COMP6245(2020/21): Foundations of Machine Learning Lab Five

Sameen Islam

I. Class Boundaries and Posterior Probabilities

Generative models solve posterior class probabilities $p(C_k|x)$ directly, but have to first determine the likelihood $p(x|C_k)$ and prior $p(C_k)$ for each class individually. Then, given an input, x, we ask what is the posterior probability that it belongs to class C_k. By adjusting the prior, we express our existing beliefs about the probabilities of each of the classes. Fig 1, we see that the innermost ring of the contour represents the greatest likelihood of a point belonging to that class. The decision function resembles a sigmoidal function whose contour is shown in Fig 2.

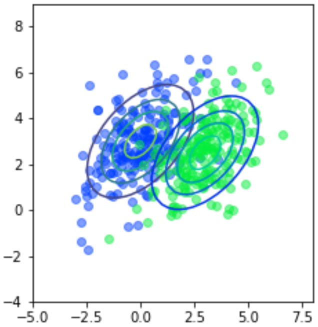

Fig. 1: Likelihood function contour overlay on top of datapoints. A datapoint is highly likely to belong to the blue class if it lies on $(0, 3)$ and very unlikely if it lies on $(-2.5, 8)$.

In this paper we consider two-class problems where two classes are denoted by C_1 and C_2. Then, given an observation x, we compute its probability of belonging to C_1 via Bayes Theorem:

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1) + p(x|C_2)p(C_2)} \tag{1}$$

A higher posterior probability for C_1 will move the boundary of the decision surface in Fig 2 further to the right. Now, let us consider the plot against the analytical expression. From (1) we can define a as:

$$a = ln\frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)} \tag{2}$$

and express the posterior for C_1 as:

$$p(C_1|x) = \frac{1}{1 + \exp(-a)} = \sigma(a) \tag{3}$$

$\sigma(a)$ is known to be a logistic sigmoid function. Then, we write the posterior probability with its corresponding components as:

$$p(C_1|x) = \sigma(\boldsymbol{W}^T\boldsymbol{X} + w_0)$$

where,

$$\begin{aligned} \boldsymbol{W} &= \Sigma^{-1}(\boldsymbol{\mu_1} - \boldsymbol{\mu_2}) \\ w_0 &= -\frac{1}{2}\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu_2} + \ln\frac{p(C_1)}{p(C_2)} \end{aligned} \tag{4}$$

since the quadratic terms in $\boldsymbol{X}$ cancel due to the fact that both covariance matrices are equal, this leads to a linear function of $\boldsymbol{X}$ in the argument of $\sigma(\cdot)$. This is consistent with the linear boundary seen in Fig 2.

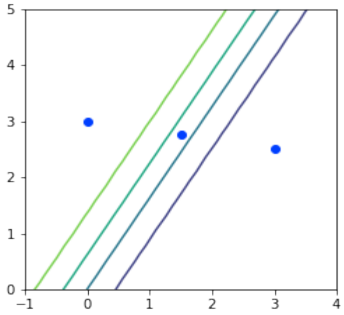

Fig. 2: Illustrative decision surface for the probability of membership to C_1 showing points x_1, x_3 and x_2 (from left to right). Contour shows x_1 has a high probability of membership to C_1 and x_3 has a low probability. As x_3 lies on the decision boundary, the probability of it belonging to C_1 is 0.5.

II. Fisher LDA and ROC Curve

Dimensionality reduction is the technique of projecting data down from a higher dimension. We can perform linear classification by using this technique; by projecting data down from hyperspace to a one-dimensional line we can separate datapoints into k classes. However, when this is done, there is a lot of overlap between points and it is difficult to find a class boundary. To solve this problem, we must find a line of projection which maximises *between-class* variance so that its tractable to find a decision boundary and minimises *within-class* variance such that most points belonging to class C_k is around the mean μ_k which gives us a greater certainty about

Published on November 28, 2020. The author can be contacted via si1u19@soton.ac.uk

class membership for a given prediction $\hat{y}$. To this end, we make use of the Fisher Criterion which states:

$$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{5}$$

where,

$$m_1 = \frac{1}{N_1} \sum_{n \in C_1}^{N_1} X_n$$

$$m_2 = \frac{1}{N_2} \sum_{n \in C_2}^{N_2} X_n$$

$$s_k^2 = \sum_{n \in C_k}^{N} (y_n - m_k)^2$$

$$y_n = \boldsymbol{W}^T \boldsymbol{X}_n$$

thus,

$$\boldsymbol{w}_F = \frac{\boldsymbol{m}_1 - \boldsymbol{m}_2}{\boldsymbol{S}_1 + \boldsymbol{S}_2} \tag{6}$$

$$\boldsymbol{w}_F = (\boldsymbol{S}_1 + \boldsymbol{S}_2)^{-1}(\boldsymbol{m}_1 - \boldsymbol{m}_2)$$

where $\boldsymbol{m_k}$ and $\boldsymbol{S_k}$ are the mean and covariance matrices respectively. In this paper we consider the 2-dimension case illustrated by Fig 3 with the objective of investigating the classification accuracy if we project data down to the Fisher discriminant direction compared to a random one.

