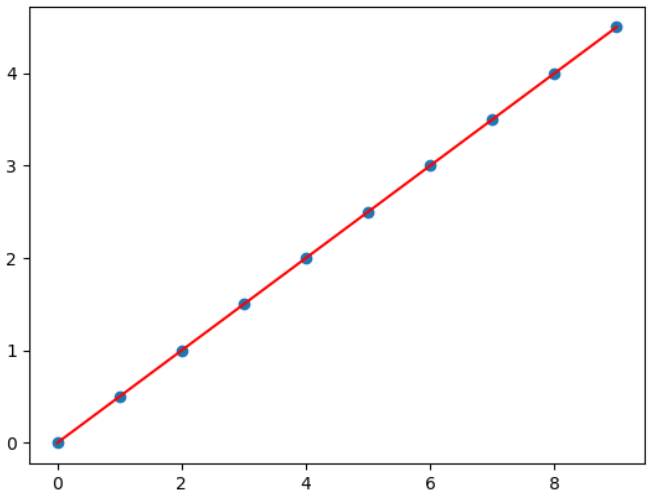
Homework 4

Question 1

a) ***1D-no-noise-lin***

Theta : [[0.0], [0.5]]

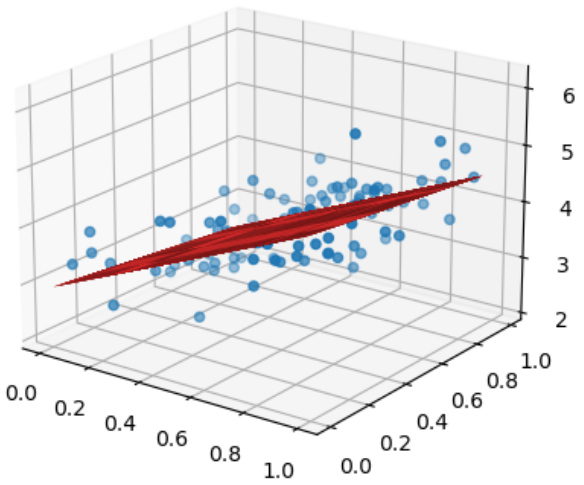
Loss : 0.0



2D-noisy-lin

Theta : [[2.9398743845692827], [2.0415614856856976], [-0.436838379937817 5]]

Loss : 0.1075928347520088



Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922

b) If one of the features is duplicated in a closed form solution, the inverse of the matrix cannot be calculated (i.e., $(X.X^T)^{-1}$) as the determinant will become 0.

This can be fixed by removing the duplicated feature using the correlation between the features.

c) When a training point is duplicated, following are the Theta values and losses

1D-no-noise-lin
Theta : [[8.881784197001252e-16], [0.4999999999999998]]
Loss : 2.196260474763044e-31

2D-noisy-lin
Theta : [[2.932841203907323], [2.033345135707229], [-0.42065597041809655]]
Loss : 0.10696989044546004

There is a slight difference compared to the original values but there will not be any problem similar to when a feature is duplicated.

However, the model becomes less generalized as it will be trained on these duplicate points and can become biased towards such points.

d) For Gradient Descent, when a feature is duplicated, we will not encounter any problem similar to what occurs in linear regression as we will not calculate inverses.

However, when a training point is duplicated, the model might be trained on the duplicate point and can become biased towards such points similar to what happens during linear regression.

Question 2

a) ***1D-no-noise-lin***
alpha : 0.05
num_iters : 10
initial_Theta : zero vector

Iteration 1
([[0.0], [0.0]], 3.5625)

Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922

Iteration 2
([[0.1125], [0.7125]], 0.7573828125000001)

Iteration 3
([[0.0590625], [0.384375]], 0.1615234863281249)

Iteration 4
([[0.082125], [0.5358515625]], 0.03493766798400873)

Iteration 5
([[0.0699521484375], [0.46628496093750005]], 0.008031704149217576)

Iteration 6
([[0.07404042480468748], [0.4985896582031506]], 0.00229943603847955)

Iteration 7
([[0.07065573046874997], [0.4839402996826172]], 0.0010651959009328556)

Iteration 8
([[0.0707363765167236], [0.4909278332794189]], 0.0007868575553308053)

Iteration 9
([[0.06924079520301817], [0.4879399861399841]], 0.0007120176497808081)

Iteration 10
([[0.06849225856137084], [0.4895463269698277]], 0.0006808439364351451)

b) ***1D-no-noise-lin***

Using closed-form solution:

Theta : [[0.0], [0.5]]
Loss : 0.0

Using gradient descent (with default parameters) solution:

Theta : [[6.447005124438478e-05], [0.49998971865582376]]
Loss : 6.017302596643245e-10

Both Theta and loss when applied gradient descent are very close to values from closed-form solution but not exactly the same.

