Homework I

STAT 6910/7810-002 - Spring semester 2019

Due: Friday, January 18, 2019 - 5:00 PM

Please put all relevant files & solutions into a single folder titled `<lastname and initials>_assignment1` and then zip that folder into a single zip file titled `<lastname and initials>_assignment1.zip`, e.g. for a student named Tom Marvolo Riddle, `riddletm_assignment1.zip`. Include a single PDF titled `<lastname and initals>_assignment1.pdf` and any Python scripts specified. Any requested plots should be sufficiently labeled for full points.

Unless otherwise stated, programming assignments should use built-in functions in Python, Tensorflow, and PyTorch. In general, you may use the `scipy` stack [?]; however, exercises are designed to emphasize the nuances of machine learning and deep learning algorithms - if a function exists that trivially solves an entire problem, please consult with the TA before using it.

Problem 1

Provide an example application where each of the following architectures or techniques would be useful. Do not reuse examples from class.

1. Multilayer perceptron - Time Series Prediction, Autonomous Driving
2. Convolutional Neural Network - Medical imaging, Radiology, Object Tracking
3. Recurrent Neural Network - Speech recognition, Rhythm Learning
4. Autoencoder - Dimension Reduction, NLP
5. Generative Adversarial Network - Super resolution Image Enhancement
6. Deep reinforcement learning - Self Driving Cars

Include your answers in a PDF titled `<lastname and initials>_assignment1.pdf`.

Problem 2

1. For a matrix A, we write $A \succeq 0$ and $A \succ 0$ when A is positive semi-definite (PSD) or positive definite (PD), respectively. Using the definition of a PD matrix, prove that the sum of two PD matrices is also PD. A very similar approach can be used to prove the sum of two PSD matrices is also PSD (although you don't have to prove it).

   Ans: Let A and B be two positive definite matrices:

   $$x^T Ax > 0,\ x^T Bx > 0$$

   On addition, the following holds:

   $$x^T Ax + x^T Bx > 0$$

   From the laws of matrix multiplication we get:

   $$x^T (A + B) > 0$$

   Thus, the sum of two PD matrices is also a PD matrix.

2. Is the sum of a PD matrix and a PSD matrix necessarily PD, PSD, or neither? Explain why.

   Ans: Yes. Since a PSD matrix is simply:

   $$x^T Ax >= 0$$

   The addition of a PD and a PSD matrix will be similar to the above proof and with one term greater than 0 and the other greater than equals to 0. So, we can say that sum of PD and PSD matrices will result in a sum that is greater than 0. Thus, the sum of two PD and PSD matrices is a PD matrix.

3. Consider the following matrices:

   $$A = \begin{bmatrix} 1 & -2 \\ 3 & 4 \\ -5 & 6 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, C = \begin{bmatrix} 2 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 2 \end{bmatrix}$$

   Determine whether the following matrices are PD, PSD, or neither. Briefly explain why. You may use the NumPy library for this problem.
   Hint: A symmetric matrix is PD if and only if all of its eigenvalues are greater than zero. A symmetric matrix is PSD if and only if all of its eigenvalues are greater than or equal to zero.

   (a) A - Not PD, since A is not a square matrix.

   (b) $A^T A$ - PD, since all of its Eigen values are positive.

(c) AA^T - PSD, since all of its Eigen values are non-negative.

(d) B - Neither PD or PSD, since some of its Eigen values are negative.

(e) $-B$ - Neither PD or PSD, since some of its Eigen values are negative.

(f) C - PD, since all of its Eigen values are positive.

(g) $C - 0.1 \times B$ - PD, since all of its Eigen values are positive.

(h) $C - 0.01 \times AA^T$ - PD, since all of its Eigen values are positive.

Include your answers in the same PDF titled `<lastname and initials>_assignment1.pdf`.

