508907719	146.68876073212704
0.9	637.9369016696116	146.68875076952446

3. Q = 2

Lambda	Training Error	Testing Error
0.1	1366.9140381105717	352.2728465905951
0.2	1366.9141973190913	352.2728885664609
0.3	1366.914356527614	352.27293054232746
0.4	1366.9145157361359	352.27297251819346
0.5	1366.9146749446595	352.2730144940599
0.6	1366.9148341531818	352.27305646992585
0.7	1366.914993361705	352.2730984457919
0.8	1366.9151525702289	352.2731404216579
0.9	1366.915311778754	352.27318239752395

4. Q = 4

Lambda	Training Error	Testing Error
0.1	765.882392902271	191.58943625495016
0.2	765.8824635194294	191.58945561576803
0.3	765.8825341365863	191.58947497658536
0.4	765.8826047537408	191.58949433740176
0.5	765.8826753708948	191.5895136982178
0.6	765.8827459880465	191.58953305903302
0.7	765.8828166051941	191.5895524198469
0.8	765.8828872223412	191.5895717806605
0.9	765.8829578394864	191.58959114147325

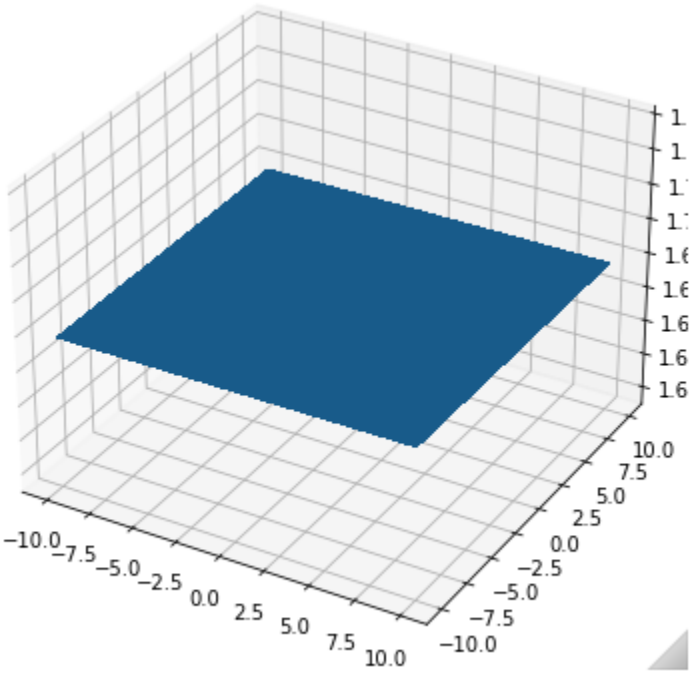
For the various values of q given, we ran models with lambda ranging from 0.1 to 0.9, and found that for all values of q, a lambda of 0.1 was optimal. Since our batch gradient descent left us with a polynomial of degree 1 having the lowest error, both training and testing, we observe that there is no overfitting here. Hence, in principle, there is no need to restrict the values of the weights, and any restriction we impose does little except restrict the values of cost function in the lower direction. Thus, for any given range, the optimal value of lambda would be the least value it could take on, and since in our case the lowest value is 0.1, we find that to be optimal.