2D-noisy-lin

Using closed-form solution:

Theta : [[2.9398743845692827], [2.0415614856856976], [-0.4368383799378175]]
Loss : 0.1075928347520088

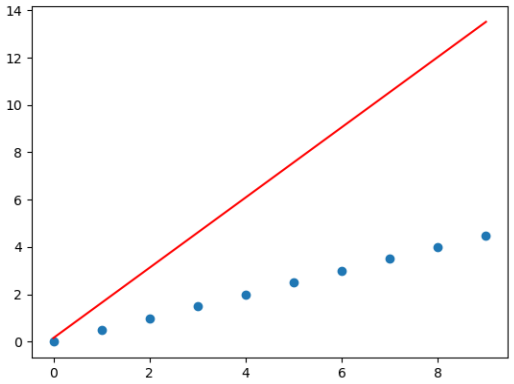
Using gradient descent (with default parameters) solution:

Theta : [[2.7559427810152752], [2.0967538862886435], [-0.16343704386978072]]
Loss : 0.11078751710215215

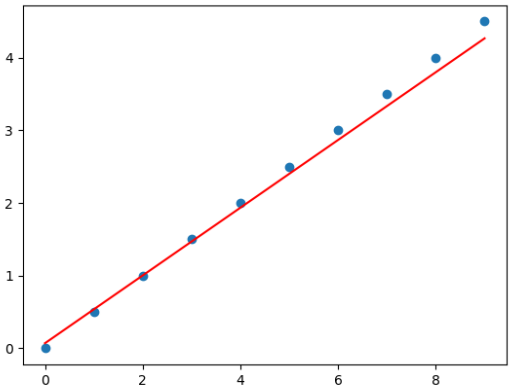
Similar to 1D data both theta and loss when applied gradient descent are very close to values from closed-form solution but not exactly the same.

c) ***1D-no-noise-lin***

When the learning rate is **> 0.068** or iterations are **<= 7** the line starts to deviate with the results being noticeably different from original and when the learning rate is **<= 0.068** and with ***default*** iterations it gives the correct results



Learning rate (alpha) = 0.068



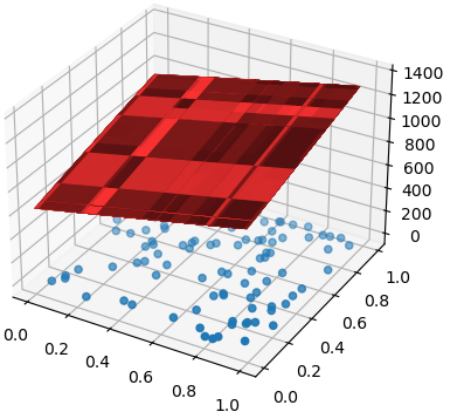
iterations = 5

Sai Abhinav Reddy Vasepalli
G01379942

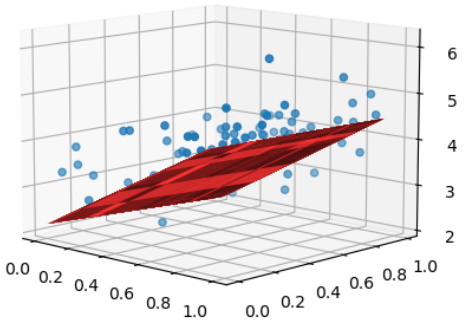
Amulya Jaladi
G01394922

2D-noisy-lin

When the learning rate is **>= 1.25** or iterations are **<= 30** the results are noticeably different from original and when the alpha is **<= 1.23** & iterations are **>= 100** the results are similar to the original values with a slight difference.



