**Due:** $17^{th}$ **Feb 2021 2:00 PM**

# 1 Introduction

This homework is designed to test your understanding of mathematics tutorial discussed in this link and in the lectures. The task is to visualize the principal components of a dataset and to fit polynomials to data using different linear least square techniques discussed in the tutorials:

- Line fitting using Linear Least Squares
- Reduce overfitting using regularization (Ridge Regression)

# 2 Data

The 2D point data is provided in the form of a .csv files (click here to download). A visualization of the datasets is shown in Figure 1.

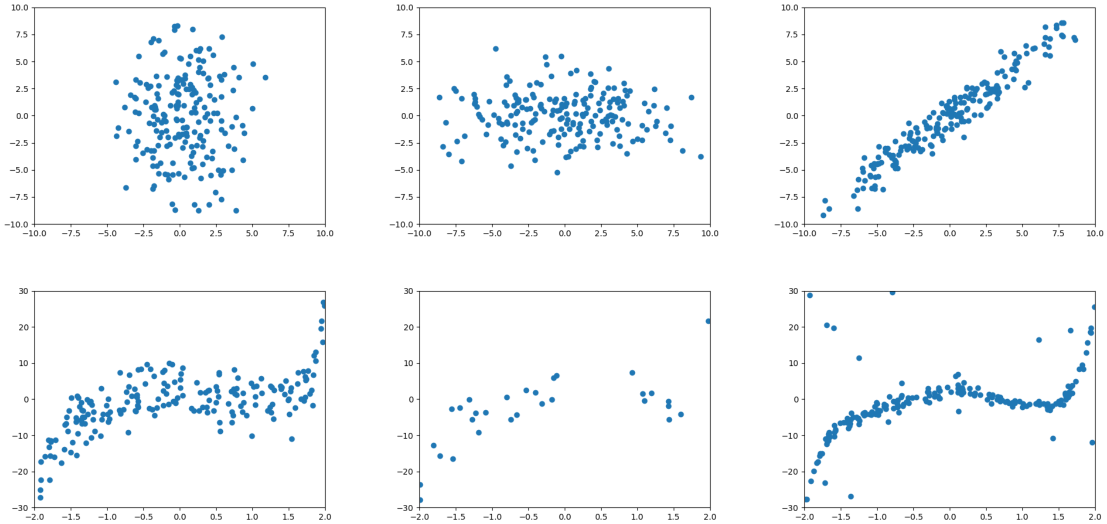

Figure 1: Visualization of the datasets.

# 3 Restricted functions

With the exception of the extra credit section where they are allowed, do not use line fitting functions (e.g., sklearn.linear_model.Ridge) to directly solve the assigned problems. Standard linear algebra functions (e.g, np.linalg.inv()) are permitted.

# 4 Helper functions

The following function will form a lifted data matrix, as described in slide 21 of Lecture 3. You will use this data matrix to fit polynomials to the datasets.

```
def lift(x,n_poly=3):
    N = np.size(x, 0)
    X = np.zeros([N,n_poly+1])
    for p in range(n_poly+1):
        X[:,p]=x**p
    return X
```

# 5 Instructions

## 5.1 Plotting eigenvalues

**40 points**

Write Python code to visualize geometric interpretations of the eigenvalues/covariance matrices associated with the files Lindata1.csv, Lindata2.csv, and Lindata3.csv datasets, as discussed in this link. The matplotlib.pyplot quiver command will be useful for this purpose.

Note that the lecture slides described an unscaled covariance matrix (missing $\frac{1}{n-1}$ term). To get the proper scaling of the principal directions (and info on how to compute it using the SVD), please see the notes here.

# 6 Least squares fitting

**40 points**

We will now fit polynomials to the data in the files Nonlindata1.csv, Nonlindata2.csv, and Nonlindata3.csv. Treat the first variable in the dataset as the independent variable (x) and the second as the dependent variable (y). If you plot the data, it should look like the second row of Figure 1.

- Fit a $12^{th}$ order polynomial to the datasets using linear least squares.
- Fit a $12^{th}$ order polynomial to the datasets using ridge regression (with varying amounts of regularization).
- For both models, plot the predictions associated with the datapoints `x_test=np.arange(-2,2,.01)`.

## 6.1 Report

**20 points**

For each section of the homework, explain briefly what you did and describe any interesting problems you encountered and/or solutions you implemented. You must include the following details in your writeup:

- Describe what the eigenvectors and eigenvalues of a covariance matrix represent
- Describe how one reduces overfitting
- Describe the limitations of ridge regression

Your report must be full English sentences, **not** commented code. There is a word limit of 750 words and no minimum length requirement

## 6.2 *Extra Credit*: Robust Regression

**10 points**

Use sklearn's Huber robust regression model (from sklearn.linear_model import HuberRegression) and fit it to the dataset and plot the fitted lines. Discuss how and why Huber regression works in your report.

# Submission Instructions

Your canvas submission should consist of a zip file named **YourDirectoryID_hw2.zip**, for example xyz123_hw2.zip. The file must have the following directory structure

- data/
- plot_eig.py or .ipynb
- least_square.py or .ipynb
- report.pdf

# Collaboration Policy

You are encouraged to discuss the ideas with your peers. However, the code should be your own and should represent your understanding of the assignment. If you reference anyone else's code in writing your project, you must properly cite it in your code (in comments) and in your writeup.