CS109A/STAT121A/APCOMP209a: Introduction to Data Science

Advanced Section 6: Topics in Supervised Classification

Instructors: Pavlos Protopapas, Kevin Rader

Nick Hoernle
nhoernle@g.harvard.edu

Section Times
Wed 3-4pm & Wed 5:30-6:30 & Thurs 2:30-3:30

1 Classification Recap

We have already seen a popular way of making a classification decision between two classes by evaluating the *log-odds* that a datapoint belongs to one or the other class. Under the assumption that the log-odds ratio is linear in the predictors, we arrived at Logistic Regression. Linear Discriminant Analysis presents another technique for finding a linear separating hyperplane between two classes of data. Consider a problem where we have data drawn from two multivariate Gaussian distributions: $X_1 \sim N(\mu_1, \Sigma_1)$ and $X_2 \sim N(\mu_2, \Sigma_2)$. If we wish to make a classification decision for a new datapoint, we can evaluate the probability that the datapoint belongs to a class and can again study the ratio of these probabilities to make that decision.

Since, we are interested in evaluating the probability that a datapoint belongs to a certain class, we wish to evaluate: $p(Y = k|X = x)$ (i.e. given datapoint x, what is the the probability that it belongs to class k). Using the axioms of probability (and specifically those of conditional probability), we can derive Bayes rule:

$$p(Y = k|X = x) = \frac{p(X = x, Y = k)}{p(X = x)} = \frac{p(X = x|Y = k)p(Y = k)}{p(X = x)}$$

Bayes' rule allows us to express $p(Y = k|X = x)$ in terms of the class-conditional densities $p(X = x|Y = k)$ and the probability that a datapoint belongs to a class $p(Y = k)$. For simplification of notation, if we let $f_k(x)$ denote the class-conditional density of x in the class $Y = k$, and we let π_k be the *prior probability* that a datapoint chosen at random will be observed as class k (note that $\sum_{k=1}^{K} \pi_k = 1$), we obtain:

$$p(Y = k|x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

In logistic regression, you actually already reasoned about the best classification decision using the *posterior* probability that a classification should be a 0 or a 1. In that case, we evaluated the *log-odds* for the classification decision of one class over another. From what was discussed in class, minimising the misclassification rate corresponds to maximising the posterior probability that a datapoint should be classified to class k. Making a classification decision by maximising the posterior probability results in a *Bayes' classifier* (note that choosing a decision boundary is a field of study in it own rights and is known as Decision Theory).

2 Linear Discriminant Analysis (LDA)

LDA makes the explicit assumption that for each of the K classes, the class conditional densities $p(x|Y = k) = f_k(x)$ are multivariate Gaussian distributions. We therefore have $P(x|Y = k) \sim N(\mu_k, \Sigma_k)$ which means that the densities follow:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\{\frac{-1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\}$$

If this is the case, we wish to analyse the posterior probability that an unlabeled datapoint belongs to one of the classes and select the class that maximises this posterior probability (i.e. classify the datapoint to the class k that maximises $p(Y=k|x)$). Thinking about the two class classification problem is illuminating. If we have $f_1(x)$ and $f_2(x)$ as the multivariate Gaussian distributions for classes 1 and 2 respectively with prior probabilities π_1 and π_2, we would classify the datapoint x to $\arg\max\limits_k(p(Y=k|x)) = \arg\max\limits_k(\log p(Y=k|x))$. We can now apply Bayes' rule and drop the denominator that is independent of k to rather find the class k which $\arg\max\limits_k(\log(\pi_k)+\log(f_k(x)))$. We therefore obtain the following discriminant function ($\delta_k(x)$) for each class k remembering that we select k with the highest $\delta_k(x)$, (noting again that the common 2π constant can also be dropped from the maximisation):

$$\delta_k(x) = log(\pi_k) - \frac{1}{2}log(|\Sigma_k|) - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$$

It is worth noting that $(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$ is known as the (squared) Mahalanobis distance metric and it is a measure of the distance between a point and a distribution. LDA further makes the assumption that the covariance matrices of the different classes are equal: $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = \Sigma$. We can analyse the *decision boundary* between two classes where the discriminant functions are exactly equal:

$$\begin{aligned}
\delta_1(x) &= \delta_2(x)\\
log(\pi_1) - \frac{1}{2}log(|\Sigma|) - \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) &= log(\pi_2) - \frac{1}{2}log(|\Sigma|) - \frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2)\\
0 &= log(\frac{\pi_1}{\pi_2}) - \frac{1}{2}(\mu_1+\mu_2)^T\Sigma^{-1}(\mu_1-\mu_2) + x^T\Sigma^{-1}(\mu_1-\mu_2)
\end{aligned}$$

This result is linear in x and it discriminates between the data from class 1 and the data from class 2 (hence the 'linear discriminant' analysis). Notice that the $x^T\Sigma x$ term is canceled out (as again this term is independent of k and therefore does not play a role in the maximisation).