Learning rate (alpha) = 1.25



iterations = 30

As we can see from the above results, when the learning rate is low more number of iterations are required for convergence. If the learning rate is high it might converge early irrespective of the number of iterations it might not reach the original minima. If both the alpha and number of iterations are low, then the gradient will never reach the optimal point and never converge.

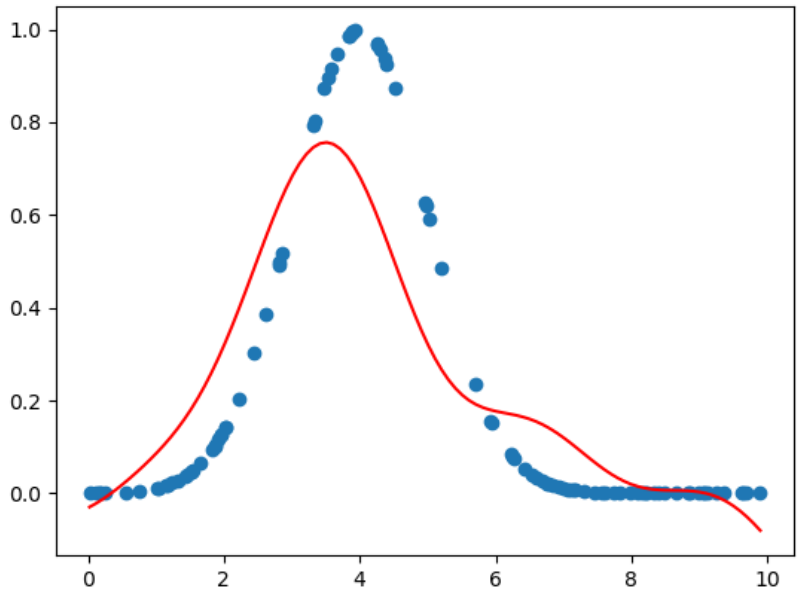
Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922