Comparative Study of the Four Regression Models

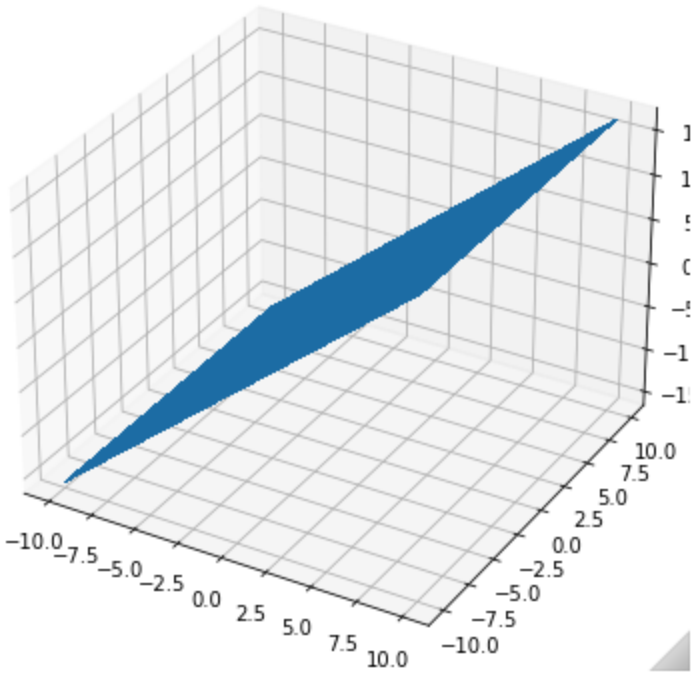
Our initial regression model of degree 1 gave a training error of around 513 and a testing error of 118 on taking initial values of the regression coefficients as 0. This approach, disappointingly, did not work for the regularized models, due to the limited number of data points. On observing the errors for the regularized models for various starting values of w, we found there to be no set trend from one value of q to another. For instance, on one execution, a value of q = 4 resulted in the least error of 646, while q = 0.5 yielded a maximum error of 1455. On one successive execution, we found q = 0.5 to give us a minimum error at 602, with q = 1 causing the maximum error of 3274. We did, however, observe consistently that the lowest value of lambda was the one that resulted in the error being minimum. Since there was neither overfitting nor explodingly large values of the weights, there is no need for any sort of regularization, and any we impose only constricts the value of our error function on the lower side. The ideal value of lambda, thus, would be zero.

Surface Plots of the Predicted Polynomials

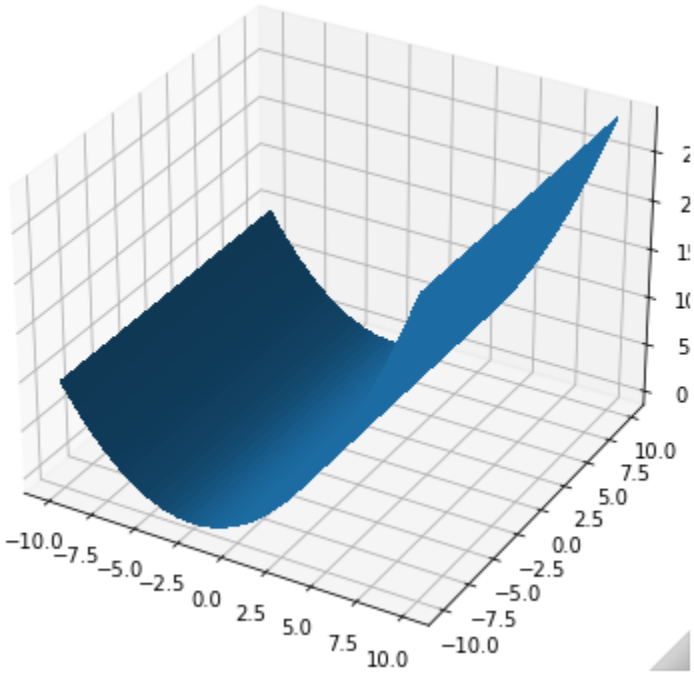
- Degree 0



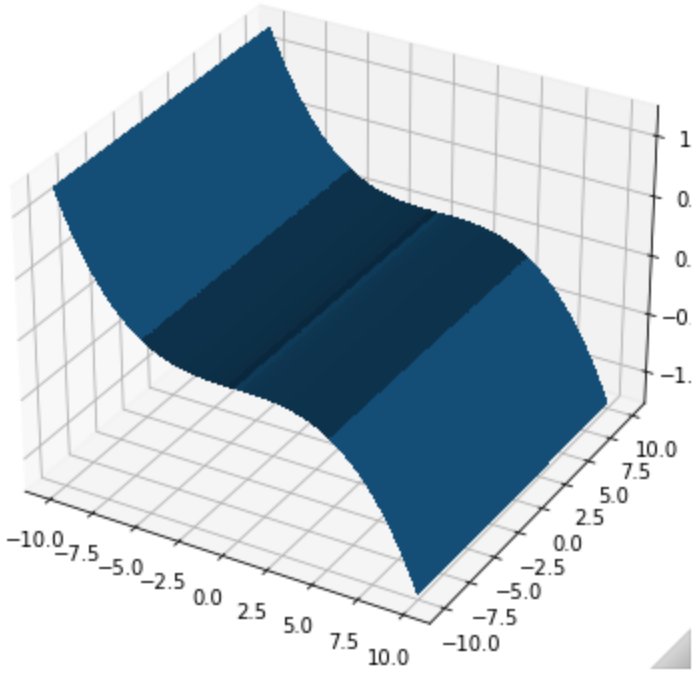
- Degree 1:



