Module-3

Antara Paul, Labani Roy

Activity 1: Binary classification

1.1 Least Squares

For two classes, each class ($\mathcal{C}_1$ and $\mathcal{C}_2$) is described by its own linear model :

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T x + w_{k0} \tag{1}$$

where $\boldsymbol{w}$ is the weight vector and k = 1,2. Grouping in the bias terms the vector notation can be written as

$$\boldsymbol{y}(\boldsymbol{x}) = \tilde{\boldsymbol{W}}^T \tilde{\boldsymbol{x}} \tag{2}$$

We can find $\tilde{\boldsymbol{W}}$ by finding the sum-of-squares error function and setting its derivative with respect to $\tilde{\boldsymbol{W}}$ as zero, which gives us

$$\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{X}}^\dagger \boldsymbol{T} \tag{3}$$

where $\tilde{\boldsymbol{X}}^\dagger$ is the pseudo inverse-matrix of $\tilde{\boldsymbol{X}}$ (which is the matrix of the augmented input vector as described above) and $\boldsymbol{T}$ is the target matrix.

Finally, the decision boundary of our hyperplane can be visualised as in Fig. 1 by plotting

$$y(x_1, x_2) = w[0]x_1 + w[1]x_2 + w[2] \tag{4}$$

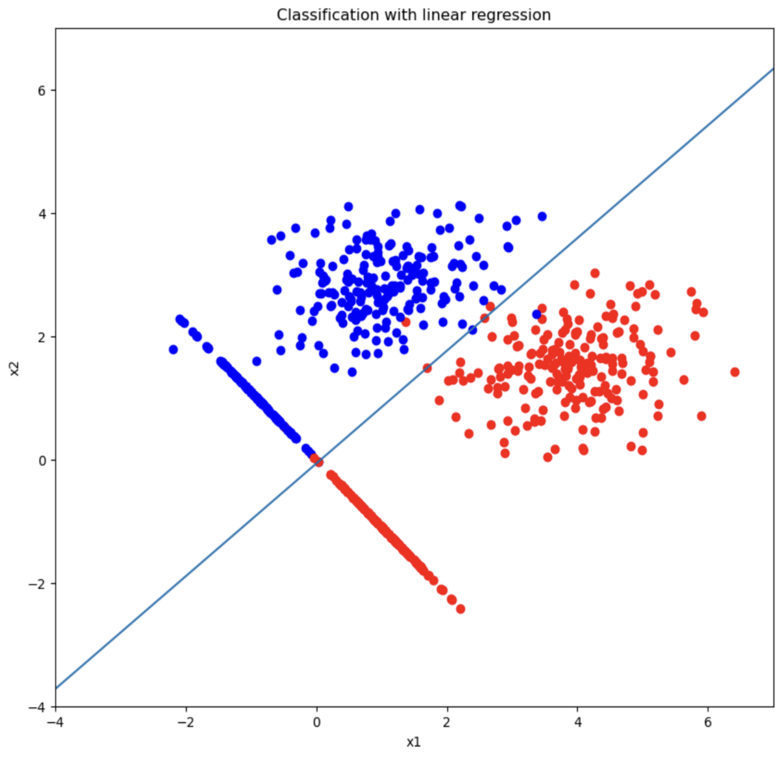

Figure 1: Classification with linear regression

1.2 Fisher Discriminant

The Fisher discriminant vector is

$$\mathbf{w} \propto \mathbf{S}_w^{-1}(\mathbf{m2} - \mathbf{m1}) \tag{5}$$

where $\mathbf{S}_w = \sum_{n \in C_1}(\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in C_2}(\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T$ is the *within-class* covariance matrix. The mean vectors of the two classes are given as $\mathbf{m}_1 = \frac{1}{N_1}\sum_{n \in C_1}\mathbf{x}_n$ and $\mathbf{m}_2 = \frac{1}{N_2}\sum_{n \in C_2}\mathbf{x}_n$.