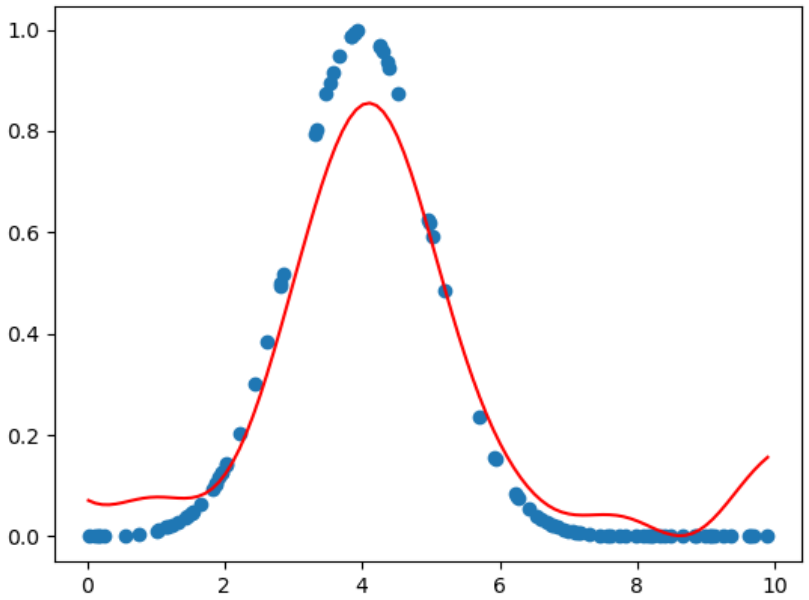
Question 3

a) ***1D-exp-samp***

Noticeably under fit with **K = 15** and number of iterations = 500



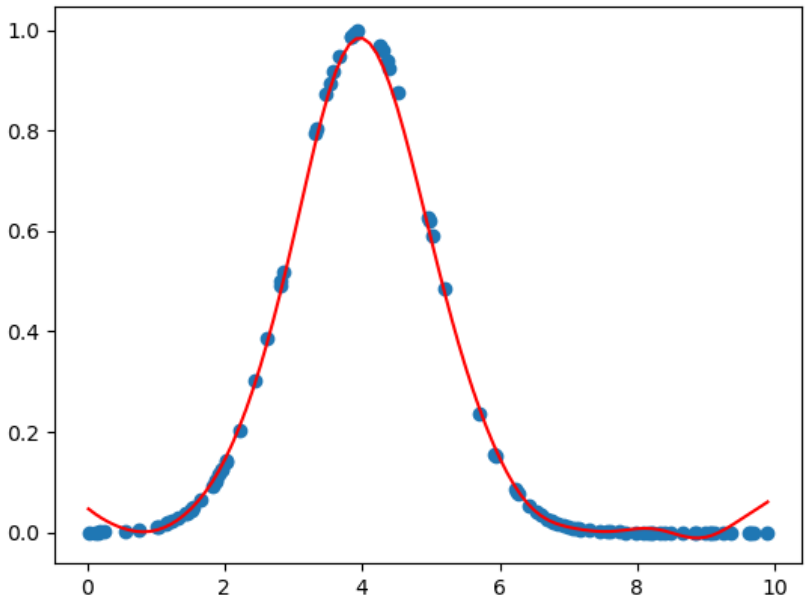
Medium fit with **K = 100** and number of iterations = 500



Sai Abhinav Reddy Vasepalli
G01379942

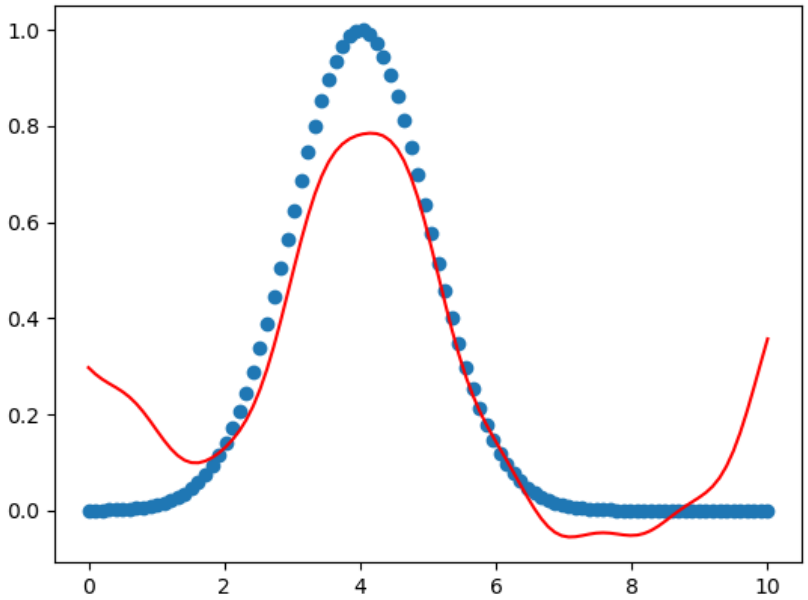
Amulya Jaladi
G01394922

Perfect fit with **K = 5000** and number of iterations = 1000



1D-exp-uni

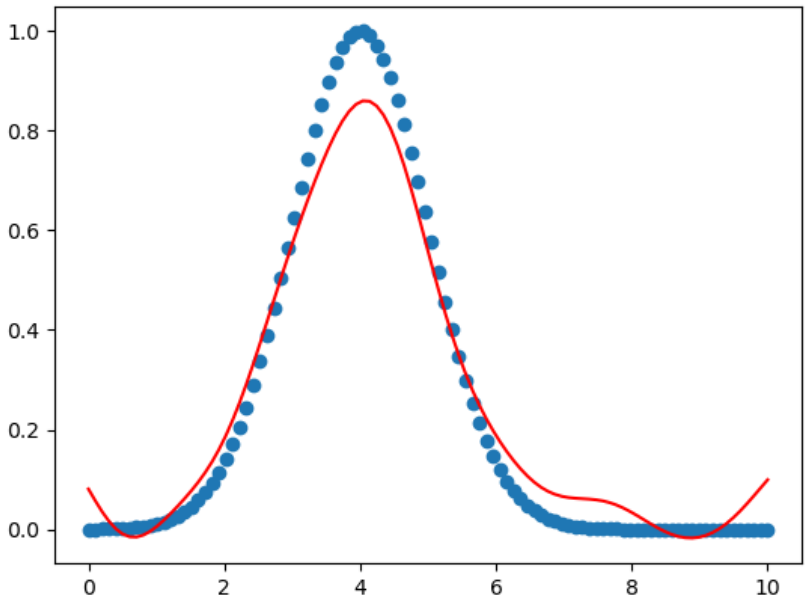
Noticeably under fit with **K = 15** and number of iterations = 500