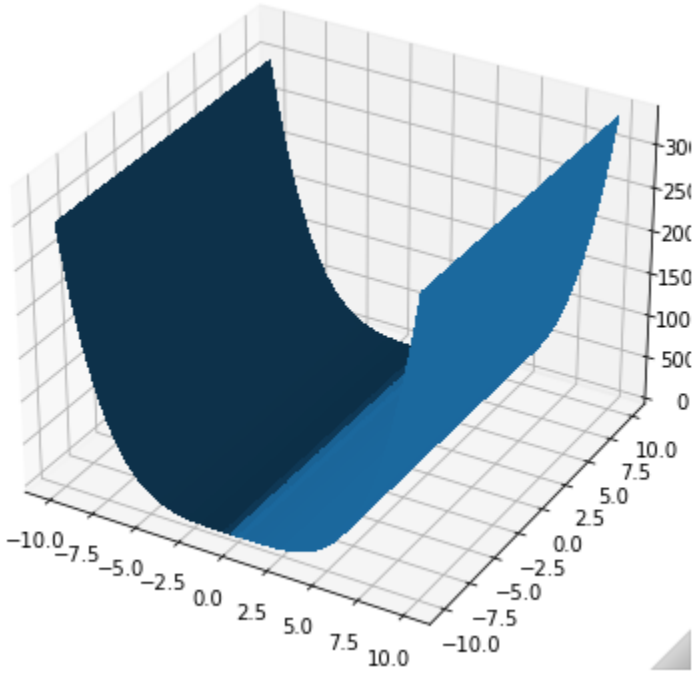
- Degree 2



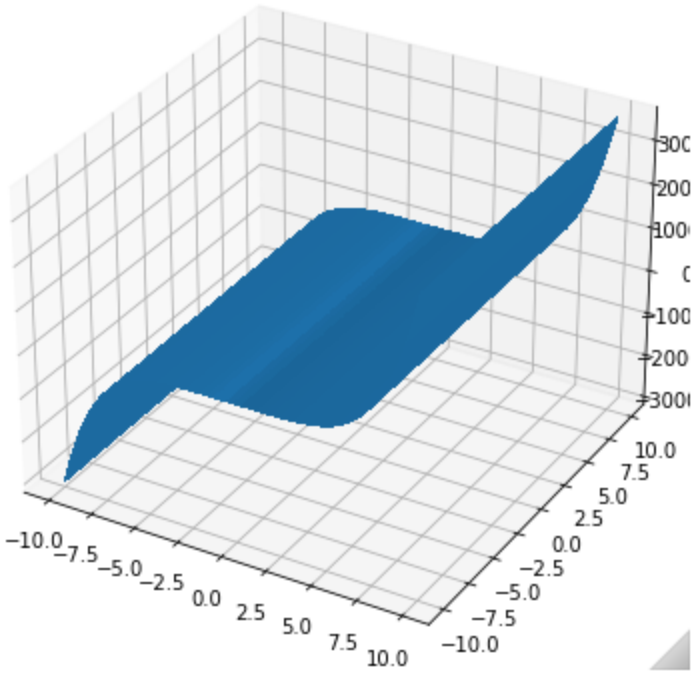
- Degree 3



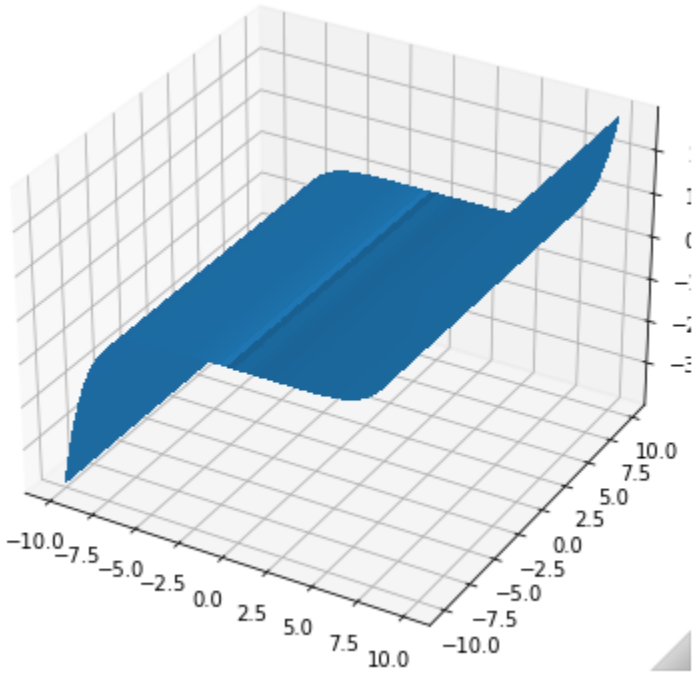
- Degree 4



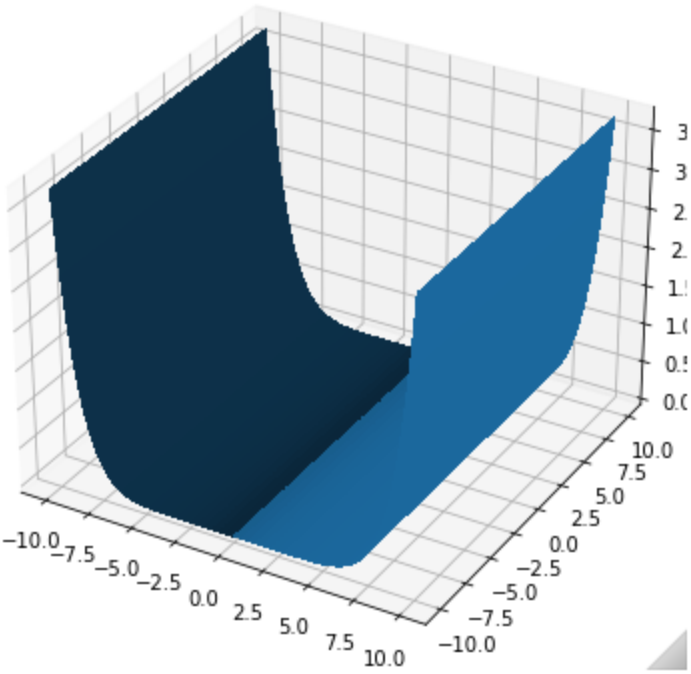
- Degree 5



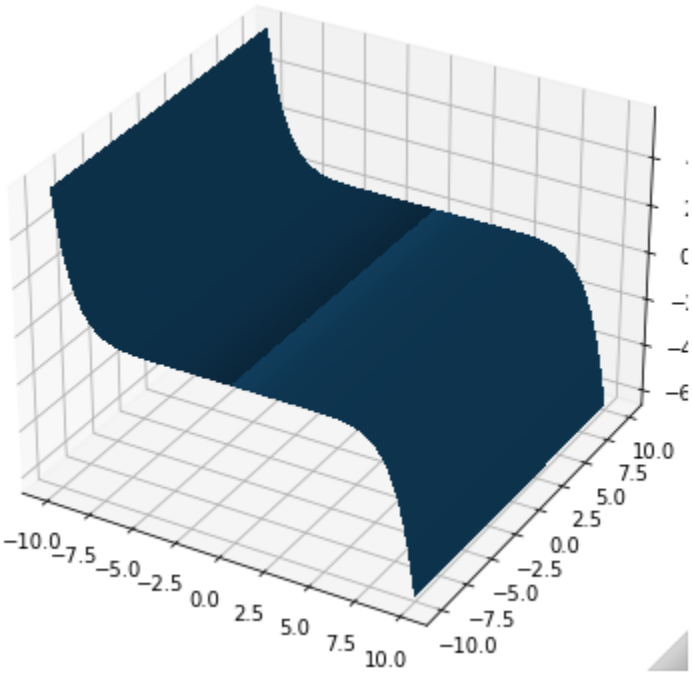
- Degree 7



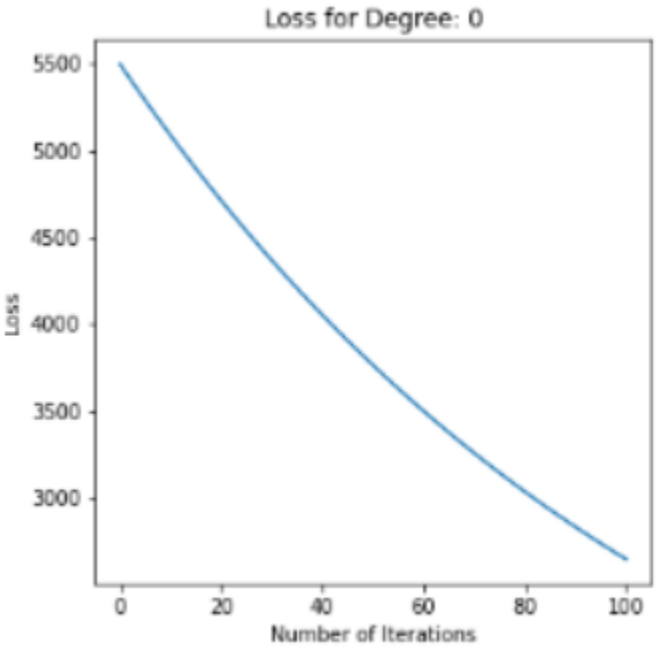
- Degree 8

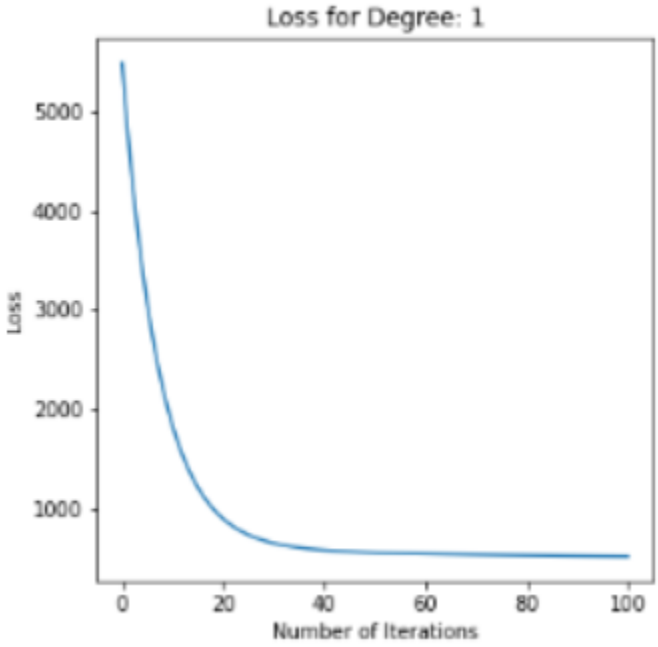