The discriminant $\boldsymbol{w}$ here isn't exactly a discriminant but a specific choice of direction for projection of the data down into one dimension. Choosing a threshold y_0 to classify the two classes $\mathcal{C}_\infty$ (eg $y(x) > y_0$) and $\mathcal{C}_\in$ for otherwise, can help to construct the actual discriminant. The threshold can be best chosen to be the midpoint of the projected class means, as that'd give the best separation. So our threshold chosen for this was $y_0 = \frac{\boldsymbol{w}^T\boldsymbol{m}_1 + \boldsymbol{w}^T\boldsymbol{m}_2}{2}$.
Plotting of the decision boundary is done similarly as in the previous one.

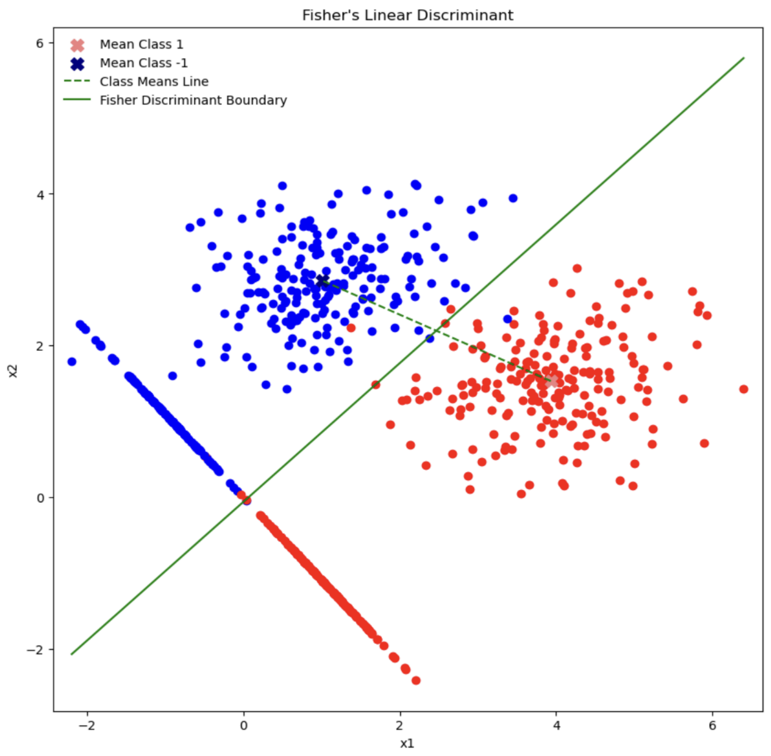

Figure 2: Classification with Fisher discriminant

1.3 Perceptron algorithm

The perceptron algorithm continually updates the parameters of the weight vector $\boldsymbol{w}$, to achieve a $\boldsymbol{w}$ such that $\boldsymbol{w}^T\phi(x_n) > 0$ for $\mathcal{C}_1$ for example, and $\boldsymbol{w}^T\phi(x_n) < 0$ for $\mathcal{C}_2$. When a point x_n is misclassified, the quantity $-\boldsymbol{w}^T\phi_n t_n$ is minimized. Thus, the change in weight vector $\boldsymbol{w}$ is given by:

$$\boldsymbol{w}^{\tau+1} = \boldsymbol{w}^\tau + \eta\,\phi_n t_n \tag{6}$$

η is the learning parameter rate set equal to 1 and $\boldsymbol{\phi}(x_n)$ is the augmented input vector.