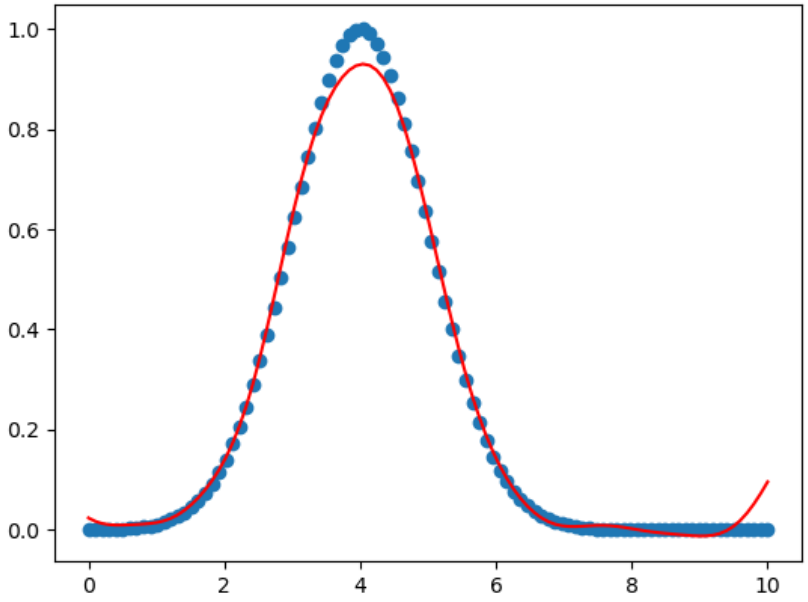
Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922