−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0

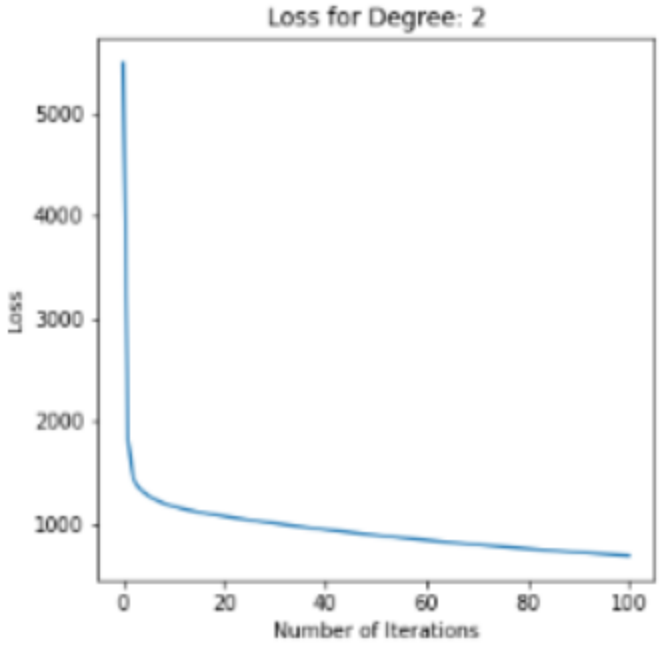
- Degree 9

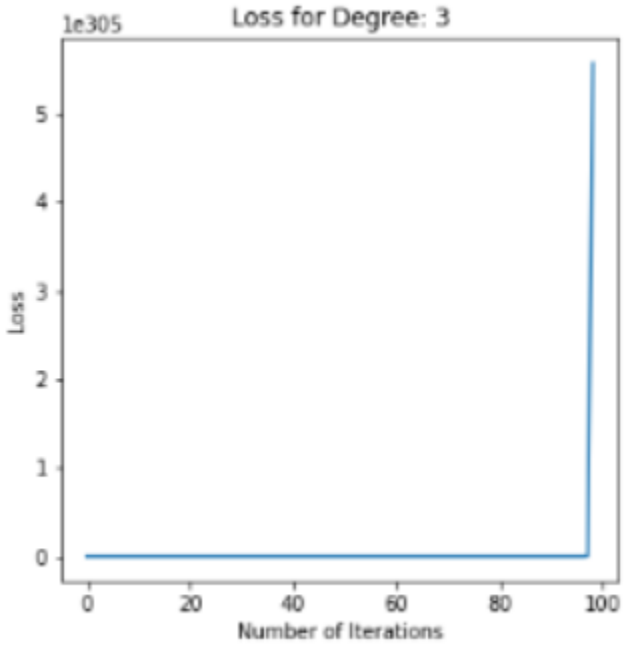

−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0

Error Plots


Loss for Degree: 0
5500
5000
4500
4000
3500
3000
Loss
0
20
40
60
80
100
Number of Iterations


Loss for Degree: 1
5000
4000
3000
2000
1000
Loss
0
20
40
60
80
100
Number of Iterations