Problem 3

Consider the function $\mathbf{f} : \mathbb{R}^2 \to \mathbb{R}^2$ as defined below:

$$\mathbf{f}(\mathbf{v}) = \begin{bmatrix} f_1(\mathbf{v}) \\ f_2(\mathbf{v}) \end{bmatrix} = \begin{bmatrix} v_1^2 + 3v_1 e^{v_2} \\ 4v_1^3 v_2 - v_1 v_2 \log v_2 \end{bmatrix}.$$

1. Compute the gradient and Hessian of f_1.

   Ans:

   $$\nabla f_1 = \begin{bmatrix} 2v_1 + 3e^{v_2} & 3v_1 e^{v_2} \end{bmatrix}$$

   $$H = \begin{bmatrix} 2 & 3e^{v_2} \\ 3e^{v_2} & 3v_1 e^{v_2} \end{bmatrix}$$

2. Compute the gradient and Hessian of f_2.

   Ans:

   $$\nabla f_2 = \begin{bmatrix} 12v_1^2 v_2 - v_2 v_2 log v_2 & 4v_1^3 - v_1 - v_1 log v_2 \end{bmatrix}$$

   $$H = \begin{bmatrix} 24v_1 v_2 & 12v_1^2 - log(v_2) - 1 \\ 12v_1^2 - log(v_2) - 1 & -v_1/v_2 \end{bmatrix}$$

3. Compute the Jacobian of $\mathbf{f}$.

   Ans:

   $$\mathbf{J} = \begin{bmatrix} 2v_1 + 3e^{v_2} & 3v_1 e^{v_2} \\ 12v_1^2 v_2 - v_2 v_2 log v_2 & 4v_1^3 - v_1 - v_1 log v_2 \end{bmatrix}$$

Include your answers in the same PDF titled `<lastname and initials>_assignment1.pdf`.

Problem 4

1. Consider two arbitrary random variables X and Y. For the following equations, describe the relationship between them. Write one of four answers to replace the question mark: "=", "≤", "≥", or "depends". Choose the most specific relation that always holds and briefly explain why. Assume all probabilities are non-zero.

   (a) $\Pr(X=x, Y=y) \, ? \, \Pr(X=x)$
   Ans.$\Pr(X=x, Y=y) \;<= \; \Pr(X=x)$
   P(X and Y) = P(X).P(Y). Since P(Y) is less than 1, P(X).P(Y) is less than P(X).

   (b) $\Pr(X=x, Y=y) \, ? \, \Pr(X=x)$
   Ans.$\Pr(X=x|Y=y) \;<= \; \Pr(X=x)$
   P(X) given Y = P(X) if they're independent and = P(X and Y) if they're dependent. P(X and Y) is less than P(X), see (a).

   (c) $\Pr(X=x|Y=y) \, ? \, \Pr(Y=y|X=x)\Pr(X=x)$
   Ans. $\Pr(X=x|Y=y) \;>= \; \Pr(Y=y|X=x)\Pr(X=x)$
   Same reason as above. P(X) given Y = P(X) if they're independent and = P(X and Y) if they're dependent. In either case, from (a), the statement holds.

Include your answers in the same PDF titled `<lastname and initials>_assignment1.pdf`.

Problem 5

1. Suppose we have d-dimensional data points $\mathbf{x}_1, \dots, \mathbf{x}_n$ and corresponding real-valued response variables $y_1, \dots, y_n$. In regression, we are trying to learn a function $f(\mathbf{x})$ such that $y \approx f(\mathbf{x})$. For linear regression, we assume that f is a linear function: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ where w_0 is an offset term.

   One approach to approximating y is to minimize the empirical mean-square-error (MSE). The empirical MSE can be written as

   $$\sum_{i=1}^{n} (\mathbf{w}^T\mathbf{x}_i + w_0 - y_i)^2.$$

   Now let X be a $n \times d+1$ matrix where the ith row corresponds to $[\mathbf{x}_i^T, 1]$ where the 1 term is added to include the offset term in the regression model. Also let y be a n-dimensional vector of the response variables where the ith entry corresponds to y_i. Show that the linear regression solution that minimizes the empirical MSE is

   $$\mathbf{w}^* = (X^TX)^{-1}X^T\mathbf{y}.$$

   Ans:

Residual of MSE:

$$R^2 = \sum_{i=1}^{n} (\mathbf{w}^T \mathbf{x}_i + w_0 - y_i)^2.$$

The partial derivatives are:

$\frac{\partial R^2}{\partial a_0} = -2 \sum_{i=1}^n [y - (a_0 + a_1 x + ... + a_k x^k)] = 0$

$\frac{\partial R^2}{\partial a_1} = -2 \sum_{i=1}^n [y - (a_0 + a_1 x + ... + a_k x^k)]x = 0$

$\frac{\partial R^2}{\partial a_k} = -2 \sum_{i=1}^n [y - (a_0 + a_1 x + ... + a_k x^k)]x^k = 0$

On solving:

$a_0 n + a_1 \sum_{i=1}^n x_1 + ...a_k \sum_{i=1}^n x_1^k = \sum_{i=1}^n y_i$

$a_0 \sum_{i=1}^n x_1 + a_1 \sum_{i=1}^n x_1 ...a_k \sum_{i=1}^n x_1^k = \sum_{i=1}^n x_i y_i$

$a_0 \sum_{i=1}^n x_1^k + a_1 \sum_{i=1}^n x_1^{k+1} ...a_k \sum_{i=1}^n x_1^{2k} = \sum_{i=1}^n x_i^k y_i$

$$\begin{bmatrix} n & \sum_{i=1}^n x_1 & ... & \sum_{i=1}^n x_1^k \\ \sum_{i=1}^n x_1 & \sum_{i=1}^n x_1^2 & ... & \sum_{i=1}^n x_1^{k+1} \\ ... & ... & ... & ... \\ \sum_{i=1}^n x_1^k & \sum_{i=1}^n x_1^(k+1 & ... & \sum_{i=1}^n x_1^{2k} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ ... \\ a_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \\ ... \\ \sum_{i=1}^n x_i^k y_i \end{bmatrix}$$

This is a Vandermonde matrix, which can also be written as:

$$\begin{bmatrix} 1 & x_1 & ... & x_1^k \\ 1 & x_2 & ... & x_2^k \\ ... & ... & ... & ... \\ 1 & x_n & ... & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ ... \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix}$$

or

$$\begin{bmatrix} y_1 \\ y_2 \\ ... \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & ... & x_1^k \\ 1 & x_2 & ... & x_2^k \\ ... & ... & ... & ... \\ 1 & x_n & ... & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ ... \\ a_k \end{bmatrix}$$

Writing the Matrix notation as a polynomial equation:

$$y = Xa$$

Pre-Multiplying both sides by X transpose

$$X^T y = X^T X a$$

On Inversion

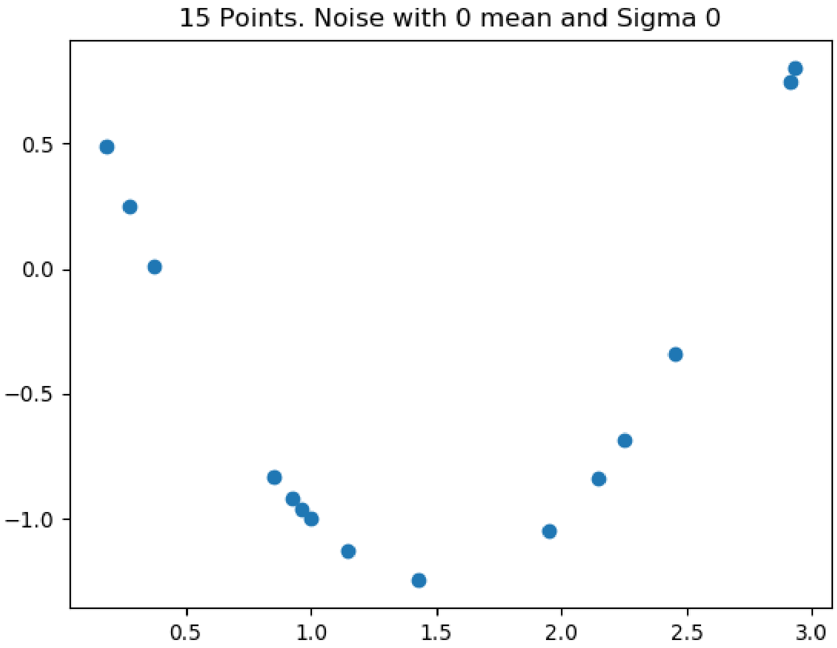
$$a = (X^T X)^{-1} X^T y$$

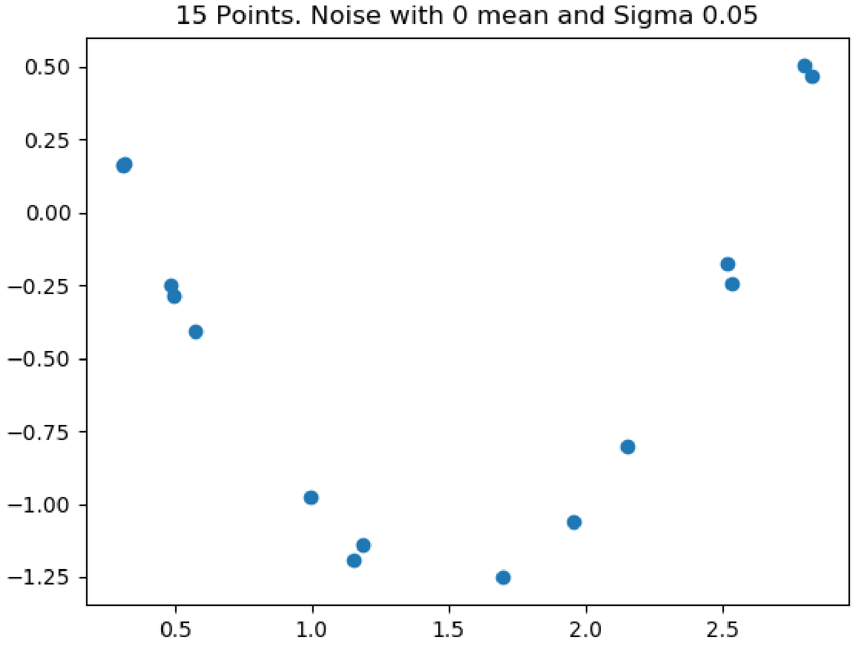
2. Write code in Python with file name `prob2_2.py` that randomly generates N points sampled uniformly in the interval $x \in [-1, 3]$. Then output the function $y = x^2 - 3x + 1$ for each of the points generated. Then write code that adds zero-mean Gaussian noise with standard deviation σ to y. Make plots of x and y with $N \in \{15, 100\}$ and $\sigma \in \{0, .05, .2\}$ (there should be six plots in total). Save the point sets for the following question.

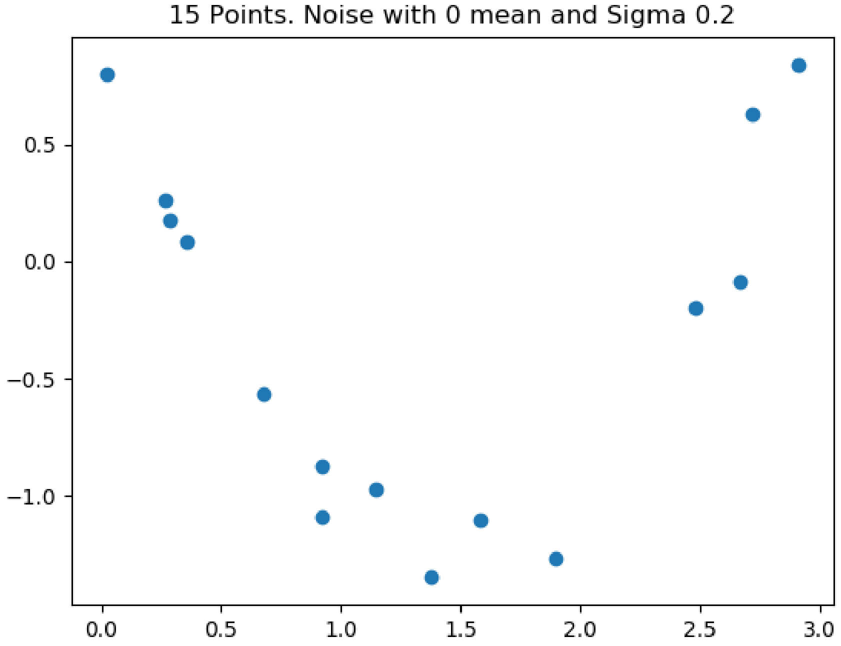
   Hint: You may want to check the NumPy library for generating noise.

