

ML-HW4

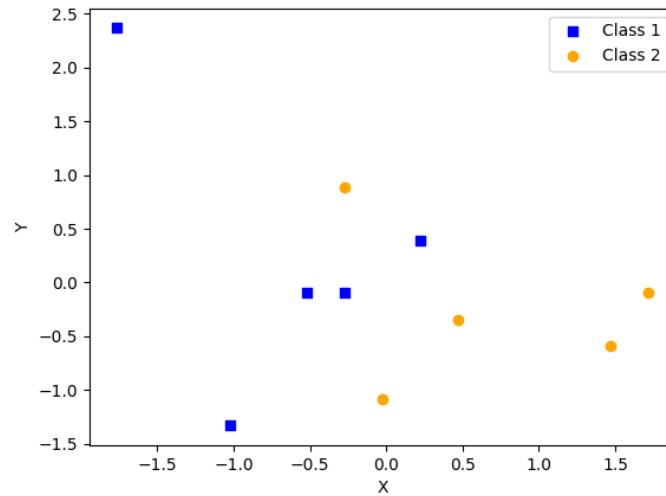
David A. Halloran

May 16, 2022

1 Theory

1. When we z score the original data this gives us:

$$X = \begin{bmatrix} -0.27422981 & -0.09865272 \\ -1.02212929 & -1.33181178 \\ -0.52352964 & -0.09865272 \\ 0.22436984 & 0.3946109 \\ -1.77002878 & 2.36766539 \\ -0.27422981 & 0.88787452 \\ 0.47366967 & -0.34528454 \\ 1.47086898 & -0.59191635 \\ -0.02492998 & -1.08517997 \\ 1.72016881 & -0.09865272 \end{bmatrix}$$



2. PCA

- (a) Since we already Z scored the data in section 1, we move onto the next step of getting the covariance matrix. We do this by using the formula $cov = \frac{X^T X}{N-1}$ which in our example gives us:

$$cov(X) = \begin{bmatrix} 1 & -0.40826218 \\ -0.40826218 & 1 \end{bmatrix}$$

After we get the covariance matrix we proceed to the next step: computing the eigenvalues and eigenvectors. We can do this easily with the `np.linalg.eig` function, which gives us:

$$Eigenvalues = [1.40826218 \quad 0.59173782]$$

$$Eigenvector_1 = \begin{bmatrix} 0.70710678 \\ -0.70710678 \end{bmatrix}$$

$$Eigenvector_2 = \begin{bmatrix} 0.70710678 \\ 0.70710678 \end{bmatrix}$$

We want to pick the maximum eigenvalue which gives us $Eigenvector_1$ to work with first. With this we will calculate each row of X by $eigenvector_1$ which will give us PCA_1 . So for the first feature of PCA1 it would look like:

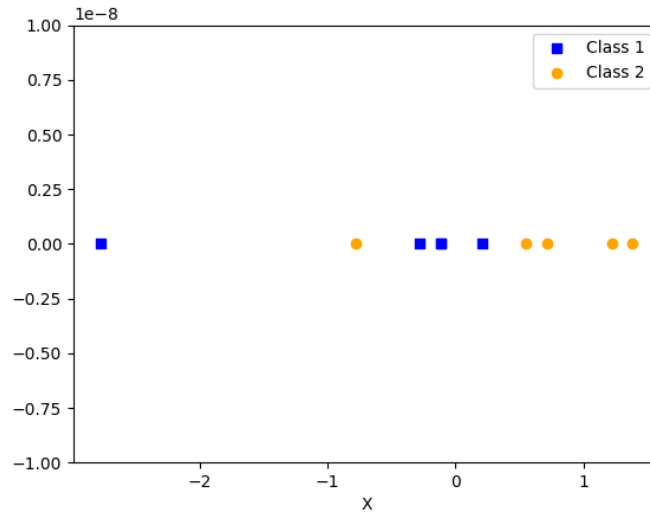
$$Z_1 = [-0.27422981, -0.09865272] * \begin{bmatrix} 0.70710678 \\ -0.70710678 \end{bmatrix} = -0.12415175$$