Medium fit with **K = 100** and number of iterations = 500



Perfect fit with **K = 5000** and number of iterations = 1000

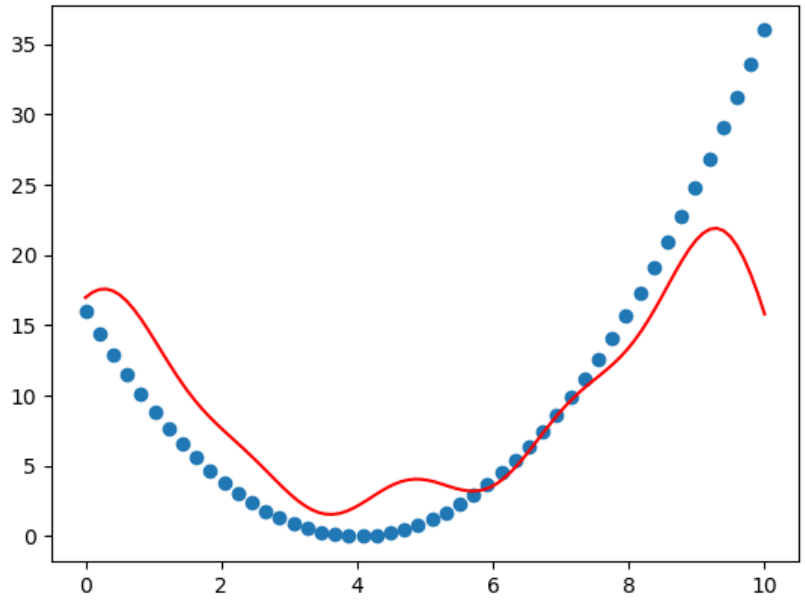


Sai Abhinav Reddy Vasepalli
G01379942

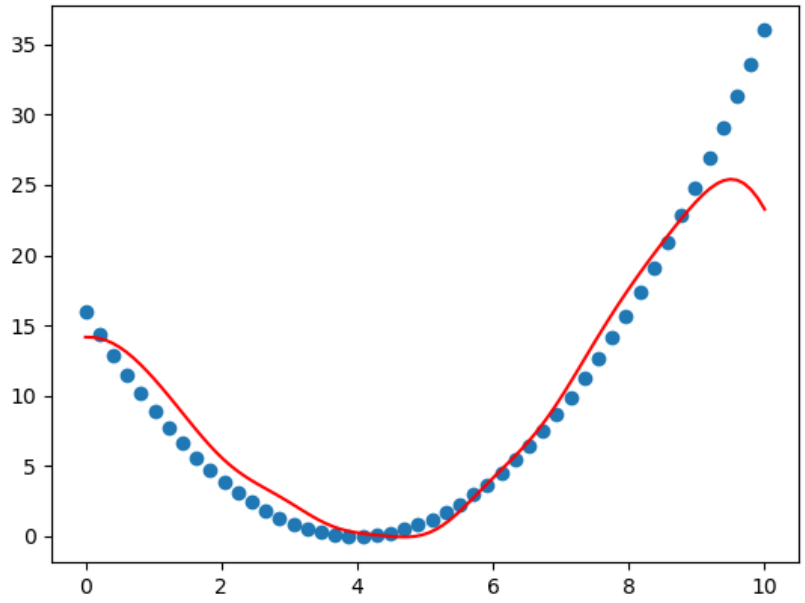
Amulya Jaladi
G01394922

1D-quad-uni

Noticeably under fit with **K = 15** and number of iterations = 500



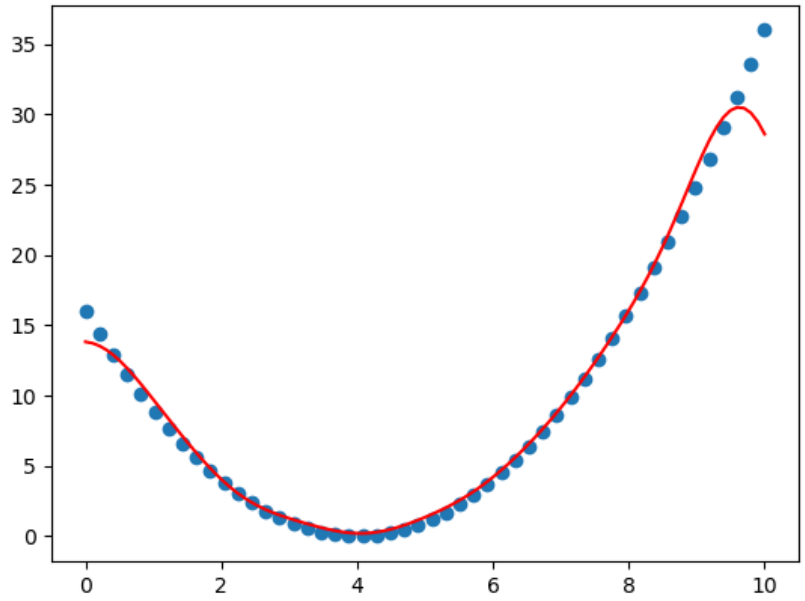
Medium fit with **K = 100** and number of iterations = 500