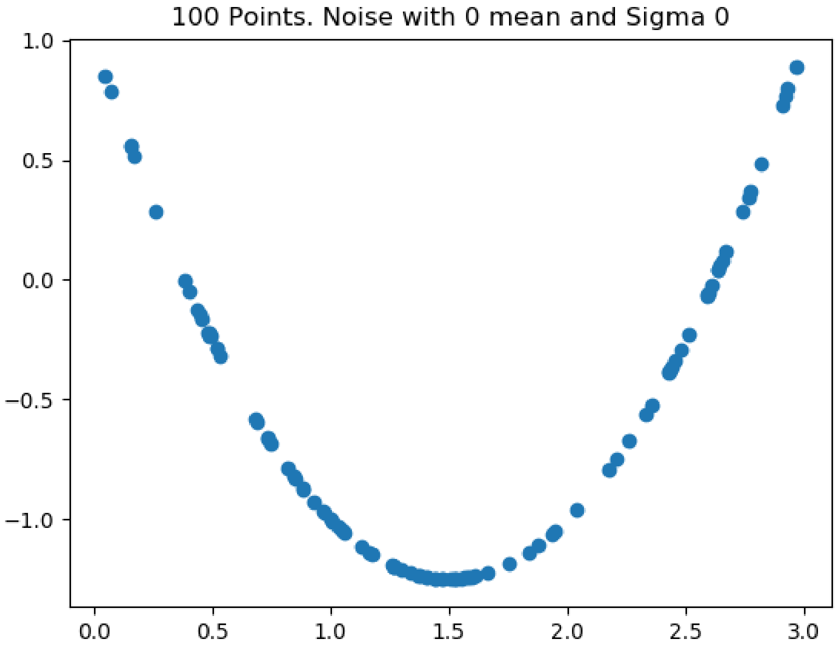
   Ans:

   I've included the point data in the 'data.txt' file.

