ECE 595: Homework 1

Manish Nagaraj, PUID: 0029904105

(Spring 2021)

Exercise 1: Histogram and Cross-Validation

(a) Plot of $f_X(x)$

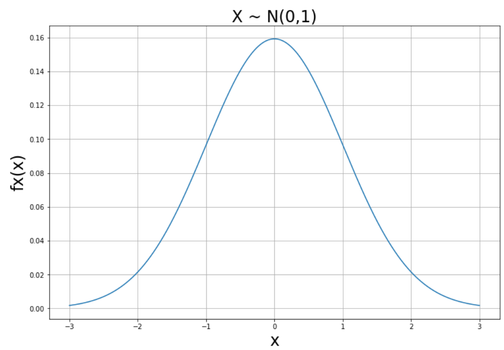
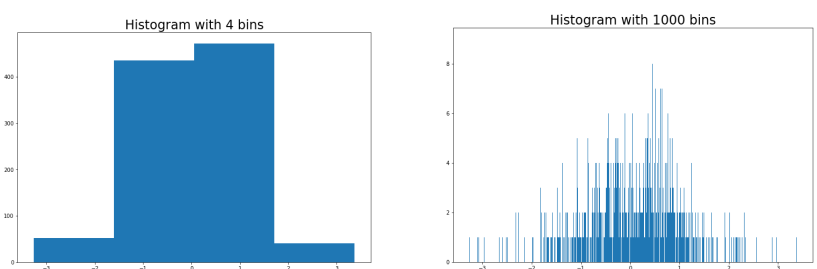


Figure 1: Plot of $f_X(x)$

(b)

i. 1000 random samples drawn.

ii. Histograms with different number of bins is represented in Figure.2



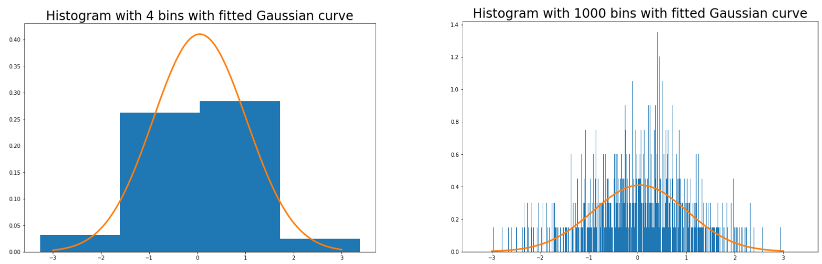
(a) 4 bins (b) 1000 bins

Figure 2: Histograms with different bins

iii. The mean and standard deviation from the data are: $\mu = -0.03408, \sigma = 0.9953$

iv. Figure.3 represents the histograms overlayed with the fitted Gaussian curve.

v. No, the histograms are not good representations of the data. The overlapping of the histograms with the fitted Gaussian curve shows the shortcomings of the bin sizes used.



(a) 4 bins (b) 1000 bins

Figure 3: Histograms with different bins

(c)

i. Plot of risk $\hat{J}(h)$ vs number of bins m in Figure.4

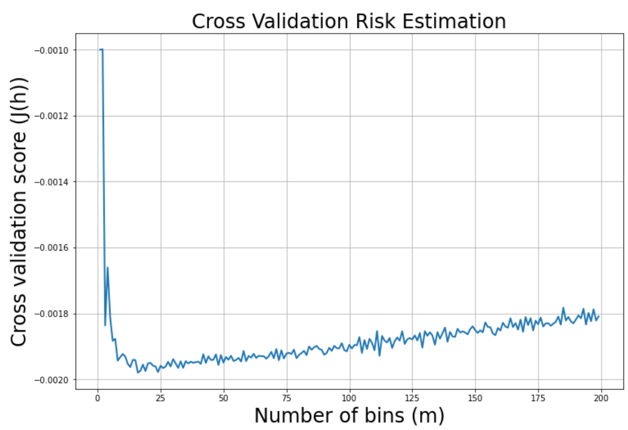
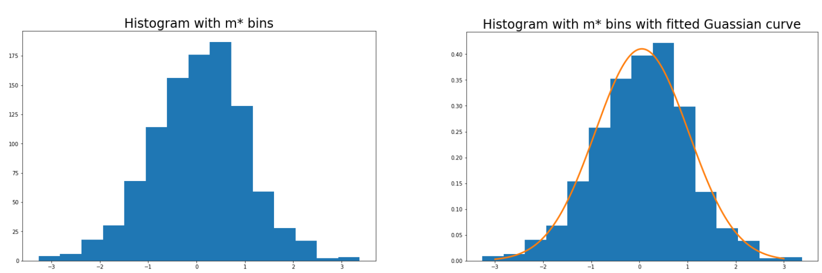