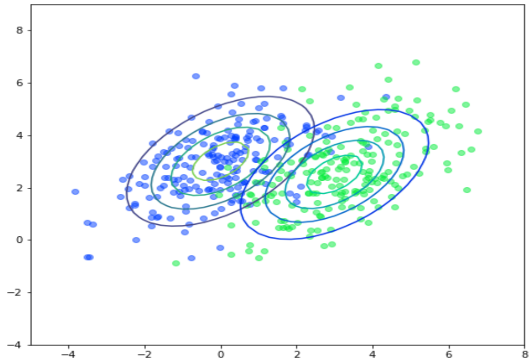

Fig. 3: 200 samples drawn from a Gaussian distribution with likelihood contours for 2 classes in blue and green with centres $\mu_1 = (0, 3)$ and $\mu_2 = (3, 2.5)$ respectively. The Fisher linear discriminant vector $\boldsymbol{w}_F = [-1.08, 0.667]^T$

Plotting the histogram distribution of $\boldsymbol{w}_F$ projection allows us to see the variance and overlap of data after applying dimensionality reduction. With a projection vector which maximises class separation, we would expect to find a minimum overlap which reduces the number of false positives made by the classifier. A *false positive* is when a classifier incorrectly identifies class membership. As opposed to this, a *true positive* is a correct identification of class membership.

In Fig 4, we find a good separation as there is little overlap between the two classes. We can now plot the *Receiving Operating Characteristics* or ROC, shown in Fig. 5. This plot provides a summary of the proportion of datapoints being misclassified and its area under the curve (AUC) gives us an accuracy value in the range $[0, 1]$ where 1 indicates 100% accuracy. Referring back to Fig 4, if imagine sweeping a threshold from -6 to 6, we can see that at -6, the threshold decision boundary will mis-classify class C_1 but correctly classify C_2. A visualisation of this sweep is given in Fig. 8. Returning to the ROC curve in Fig 5, this scenario is shown at $(1.0, 1.0)$ where both false positive and true positive rates are 100% as expected. When the threshold is moved to 6, we find in the ROC curve that both true and false positive rates fall to 0, as now, both classes are mis-classified. From this, we infer that theoretically, the best separation will result in a ROC curve that passes through the point (0.0,1.0).

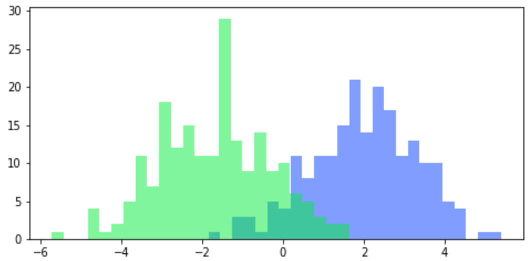

Fig. 4: Histogram of data distribution after projecting points onto $\boldsymbol{w}_F$. This projection vector has separated classes C_1 (blue) and C_2 (green) with a small overlap.

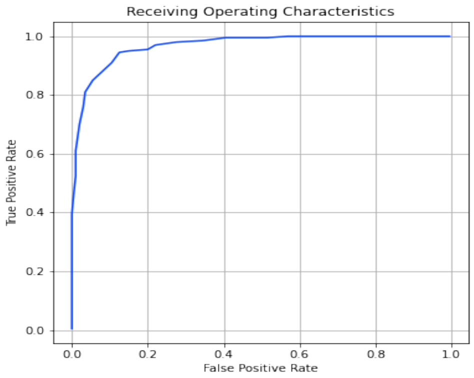

Fig. 5: ROC for a two class dataset with AUC = 0.967. The graph also shows it is best to use threshold = 0.18. The maximum accuracy we were able to achieve is 0.91.

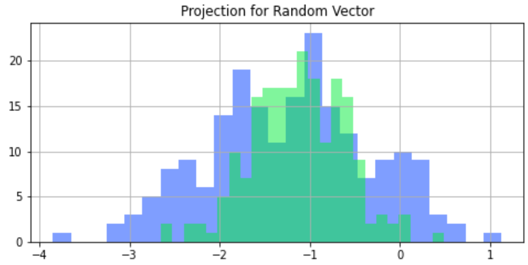

Fig. 6: Histogram of data distribution after projecting points onto a random vector. This projection vector provides poor separation of classes C_1 and C_2 as there is a large overlap.