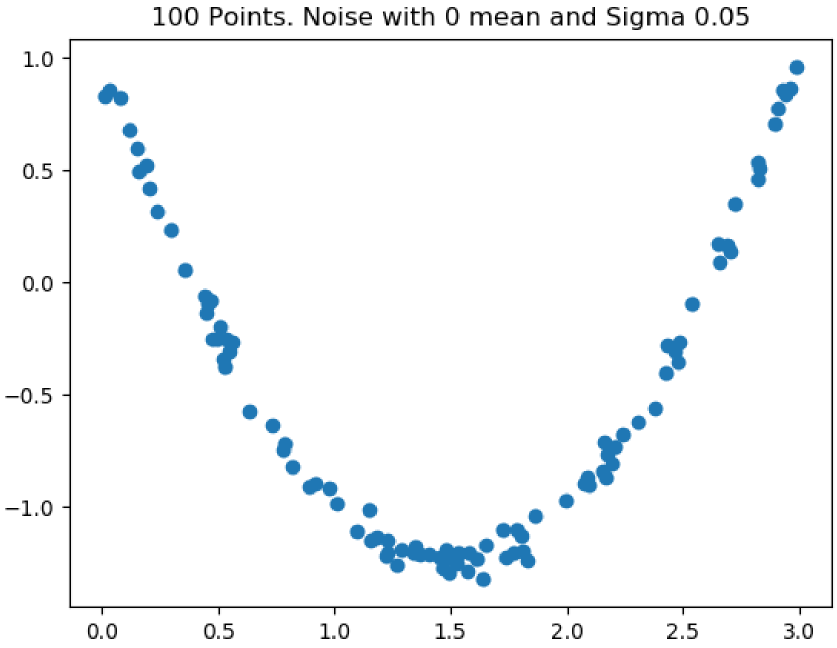



15 Points. Noise with 0 mean and Sigma 0.05
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
0.5
1.0
1.5
2.0
2.5

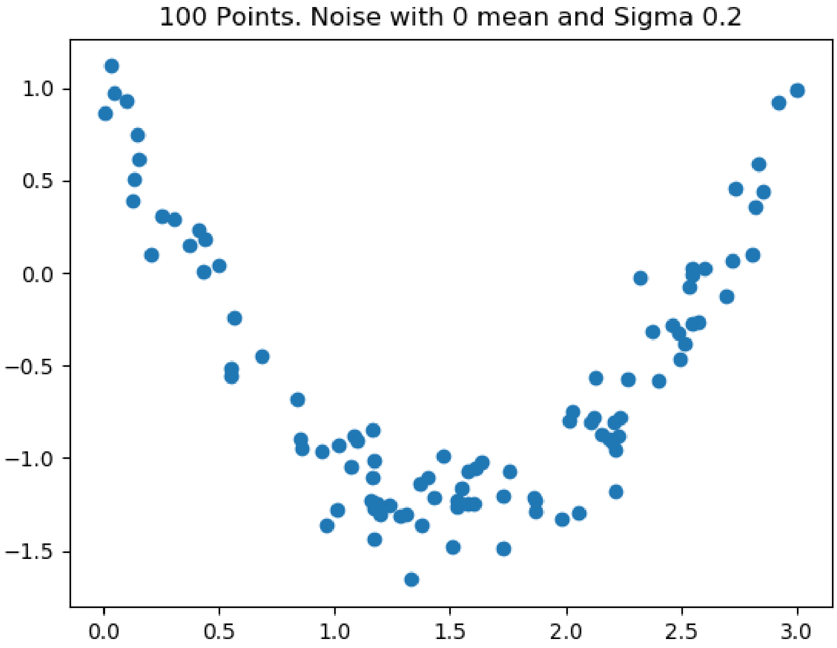

15 Points. Noise with 0 mean and Sigma 0.2
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0



100 Points. Noise with 0 mean and Sigma 0



100 Points. Noise with 0 mean and Sigma 0.05

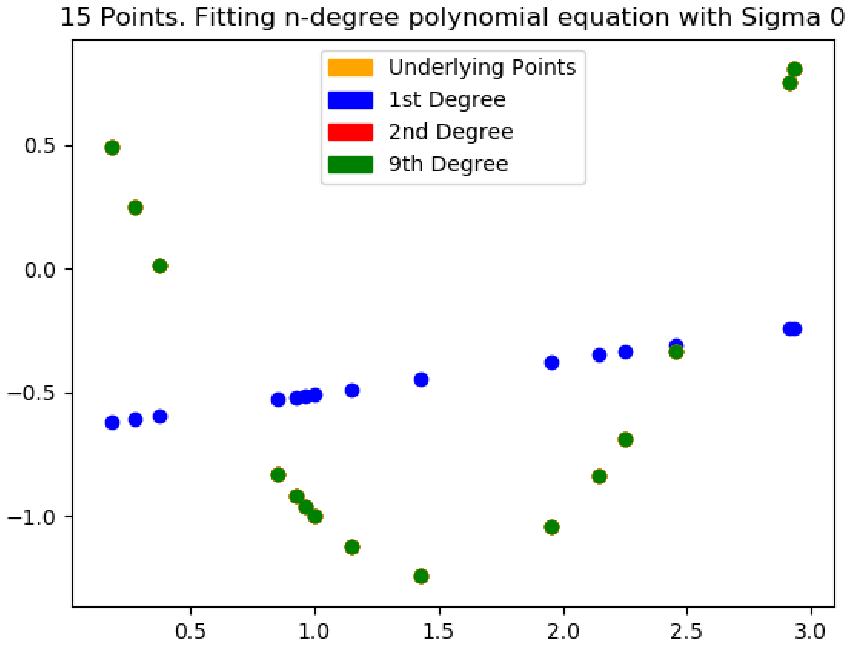


3. Find the optimal weights (in terms of MSE) for fitting a polynomial function to the data in all 6 cases generated above using a polynomial of degree 1, 2, and 9. Use the equation given above. Include your code in `prob2_3.py`. Do not use built-in methods for regression. Plot the fitted curves on the same plot as the data points (you can plot all 3 polynomial curves on the same plot). Report the fitted weights and the MSE in tables. Do any of the models overfit or underfit the data?

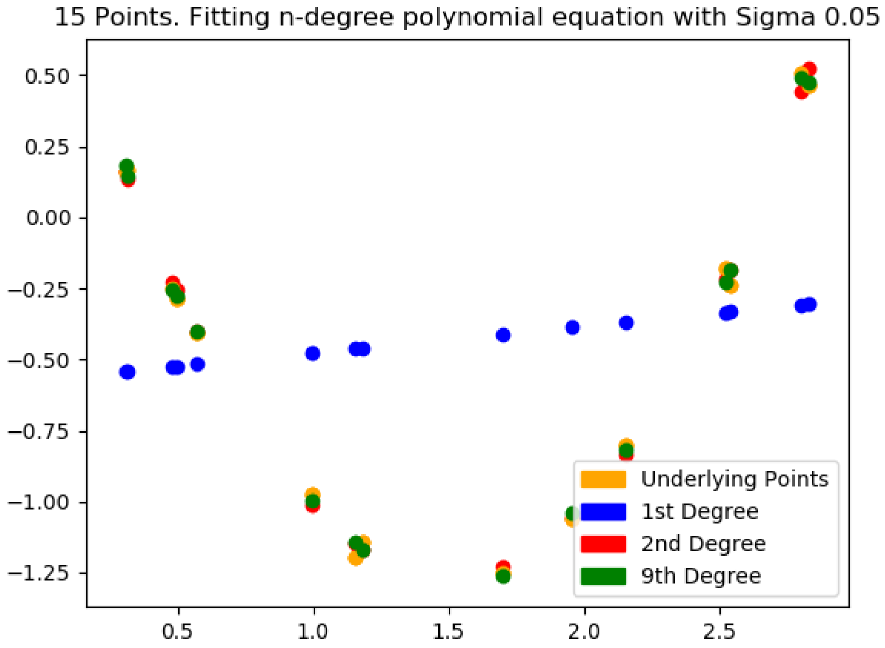
   Ans:

   The 1st degree polynomial underfits the data in all models.
   The 9th degree polynomial very slightly overfits the data in figure 3. (15 points, Sigma 0.2)
   The rest don't over or underfit.