For all subsequent degrees, our error plots look the same as that of degree 3, as the weights take on NaN values and the loss functions become too high to compute.

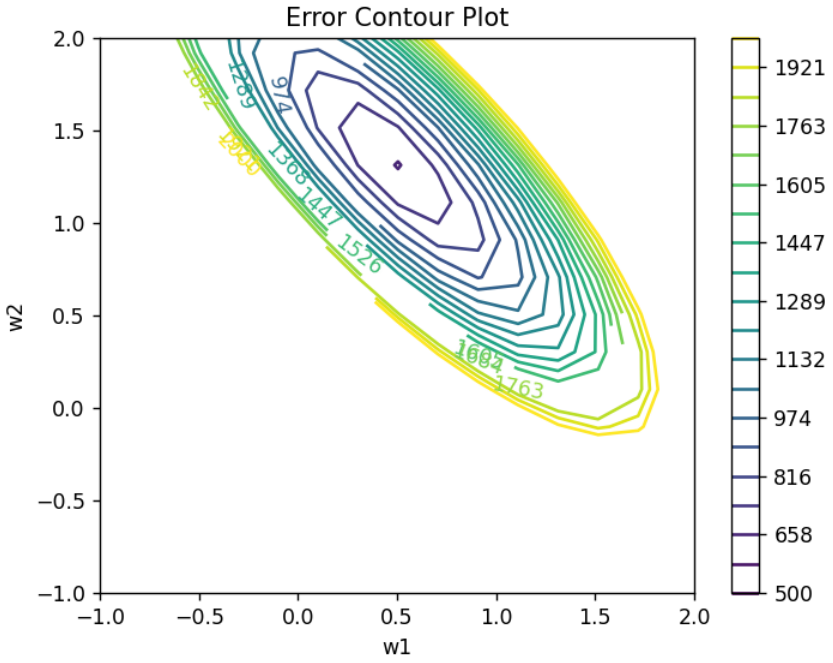
Question 1C

Visualizing Regularization

The error contour plot of any regression problem can be defined as the contour graph of its error function. The contour graph is a way of modeling an N>2 axis graph in 2 dimensions with the third dimension being flattened down to represent circles holding the same error value on their circumference. This implies that every point on the circle has equal error. In our case, we will take w1 and w2 (the 2 varying values) as the x and y axes and the error value (found by the loss function) will represent the contours.
The error function is