Figure 4: Plot of risk $\hat{J}(h)$ vs number of bins m

ii. The m^* that minimizes the loss is 15. Figure.5(a) plots the histogram with 15 bins.

iii. The histogram is overlayed with the fitted curve in Figure. 5(b).



(a) 15 bins (b) Fitted curve overlayed

Figure 5: Histograms with $m^* = 15$ bins

Exercise 2: Gaussian Whitening

(a) Solving for $f_X(x)$:

(i)

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mu = \begin{bmatrix} 2 \\ 6 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad \therefore \Sigma^{-1} = \frac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \qquad \text{and } |\Sigma| = 3$$

Substituting these values in

$$f_X(x) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}} \exp\left\{ -\frac{1}{2}(x-\mu)^T \sigma^{-1}(x-\mu)\right\}$$

$$= \frac{1}{\sqrt{12\pi^2}} \exp\left(\frac{-1}{6} \begin{bmatrix} x_1 - 2 \\ x_2 - 6 \end{bmatrix}^T \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 - 2 \\ x_2 - 6 \end{bmatrix} \right)$$

$$= \frac{1}{\sqrt{12\pi^2}} \exp\left(\frac{-1}{6} \begin{bmatrix} x_1 - 2 \\ x_2 - 6 \end{bmatrix}^T \begin{bmatrix} 2x_1 - x_2 + 2 \\ -x_1 + 2x_2 - 10 \end{bmatrix} \right)$$

$$= \frac{1}{\sqrt{12\pi^2}} \exp\left(\frac{-1}{6} \left(2x_1^2 + 2x_2^2 + 4x_1 - 20x_2 - 2x_1x_2 + 56\right) \right)$$

$$\boxed{\therefore f_X(x) = \frac{1}{\sqrt{12\pi^2}} \exp\left(\frac{-1}{3}(x_1^2 + x_2^2 + 2x_1 - 10x_2 - x_1x_2 + 28) \right)}$$

(ii) The plot of $f_X(x)$ is shown in Figure.6

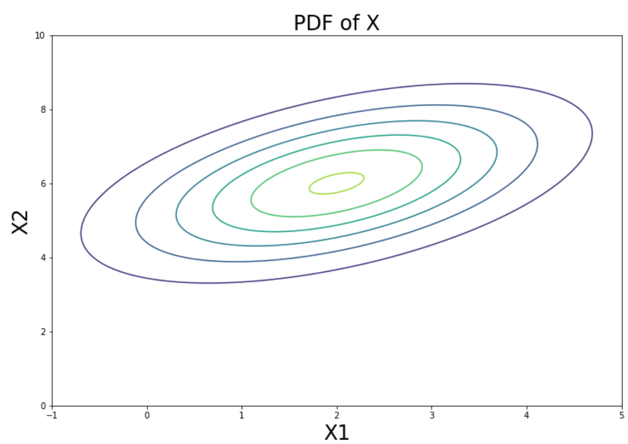


Figure 6: Contour plot of $f_X(x)$

(b) (i) Given, $Y = AX + b$ and $X \sim \mathcal{N}(0, I)$

$$\therefore \mu_Y = \mathbf{E}[Y] = \mathbf{E}[AX + b] = A\mathbf{E}[X] + b = \mathbf{b}$$

And,

$$\therefore \Sigma_Y = \mathbf{E}[(Y - \mu_Y)(Y - \mu_y)^T] = \mathbf{E}[(AX)(AX)^T] = A\mathbf{E}[XX^T]A^T = AIA^T = \mathbf{AA^T}$$

(ii) We need to show that Σ_Y is symmetric positive semi-definite.
Since $\Sigma_Y = \Sigma_Y^T$, Σ_Y is symmetric.