Sai Abhinav Reddy Vasepalli
G01379942

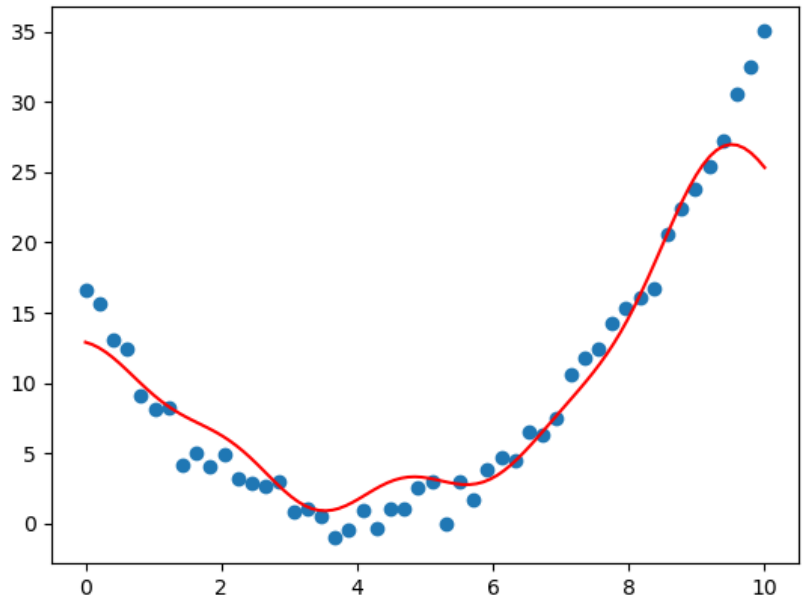
Amulya Jaladi
G01394922

Perfect fit with **K = 5000** and number of iterations = 1000



1D-quad-uni-noise

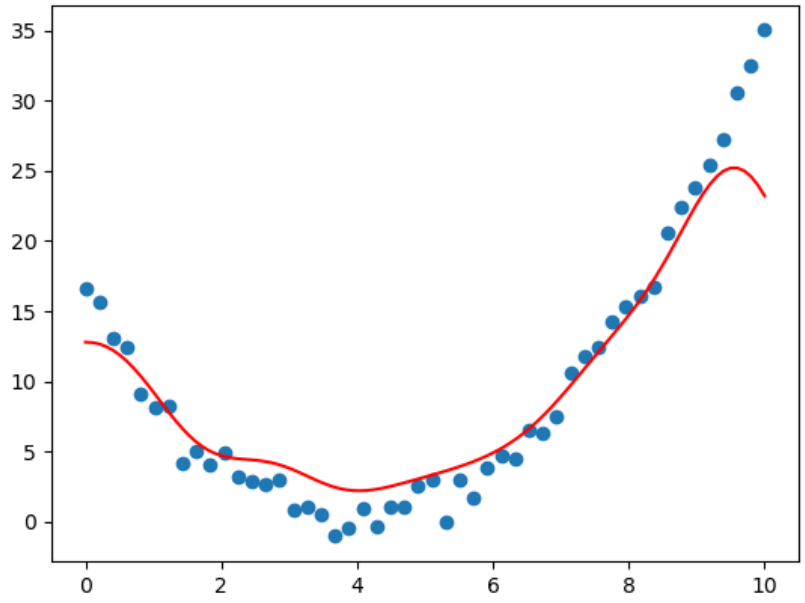
Noticeably under fit with **K = 15** and number of iterations = 500



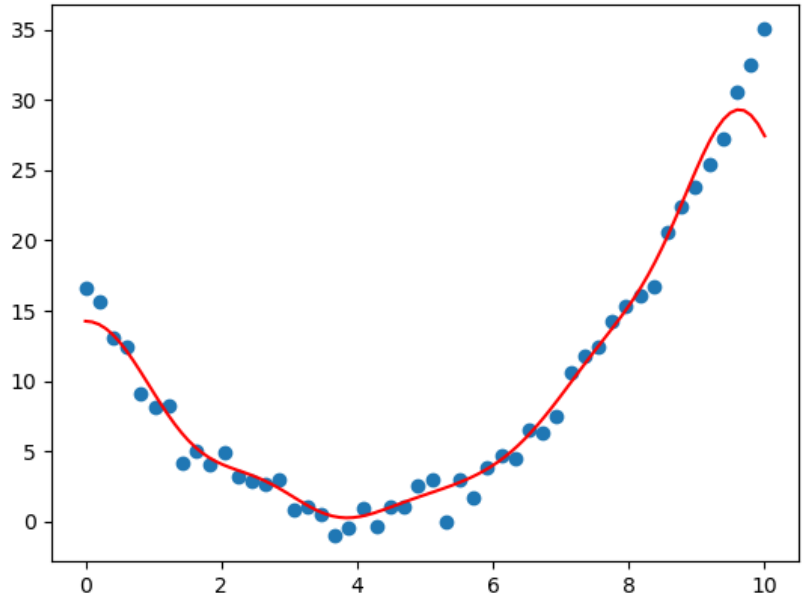
Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922

Medium fit with **K = 100** and number of iterations = 500



Perfect fit with **K = 5000** and number of iterations = 1000



Sai Abhinav Reddy Vasepalli
G01379942

Amulya Jaladi
G01394922