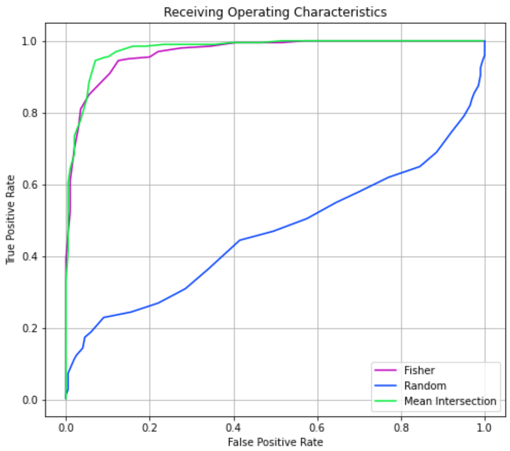

Fig. 7: ROC showing performance of mean intersection (AUC_μ), Fisher (AUC_F) and random vector (AUC_R). Comparing the AUCs, we find $AUC_\mu = 0.972$ is the highest, indicating best performance.

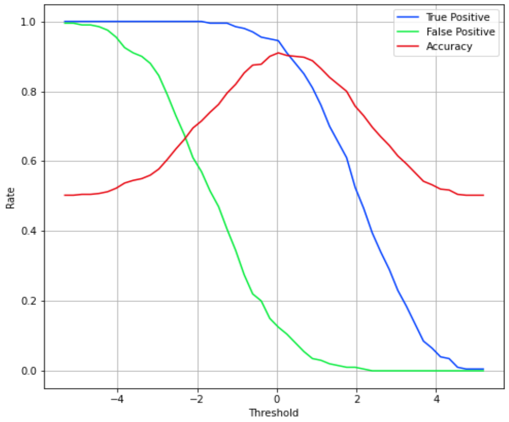

Fig. 8: As we sweep the threshold from -6 to 6, the accuracy begins to increase until we arrive at 0 where there is maximum separation of both classes.

Next, compare the performance of Fisher with two other alternatives: a vector which connects the means of two classes (mean intersection), μ_1 and μ_2 and a random vector. Our analysis shows the random vector has the poorest performance. In this case, the random vector curve passes close to (0.5,0.5) so we know it assigns class membership with almost 50:50 probability. But it should be noted that on occasions, the random vector can in fact assign opposite class membership, making it worse than random as is currently seen in Fig 7. The main basis for comparing performance in ROC plots is to compute the AUC for each curve. Thus, we find the mean intersection vector has the greatest area $AUC_\mu = 0.972$, followed by Fisher $AUC_F = 0.963$ and then random $AUC_R = 0.465$.

III. Mahalanobis Distance

Mahalanobis distance is a measure of the distance between a point and the mean of a multivariate distribution. It is a generalisation which uses the number of standard deviations away from the mean a given point is, irrespective of the scale, making it independent of the magnitude of the scale. It is computed by:

$$D_M(\boldsymbol{x}) = \sqrt{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \tag{7}$$

where $\boldsymbol{x}$ is the datapoint, $\boldsymbol{\mu}$ is the mean and Σ is the covariance. We now consider using a distance to mean classifier which assigns class membership based on the distance a given point is from the mean. Since we know euclidean distance does not take into account the kurtosis and variance, we expect D_M to perform better as it is aware of both the aforementioned factors. Table I shows D_M attains an accuracy of 99%.

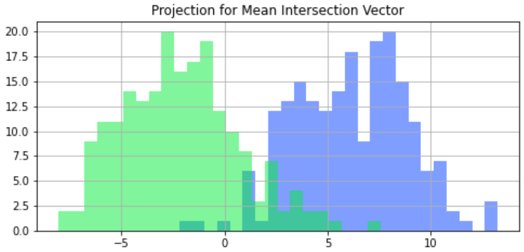

Fig. 9: Histogram of data distribution after projecting points onto the mean intersection vector. This projection vector provides good separation of classes C_1 and C_2 as there is little overlap.

TABLE I: Distance to mean classifier performance based on Euclidean and Mahalanobis distance metrics.

Method	Set	Accuracy
Euclidean	Train	94%
Euclidean	Test	96%
Mahalanobis	Test	93%
Mahalanobis	Train	99%

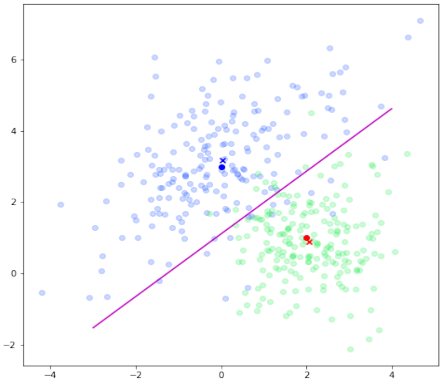

Fig. 10: μ_1 and μ_2 in blue and red showing the mean of C_1 and C_2 in bold. The magenta line shows the decision boundary.