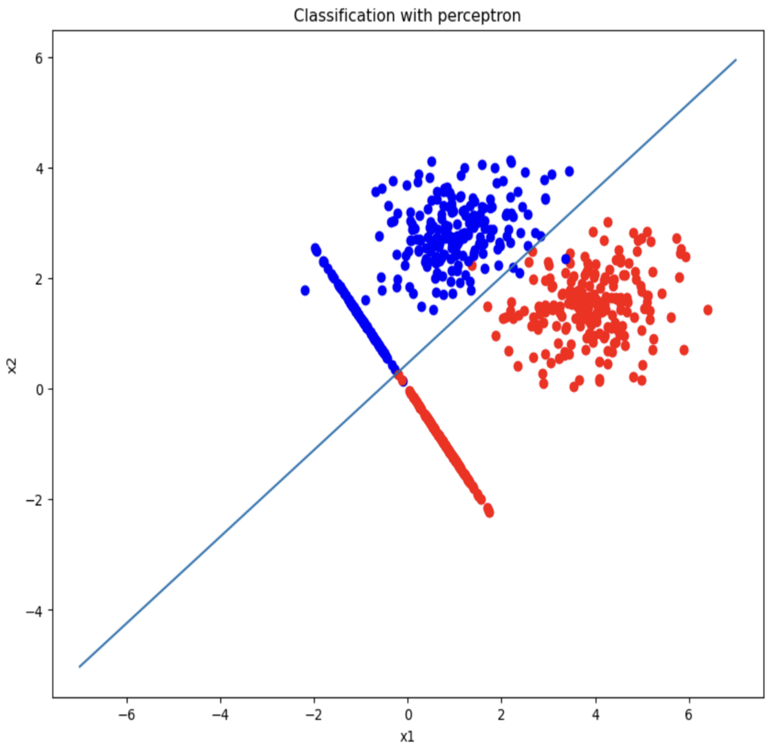

Figure 3: Classification with Perceptron algorithm

Activity 2: Binary classification with Ridge Regression

The python library `scikit-learn` is used for classification using the regularization method, which is “Ridge Regression”. Plot is shown below in Fig. 4, which on comparison was seen to have the same performance as linear regression and fisher discriminant.

Comparisons of all algorithms are shown in Fig. 5. Perceptron gives a slightly different performance (worse) than the other three. This is because perceptron works the best for classes that are linearly separable, whereas ours had some overlaps. For our case, the least squares and the fisher discriminant algorithms seem to converge.

Furthermore, when we added some outliers to our data, perceptron seemed to be the least sensitive to the outliers, whereas the other three were highly sensitive, as seen in Fig. 6.