Following this through for the first eigenvector for the PCA_1 and the second eigenvector for PCA_2 this gives us:

$$PCA_1 = \begin{bmatrix} -0.11778069 \\ 0.20774133 \\ -0.2850161 \\ -0.11420117 \\ -2.77564963 \\ -0.77956329 \\ 0.54937118 \\ 1.38375847 \\ 0.71123731 \\ 1.22010258 \end{bmatrix}$$

$$PCA_2 = \begin{bmatrix} -0.25013721 \\ -1.57907168 \\ -0.41737262 \\ 0.41522491 \\ 0.40090683 \\ 0.41164539 \\ 0.08612337 \\ 0.58961935 \\ -0.7446844 \\ 1.08774606 \end{bmatrix}$$

(b) Graph:



(c) As you can see above, PCA doesn't provide that good of class separation. This is because PCA is used for variance maximization to determine which features contribute the most to a dataset. This doesn't really regard class labels in this calculation

3. LCA

(a) Since we have Z-Scored this data in part 1 we can skip to the second step of getting the different means for class 1 and class 2. This gives us:

$$\mu_1 = [-0.63856777, 0.23397548]$$

$$\mu_2 = [0.63856777, -0.23397548]$$

Next we calculate the scatter matrix for each of these classes. To get the scatter we use this formula for each class:

$$\sigma_i = (X_i - \mu_i)^T (X_i - \mu_i)$$

where X is a matrix and i is the current class being calculated. When fully calculated out this gives us:

$$\sigma_1 = \begin{bmatrix} 2.08079553 & -1.64903489 \\ -1.64903489 & 6.52554745 \end{bmatrix}$$

$$\sigma_2 = \begin{bmatrix} 2.84151647 & -0.53123272 \\ -0.53123272 & 1.9270073 \end{bmatrix}$$

With this info we can calculate the next steps, which are to get the Scatter Matrix between classes, within classes and the inverse of the within classes matrix. The scatter within matrix is simple enough as it is just adding the the above scatter matrices into one. The inverse matrix is just the inverse of the the Scatter Within Class matrix. The Scatter Between Classes is a little bit more work and looks like:

$$S_b = (\mu_1 - \mu_2)^T (\mu_1 - \mu_2)$$

Plugging the values calculated above for the μ and σ values gives us the results of:

$$S_w = \begin{bmatrix} 4.922312 & -2.18026761 \\ -2.18026761 & 8.45255474 \end{bmatrix}$$

$$S_w^{-1} = \begin{bmatrix} 0.22936149 & 0.05916193 \\ 0.05916193 & 0.13356776 \end{bmatrix}$$

$$S_b = \begin{bmatrix} 1.6310752 & -0.59763681 \\ -0.59763681 & 0.2189781 \end{bmatrix}$$

Next we have to perform eigen decomposition. We do this function on one matrix so first step is the matrix multiply the inverse matrix with the scatter between matrix. This gives us:

$$S_w^{-1} S_b = \begin{bmatrix} 0.3387485 & -0.1241197 \\ 0.01667254 & -0.00610893 \end{bmatrix}$$

With this info we can take the eigenvalue and eigenvectors of this matrix:

$$Eigenvalues = [.332639570 \quad 0]$$

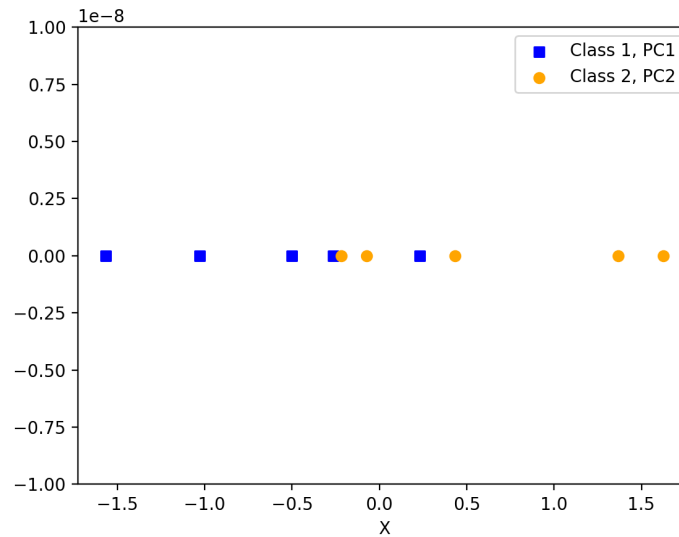
$$Eigenvectors_{s_1} = \begin{bmatrix} 0.99879099 \\ 0.04915855 \end{bmatrix}$$