3 Quadratic Discriminant Analysis

The setup for this problem is exactly the same, but here we relax the pooled variance assumption and rather allow the individual classes to have their own class specific covariances. We therefore still arrive at the same discriminant function as before:

$$\delta_k(x) = log(\pi_k) - \frac{1}{2}log(|\Sigma_k|) - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$$

But when evaluating the decision boundary between the classes, the algebra becomes a little messy in deriving the result [1]:

[1] You should validate this result yourselves for practice

$$0 = log(\frac{\pi_1}{\pi_2}) + \frac{1}{2}log(\frac{|\Sigma_2|}{|\Sigma_1|}) - \frac{1}{2}\left[\mu_1^T\Sigma_1^{-1}\mu_1 - \mu_2^T\Sigma_2^{-1}\mu_2 + 2x^T(\Sigma_2^{-1}\mu_2 - \Sigma_1^{-1}\mu_1) + x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x\right]$$

The $x^T(\Sigma_1^{-1} - \Sigma_2^{-1})x$ term now shows that this decision boundary is quadratic in x.

4 Fisher's Linear Discriminant

In Section 4 we discussed the dimensionality reduction method of Principal Component Analysis (PCA). Similarly, LDA can be thought of as a dimensionality reduction technique where a linear discriminant is found that attempts to maximally separate two different classes. PCA, under its assumptions, attempts to find the Principal Components that account for most of the variance in the dataset. On the other hand, LDA attempts to model the difference between the classes of data.[2]

Let's imagine an example in two dimensions with data belonging to one of two classes, where the two classes have (Gaussian) marginal distributions that are highly elongated but aligned (see figure 1 for an example). As you learned, the first principal component will extract the dimension that captures the highest variance in the data (in this case it will be exactly x_1). For the purposes of dimensionality reduction, projecting the data onto this component will result in a one dimensional representation where the data is entirely inseparable (see figure 2).[3]

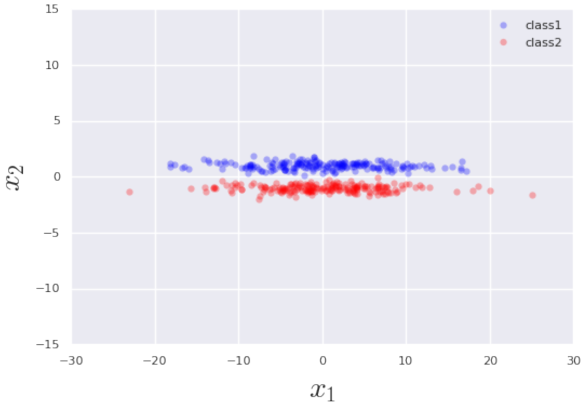


Figure 1: Example dataset where LDA will present a more useful dimensionality reduction than PCA

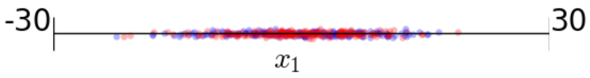


Figure 2: Example of a projection that does not discriminate between the data classes

[2] It is worth noting LDA is a supervised technique whereas PCA is unsupervised even though, in this case, we are comparing them for the same function of dimensionality reduction.

[3] For the purposes of visualising the data, I have added a small amount of vertical jitter for plotting the points.

We can clearly see that for the task of classification, a more useful projection would be one onto the x_2 component. Fisher Linear Discriminant Analysis (LDA) considers maximising an objective with the goal of finding a *discriminating* hyperplane between these classes [1] (and not the hyperplane that describes the variance of the dataset in its entirety). LDA uses the additional information of known class labels to find a more useful discriminator between the data classes.

Classification in a high dimensional space is hard as the relative distances between data becomes extremely sparse making distance metrics less useful. This is known as the curse of dimensionality [2]. Rather, imagine we were able to perform a projection of the data onto a lower dimensional subspace, such that the means of the data clusters are maximally separated, yet the variances of the within cluster data are minimised. LDA actually performs this projection (under the assumption of Gaussian data).