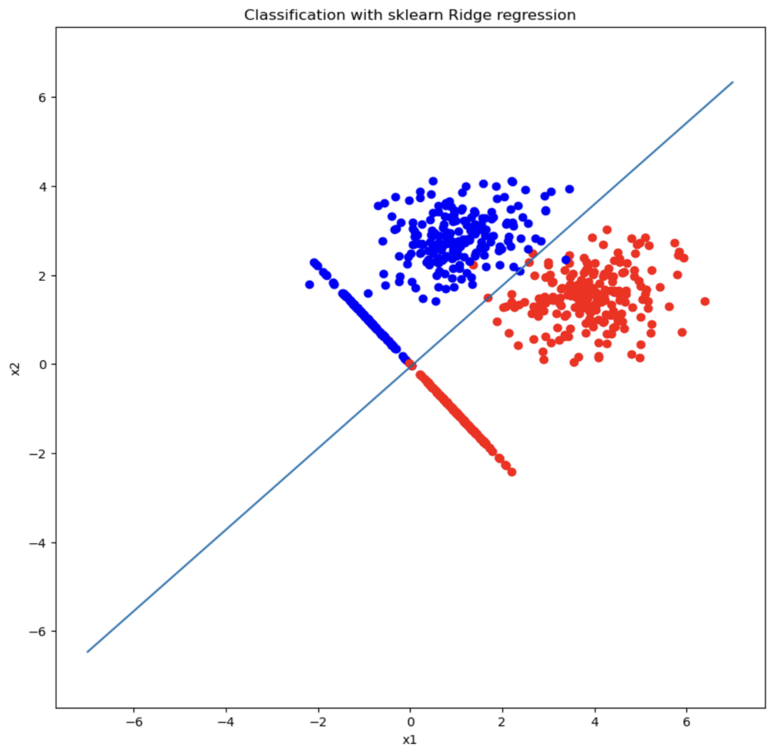

Figure 4: Classification with Ridge regression

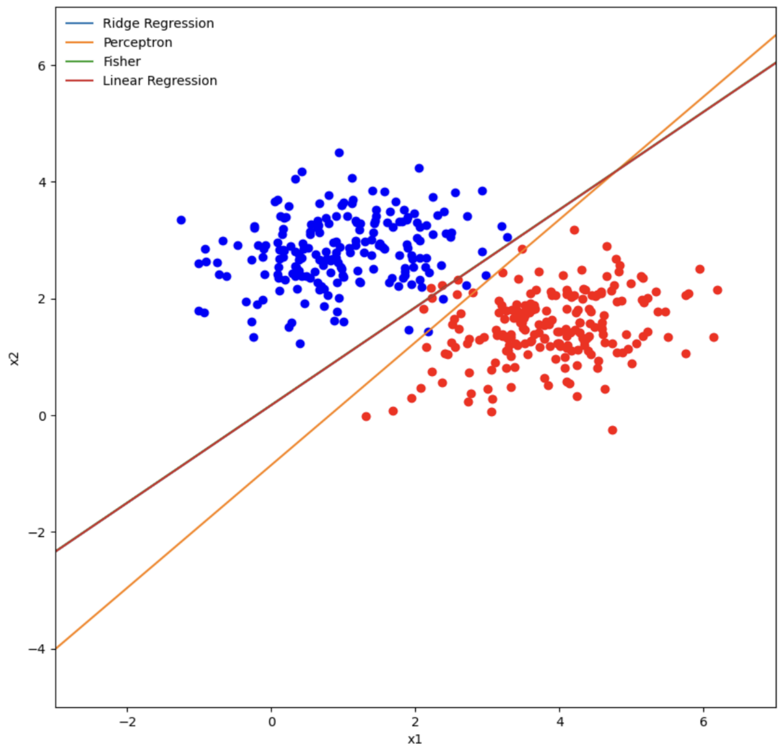

Figure 5: Comparison of all algorithms

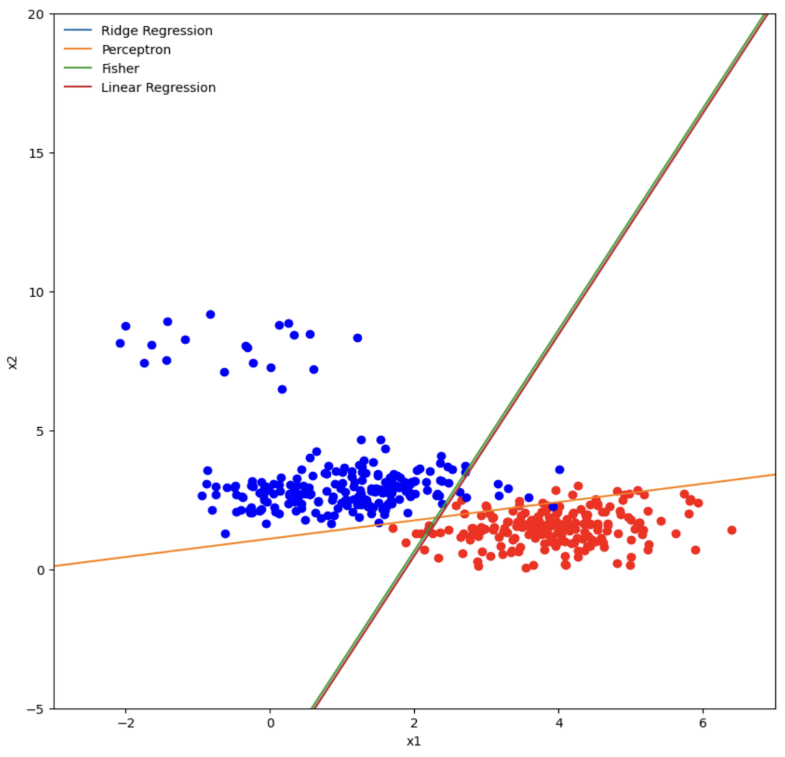

Figure 6: Comparison of all algorithms after adding outlier data