Consider any arbitrary $x \in \mathbf{R}^d$,

$$x^T \Sigma_Y x = x^T AA^T x = (A^T x)^T (A^T x) = \|Ax\|^2 \geq 0$$

Since this is true for any x, Σ_Y is positive semi-definite.

$$\boxed{\text{Hence } \Sigma_Y \text{is symmetric positive semi-definite.}}$$

(iii) For Σ_Y to be positive definite, then $\forall x \in \mathbf{R}^d, x^T \Sigma_Y x > 0$.
Since Σ_Y is positive semi definite, Σ_Y will be positive definite iff,

$$\nexists x \in \mathbf{R}^d, \quad x^T \Sigma_Y x = 0$$

.

$$x^T \Sigma_Y x = 0 \implies x^T AA^T x = (A^T x)^T (A^T x) = 0 \implies A^T x = 0$$

Hence if the null space of A^T or the column space of A is not empty, Σ_Y can not be positive definite. **For Σ_Y to be positive definite, A must be a full rank matrix, i.e., columns must be linearly independent.**

(iv)

$$\because \mu_y = b, \quad b = \begin{bmatrix} 2 \\ 6 \end{bmatrix} \qquad\qquad \Sigma_y = AA^T = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Let us consider the eigen value decomposition of $\Sigma = U\Lambda U^T$. Then

$$\Sigma_Y = AA^T = U\Lambda U^T = U\big(\Lambda^{1/2}(\Lambda^{1/2})^T\big)U^T \implies A = U\Lambda^{1/2}$$

By decomposing Σ_Y, we get

$$\Sigma_Y = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Since Λ is a diagonal matrix, $\Lambda^{1/2}$ will be a diagonal matrix with $\sqrt{3}, 1$.

$$\therefore A = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \sqrt{\frac{3}{2}} & \sqrt{\frac{1}{2}} \\ \sqrt{\frac{3}{2}} & -\sqrt{\frac{1}{2}} \end{bmatrix}$$

(c) (i) The plot of random samples drawn from a standard 2D normal distribution is shown in Figure.7

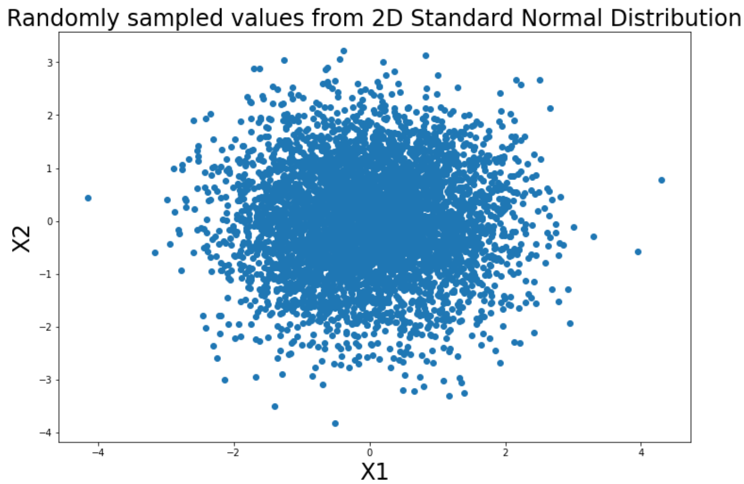
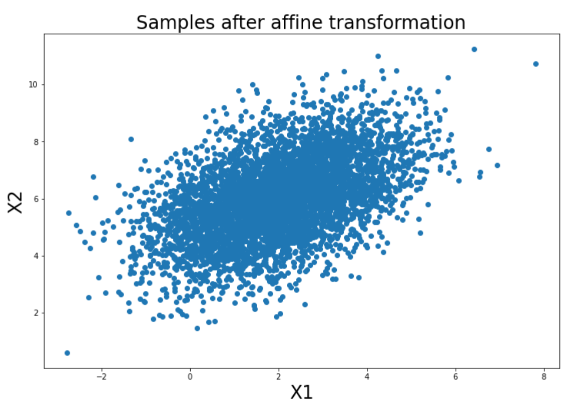
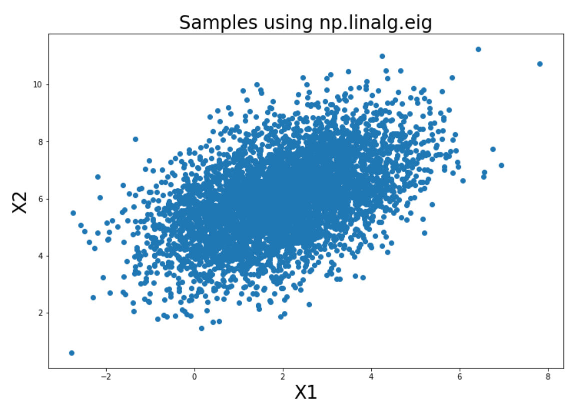