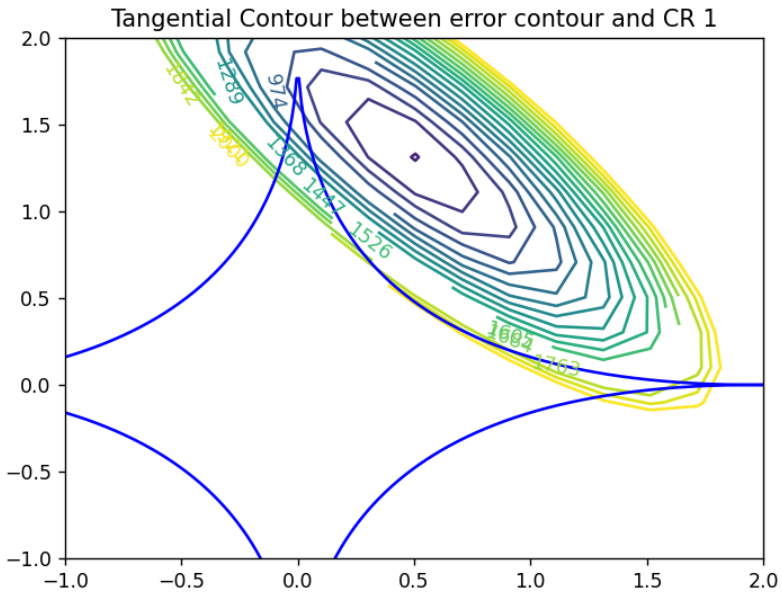
$$minimise\ E(w) = \frac{1}{2}\sum_{n=1}^{N}(y_n - t_n)^2$$

The error function plot after drawing is given below (The bar next to it denotes the range of the error values)



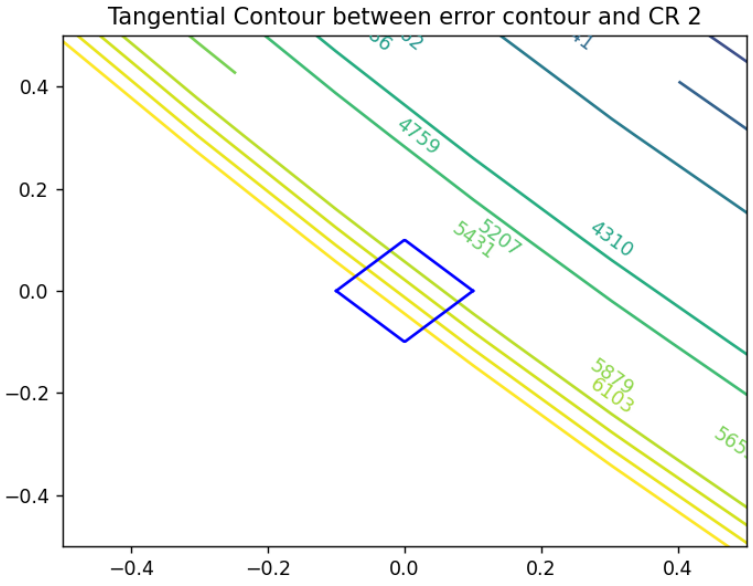
Note here that the regression function is represented as a degree 1 polynomial with no bias. The second part of this question asks us to draw the constraint regions and to find the point of tangential intersection of those constraint regions with the contour plot that insofar finds the minima of the error function given the constraints provided (represented by the boundaries of the constraint region)