Let $\mathbf{W}$ be a matrix that consists of column vectors $\mathbf{w}$ that each represent a hyperplane that discriminates between two classes. Our aim is to find $\mathbf{W}$ such that the objective of maximising the distances between the means of the classes and minimising the within class variance is achieved. To do this, we need the *between-class scatter* matrix (measuring the spread of the class means)

$$S_B = \sum_k (\mu_\mathbf{k} - \bar{\mathbf{x}})(\mu_\mathbf{k} - \bar{\mathbf{x}})^T$$

where μ_k is the mean of class k and $\bar{X}$ is the mean of the entire dataset. And we need the *total within-class scatter* matrix

$$S_W = \sum_k \sum_{i \in k} (\mathbf{x_i} - \mu_\mathbf{k})(\mathbf{x_i} - \mu_\mathbf{k})^T$$

there are K classes, with $k = \{1, \ldots, K\}$ indexing the k^{th} class and $i \in k$ means that observation i is in class k; thus x_i is a datapoint in class k. The optimisation task is then to find the vector $\mathbf{w}$ that maximises the following objective (also called the Rayleigh quotient):

$$J(\mathbf{w}) = \frac{\text{between class scatter}}{\text{within class scatter}} = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

Again, it is helpful to study the two class case: let's assume we have data from two Gaussian distributions $(X|Y = 1) \sim N(\mu_1, \Sigma_1)$ and $(X|Y = 2) \sim N(\mu_2, \Sigma_2)$ for classes 1 and 2 respectively. We have $S_B = \sum_{k=1}^{2} (\mu_\mathbf{k} - \bar{\mathbf{x}})(\mu_\mathbf{k} - \bar{\mathbf{x}})^T = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ and $S_W = (\Sigma_1 + \Sigma_2)$.

Maximising the objective J with respect to $\mathbf{w}$ is equivalent to maximising the numerator while holding the denominator constant as both the numerator and denominator have the same order of $\|\mathbf{w}\|$ and thus J is invariant to scale changes in $\|\mathbf{w}\|$. Further, as we are only interested in the direction of $\mathbf{w}$, we can hold the denominator constant to 1. So we find $\mathbf{w}$ to $max(\mathbf{w}^T S_B \mathbf{w})$ such that $\mathbf{w}^T S_W \mathbf{w} = 1$ which results in the following Legrangian:

$$L = \mathbf{w}^T S_B \mathbf{w} + \lambda(\mathbf{w}^T S_W \mathbf{w} - 1)$$

Setting $\frac{\partial L}{\partial \mathbf{w}}$ to 0 yields $2(S_B - \lambda S_W)\mathbf{w} = 0$ and so:

$$S_W^{-1}S_B\mathbf{w} = \lambda\mathbf{w}$$

In general the solution exists if S_W^{-1} exists. Moreover, the solution is not unique and corresponds to the eigenvalue problem for the $S_W^{-1}S_B$ matrix.

Referring back to the two class case, and using the definition of S_B in the two class case, we see that

$$S_W^{-1}S_B\mathbf{w} = S_W^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\mathbf{w} = \lambda\mathbf{w}$$

and finally noting that $(\mu_1 - \mu_2)^T\mathbf{w} = \alpha$ is a scalar we get:

$$S_W^{-1}(\mu_1 - \mu_2) = \frac{\lambda}{\alpha}\mathbf{w}$$

Again, we are only interested in the direction of $\mathbf{w}$ (and can discard the scalar multiplier), so

$$\mathbf{w}^* = S_W^{-1}(\mu_1 - \mu_2) \tag{1}$$

If you refer back to Section 2 you should see that $S_W^{-1}(\mu_1 - \mu_2) = 2\Sigma^{-1}(\mu_1 - \mu_2)$ gives the same direction vector for the decision boundary that was found in that Section.

In the more general case where we have K classes (and K associated means μ_k), with $K \geq 2$, LDA constructs a $\mathbf{W}$ matrix that spans a maximum of $K-1$ dimensions. Moreover, when the data are in $\mathcal{R}^P$, with $P > K$, LDA can be seen as a method that projects the P dimensional data onto a $K-1$ dimensional subspace for which the data is maximally (linearly) separated into its clusters.