Figure 7: Random sampling from multivariate Normal distribution

(ii) The randomly sampled points transformed after using the affine trandormation derived and the points transformed using Python's numpy.linalg.eig are plotted in Figure.8



(a) Using derived Affine Transformation



(b) Using numpy.linalg library

Figure 8: Transformed data points

(iii) As we can see from the Figure.8, both the plots depict the same distribution of points, hence the theoretically derived transformation is correct.

Exercise 3: Linear Regression

(a) Figure.9 shows the scatter plot of the data.

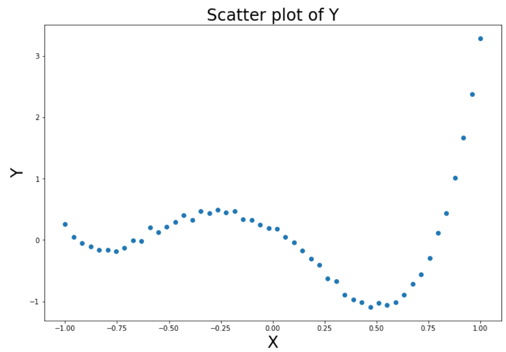


Figure 9: Scatter plot of the data

(b) The problem can be written down as: $\hat{\beta} = \arg\min_\beta \|y - X\beta\|^2$, where

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{50} \end{bmatrix}, \quad X = \begin{bmatrix} L_4(x_1) & L_3(x_1) & L_2(x_1) & L_1(x_1) & 1 \\ L_4(x_2) & L_3(x_2) & L_2(x_2) & L_1(x_2) & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ L_4(x_{50}) & L_3(x_{50}) & L_2(x_{50}) & L_1(x_{50}) & 1 \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_4 \\ \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \end{bmatrix}$$

The solution of this can be found when $\nabla_\beta \|y - X\beta\|^2 = 0$.

$$\begin{aligned} \nabla_\beta \|y - X\beta\|^2 &= -2X^T(y - X\beta) = 0 \\ \implies X^T y - X^T X\hat{\beta} &= 0 \\ & \boxed{\therefore \hat{\beta} = (X^T X)^{-1} X^T y} \end{aligned}$$

(c) Figure.10 shows the solution to the regression problem overlayed on the data.

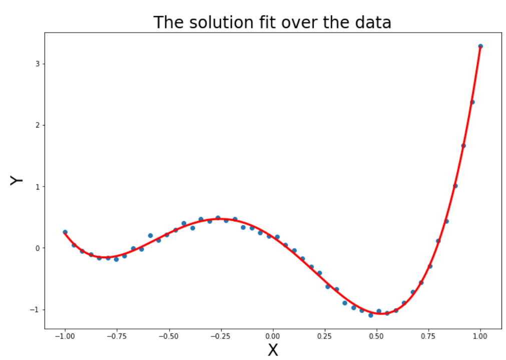


Figure 10: Solution of the regression problem

(d) Figure.11 shows the solution overlayed on the data with outliers. With the outliers present, the solution to the regression problem does not fit very well.

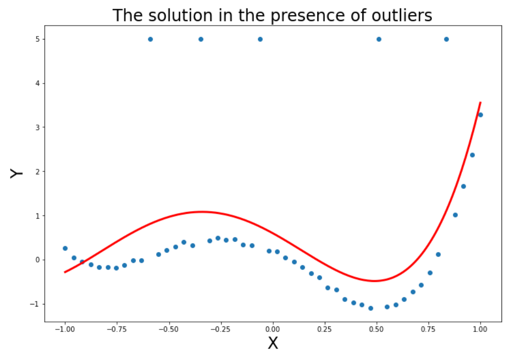


Figure 11: Solution of the regression problem with outliers

(e) The optimization $\hat{\beta} = \arg\min_\beta \|y - X\beta\|_1$ can be converted to linear programming as follows,

$$\hat{\beta} = \arg\min_{\beta} \|y - X\beta\|_1 \equiv \min \sum_{n=1}^{50} |y_n - X_n^T \beta|$$