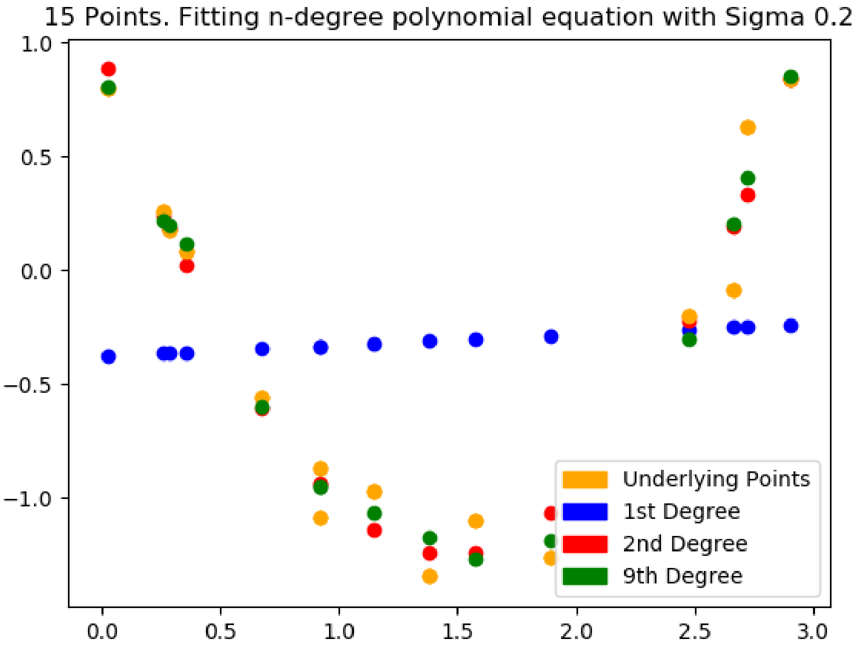
   I've included the MSE and weights in the 'MSEandweights.txt' file.