You can reach this result by studying the rank of the $S_W^{-1}S_B$ matrix. We firstly need the result[3] that:

$$rank(S_W^{-1}S_B) \leq \min\{rank(S_W^{-1}), rank(S_B)\} \tag{2}$$

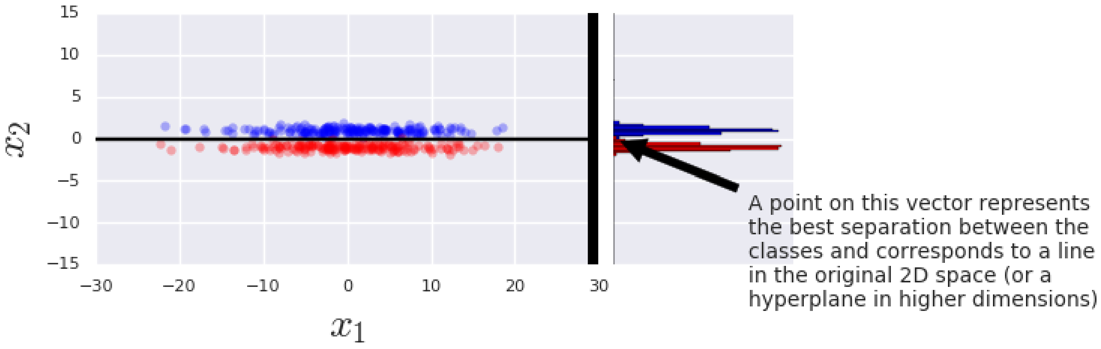
Secondly, **Lemma 1:** $rank(S_B) \leq K-1$.

Proof: Consider P dimensional multivariate vectors μ_k with $k = 1\ldots K$ drawn i.i.d from a distribution with mean $\bar{\mathbf{x}}$. The covariance matrix can be estimated by the drawn samples:

$$V\hat{ar}(\mu) = \frac{1}{K}\sum_{k=1}^{K}(\mu_k - \bar{\mathbf{x}})(\mu_k - \bar{\mathbf{x}})^T$$

Interpreting this result as a covariance matrix is noting that the estimate must have K-1 degrees of freedom. Now, note that $S_B = KV\hat{ar}(\mu)$. Recall again that an assumption is that K, the number of classes, is less than P, the number of dimensions, and therefore S_B is not a full rank matrix. It follows that S_B also has $K-1$ degrees of freedom corresponding to $rank(S_B) \leq K-1$.

Finally, since $rank(S_B) \leq K-1$ and from eq 2, we see that the rank of $S_W^{-1}S_B$ is less than or equal to $K-1$. A last note on the dimensionality reduction is that if P, the number of dimensions, is greater than K, the number of classes, then S_B is full rank with $rank(S_B) = P$ and thus LDA results in no dimensionality reduction.



A point on this vector represents the best separation between the classes and corresponds to a line in the original 2D space (or a hyperplane in higher dimensions)

Figure 3: Example of a projection that does discriminate between the data classes

5 Notes

1. LDA/QDA classifies a new datapoint to the class with the closest centroid. Here 'closest' is measured in the Mahalanobis metric, using a pooled covariance estimate.

2. Under the assumption that the data are multi-variate Gaussian within each class, LDA is a Bayes' classifier (i.e. it minimises the probability of making a mis-classification).

3. LDA results in linear decision boundaries.

4. In many cases, when linear decision boundaries are inadequate for separating the classes, QDA can be used, with the cost of needing to estimating more parameters.

5. LDA needs to approximate the following $(K + K + P^2)$ statistics from the data:

   (a) K priors: $\hat{\pi}_k = \frac{N_k}{N}$

   (b) K means: $\hat{\mu}_k = \sum_{i \in k} \frac{x_i}{N_k}$

   (c) 1 covariance matrix: $\hat{\Sigma} = \sum_{k=1}^{K} \sum_{i \in k} \frac{(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{N-K}$

6. QDA needs to approximate the following $(K+K+K \times P^2)$ statistics from the data (notice the possibility for over-fitting your training data here):

   (a) K priors: $\hat{\pi}_k = \frac{N_k}{N}$

   (b) K means: $\hat{\mu}_k = \sum_{i \in k} \frac{x_i}{N_k}$

   (c) K covariance matrices: $\hat{\Sigma}_k = \sum_{i \in k} \frac{(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T}{N_k}$

Other interesting and worthwhile references are: [4], [5], [6].

References

[1] M. Welling, "Fisher linear discriminant analysis," *Department of Computer Science, University of Toronto*, vol. 3, no. 1, 2005.

[2] L. Parsons, E. Haque, and H. Liu, “Subspace clustering for high dimensional data: a review,” *Acm Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 90–105, 2004.

[3] K. B. Petersen, M. S. Pedersen, *et al.*, “The matrix cookbook,” *Technical University of Denmark*, vol. 7, p. 15, 2008.

[4] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[6] A. M. Martínez and A. C. Kak, “Pca versus lda,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.