$$Eigenvectors_{s_2} = \begin{bmatrix} 0.34403943 \\ 0.9389552 \end{bmatrix}$$

Since vector 2 has an eigenvalue of 0, we will only consider eigenvector 1 and set that for our "w" value. Finally we can get our answer with the formula $Z = XW$. Plugging in our X matrix and multiplied by eigenvector 1 this gives us the values of:

$$Z = \begin{bmatrix} -0.26444347 \\ -1.03061487 \\ -0.50066411 \\ 0.23100161 \\ -1.56674837 \\ -0.21843588 \\ 0.43271657 \\ 1.36609725 \\ -0.0742304 \\ 1.62532168 \end{bmatrix}$$

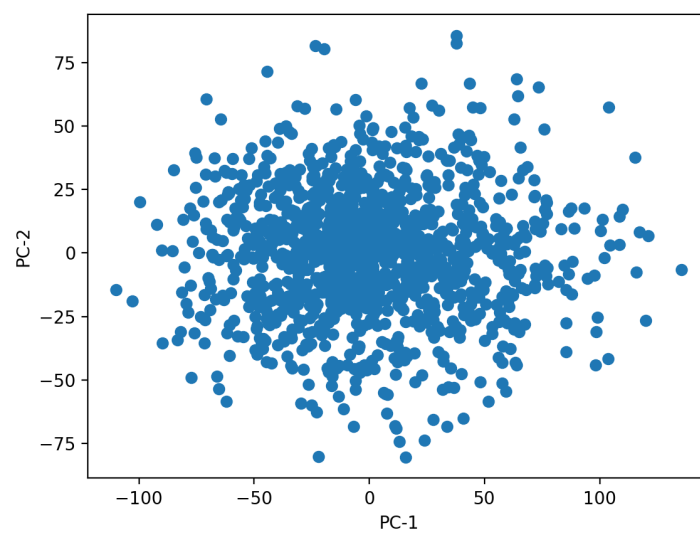
(b) Graph:



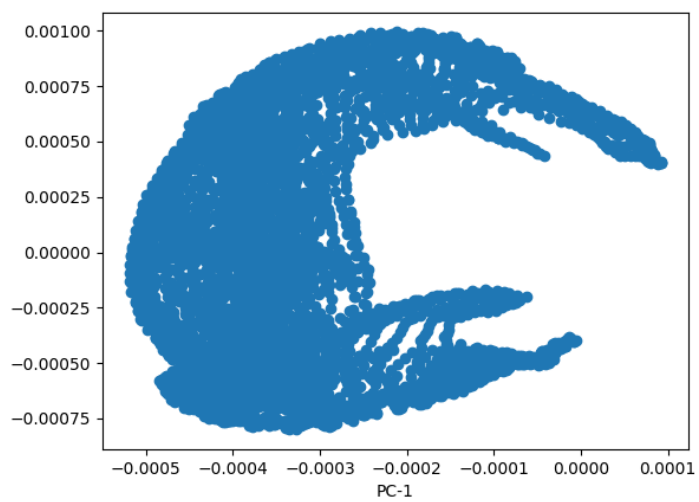
(c) LDA provides better class separation than PCA. This is cause, by design, it finds a projection line that maximizes seperability of the classes

2 Dimension Reduction

1. Non-Whitened PCA data:



2. Whitened PCA data:



3 KNN

All of these values have a $k = 1$

1. D = Original Size: 19.128329
2. D = 100 PCs: 17.917676
3. D = 100 PCs + Whitening: 10.411622

4 Eigen Compression

Out of the 5655 total eigenvectors, only the top 181 were needed to to recreate an image with 95% information