1. Q = 0.5 and η = 1.4



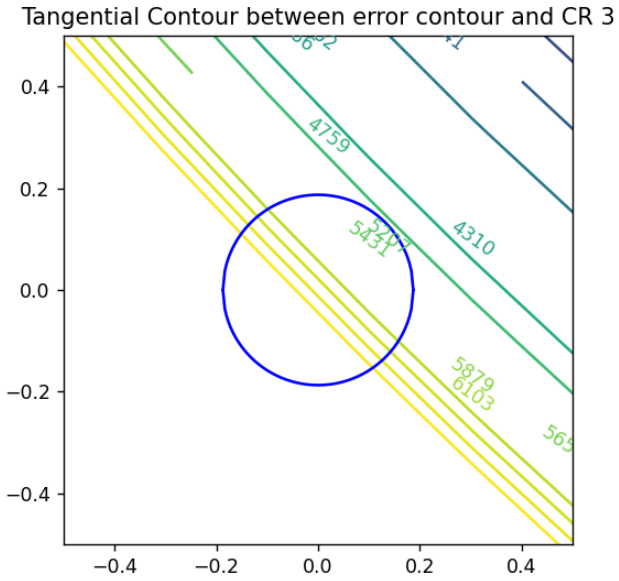
Purely visually speaking, the minima of this plot can be seen to be the topmost point. However, after calculating manually, the Intersection point (i.e where the minima occurs) between the error contour and the constraint region is [w1, w2] = [0.00594729, 1.75001506]. The Sum of Squares Error at that point is 820.887 - this can be confirmed by visual inspection and mathematical calculation of the loss function.

2. Q = 1 and η = 0.1



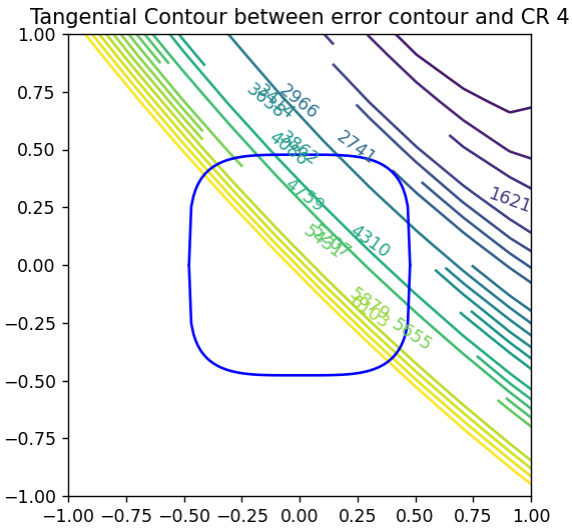
After performing Lagrangian mathematics on a solver, it can be found that the tangential intersection between the contour plot and the contour plot is [w1, w2] = [0.09123877, 0.00876123]. The Sum of Squares Error at that point is 6029.701.

3. Q = 2 and η = 0.035



The intersection point of this graph is w1, w2] = [0.13222999, 0.13234511]. The Sum of Squares Error at that point is 5045.281.

4. Q = 4 and η = 0.052



The Intersection point (i.e where the minima occurs) between the error contour and the constraint region is [w1, w2] = [0.39742991, 0.40555369]. The Sum of Squares Error at that point is 2535.209 (can be confirmed visually and empirically)

The Mean Square Errors for each case can be calculated by dividing the Sum of Squares Error by the total number of data points that exist within our Data Frame.

The Mean Square Error is
Case 1 - Q = 0.5 and η = 1.4 - MSE = 1.5034563324425156 per data point

Case 2 - Q = 1 and η = 0.1 - MSE = 11.043409784116067 per data point
Case 3 - Q = 2 and η = 0.035 - MSE = 9.240443222101723 per data point
Case 4 - Q = 4 and η = 0.052 - MSE = 4.643240049914559 per data point

Hence, it can be seen that a limited regularization (neither ridge nor lasso) of q = 0.5 will be better to reduce the Mean Square Error for that particular polynomial and the restriction of η = 1.4.
Note:- Thev alues of the error from 1B and 1C are going to be variably different for the following reasons

1. Our regularization models in 1C impose a constraint upon the value of η
2. The regression polynomial given in 1C has no bias (no value of w0) and hence, will give a vastly different polynomial (with a greater error) than that obtained from the non-regularized polynomial in 1B as well as the 4 regularization polynomials obtained thus. Furthermore, the comparative analysis of the same will also be different.