This is equivalent to the constrained optimization,

$$\min_{\beta, u_n} \sum_{n=1}^{50} u_n, \qquad \text{s.t.} \quad u_n = |y_n - X_n^T \beta|$$

Which is equivalent to

$$\begin{aligned} \underset{\beta, u_n}{\text{minimize}} \quad & \sum_{n=1}^{50} u_n \\ \text{subject to} \quad & u_n \geq (y_n - X_n^T \beta) \\ & u_n \geq -(y_n - X_n^T \beta) \end{aligned}$$

Hence,

$$\boxed{c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \qquad x = \begin{bmatrix} \beta_4 \\ \beta_3 \\ \beta_2 \\ \beta_1 \\ \beta_0 \\ u_1 \\ \vdots \\ u_{50} \end{bmatrix} \qquad A = \begin{bmatrix} \phi_1^T & -1 & 0 & \dots & 0 \\ \phi_2^T & 0 & -1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \phi_{50}^T & 0 & 0 & \dots & -1 \\ -\phi_1^T & -1 & 0 & \dots & 0 \\ -\phi_2^T & 0 & -1 & \dots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ -\phi_{50}^T & 0 & 0 & \dots & -1 \end{bmatrix} \qquad b = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{50} \\ -y_1 \\ -y_2 \\ \vdots \\ -y_{50} \end{bmatrix}}$$

where,

$$\phi_n^T = \begin{bmatrix} L_4(x_n) & L_3(x_n) & L_2(x_n) & L_1(x_n) & L_0(x_n) \end{bmatrix}$$

(f) The solution to the linear programming problem is plotted in Figure.12

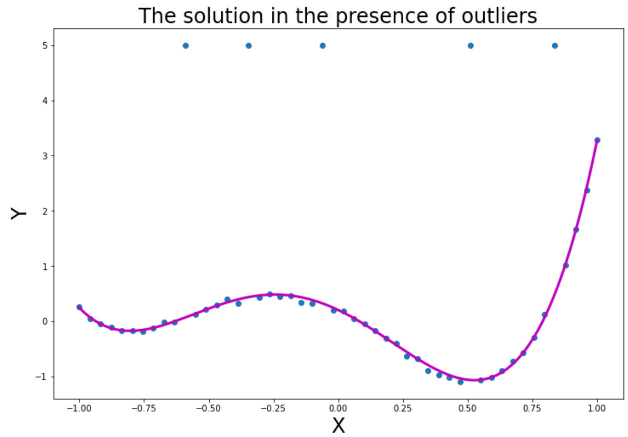


Figure 12: Solution to the linear programming optimization problem

Exercise 4: Project: Check Point 1

The one page PDF with the details is attached below.

ECE595 Project - Exploring Limitations of Learning to Reweight Examples for Robust Deep Learning - Checkpoint 1

Manish Nagaraj [1]

Abstract

This is checkpoint 1 of the project.

1. Author Details

- **Name:** Manish Nagaraj
- **Major:** Electrical and Computer Engineering
- **Level:** Ph.D.

2. Details of Project

This project aims to study the limitations of the methods proposed in the paper referenced in the title (Ren et al., 2018b). Further, it also aims in identifying and implementing any algorithms or techniques that will overcome these limitations.

2.1. Datasets

For the purpose of evaluating existing algorithms as well as potential methods proposed, we will be using **CIFAR10**, **MNIST**, **FMNIST** datasets.

2.2. Reference Codes

We will be using the following code sources available as a reference to help implement the project.

1. `https://github.com/danieltan07/learning-to-reweight-examples`
2. Learning to Learn by Gradient Descent

Some of the papers that we will be using for reference are cited in the References.

[1]Department of Electrical and Computer Engineering, Purdue University, USA. Correspondence to: Manish Nagaraj <mnagara@purdue.edu>.

Proceedings of the 38^{th} *International Conference on Machine Learning*, PMLR 139, 2021.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474*, 2016.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. 2016.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018a.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018b.