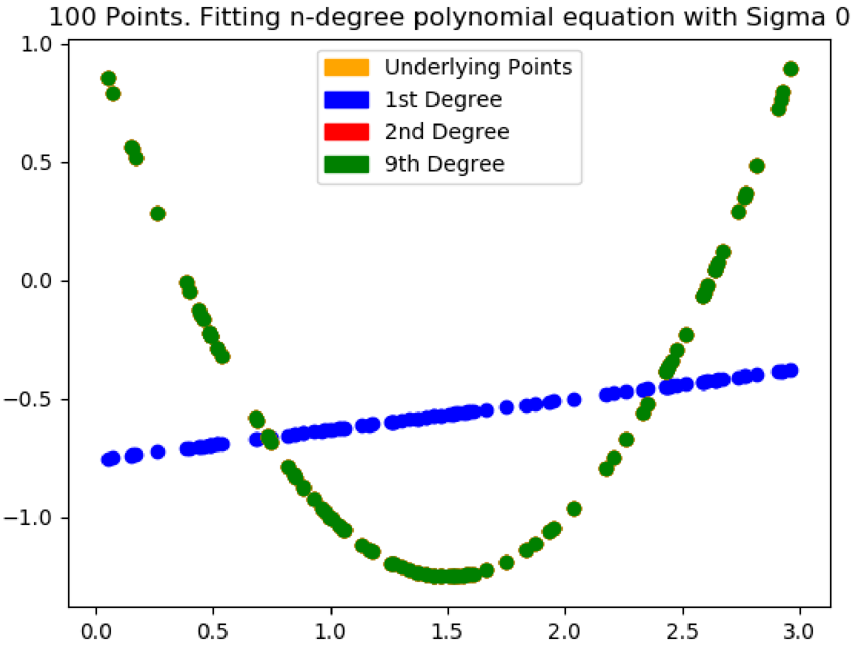


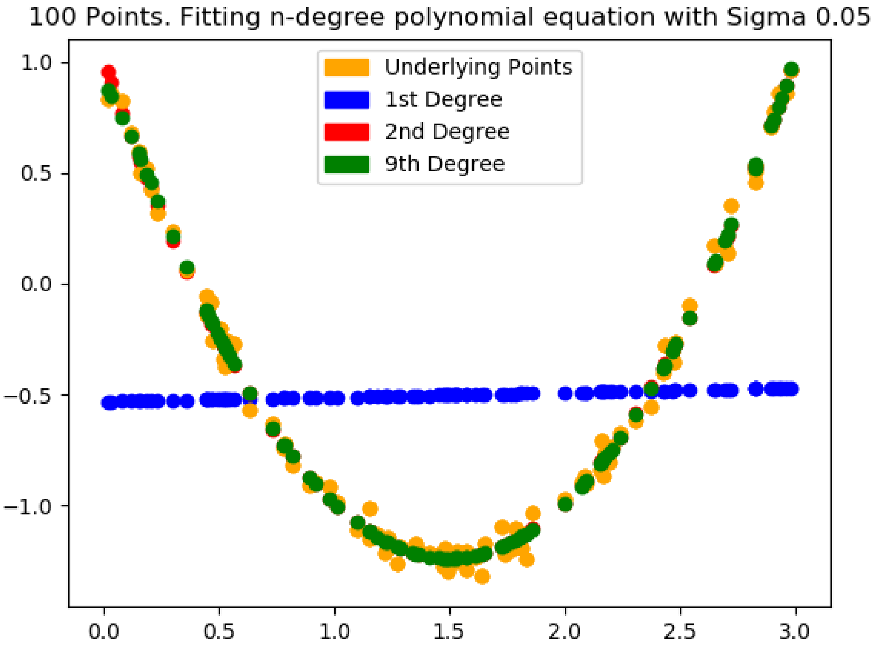
15 Points. Fitting n-degree polynomial equation with Sigma 0

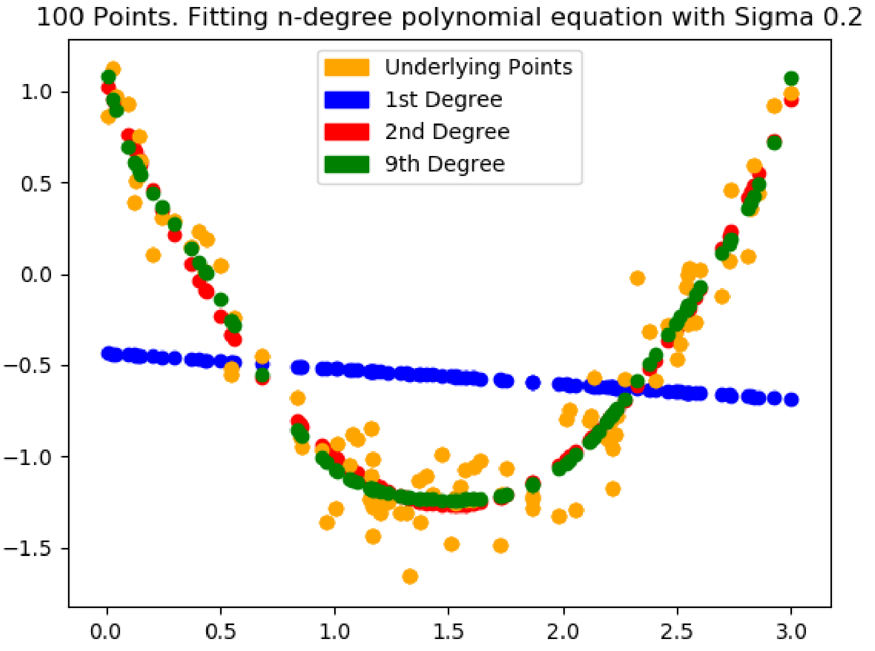


15 Points. Fitting n-degree polynomial equation with Sigma 0.05


15 Points. Fitting n-degree polynomial equation with Sigma 0.2
1.0
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Underlying Points
1st Degree
2nd Degree
9th Degree


100 Points. Fitting n-degree polynomial equation with Sigma 0
1.0
0.5
0.0
−0.5
−1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Underlying Points
1st Degree
2nd Degree
9th Degree





Include your answers and plots in the same PDF titled `<lastname and initials>_assignment1.pdf`. Include all your code in `<lastname and initials>_assignment1.zip`.