Discriminant Analysis

Jo Dang

Introduction

Logistic Regression:

$$Pr(Y = k|X = x)$$

models the conditional distribution of the response Y, given the predictor(s) X's.

In this approach, however, we model the **distribution of the X's separately for each value of** Y. Then, we use Bayes' theorem to flip these into estimates for $Pr(Y = k|X = x)$.

Motivations:

- When there is substantial separation between 2 classes, the parameter estimates are unstable. The discriminant method accounts for this
- If the distribution of X's is approximately normal and the sample size is small, this might even be more accurate than logistic regression
- This can extend to 2 or more response classes, although multinomial logistic regression can also work

Methodology

Classifying Observations

Suppose there are $K \geq 2$ classes. Let:

- π_k be the prior probability that an observation is in the k^{th} class.
- $f_k(x) \equiv Pr(X|Y = k)$ is the density of X for an observation that comes from the k^{th} class. This is large if there is a high chance that an observation in class k^{th} has $X = x$.

Then, Bayes' Theorem says: which we will refer to as $p_k(x)$. This is the posterior probability that an observation $X = x$ belongs to the k^{th} class, given the predictor value.

P = 1: Only 1 Predictor

Assume that $f_k(x)$ is normal or Gaussian. Then, plug the normal density function in and:

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu_k)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu_l)^2}}$$

π_k denotes the prior probability that an observation belongs to the k^{th} class, not to be confused with the mathematical constant. The observation is assigned to a class where this value is the largest. A simpler expression, however, can be achieved if we take the *log* of this and rearrange the terms. Then, we get

$$\delta_k(x) = x.\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$$

The Linear Discriminant Analysis (LDA) method approximates the Bayes classifier by using estimates of π_k, μ_k, and σ^2 in the expression above.

The following estimates are used:

$$\hat{\mu} = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

Where n is the total number of training observations, n_k is the number of observations in the k^{th} class. μ_k is the average of all training observations in the k^{th} class, and $\hat{\sigma}^2$ is the weighted average of the sample variances for each of the K classes. The πs here are our prior information. When we have weakly informative priors, the LDA estimates π_k using the proportion of the training observations that belong to the k^{th} class. In other words, $\hat{\pi}_k = \frac{n_k}{n}$.

To find an appropriate classification for observation x, we choose the class with which we get the largest value of $\hat{\delta}_k(x)$, calculated by plugging in $\hat{\mu}_k$, $\hat{\sigma}$, $\hat{\pi}_k$.

The discriminant functions are linear functions of x. Since the covariance matrix determines the shape of the Gaussian density, the Gaussian densities for different classes have the same shape but are shifted versions of each other.

LDA for P > 1

We assume that $X = (X_1, X_2, ..., X_p)$ is drawn from a **multivariate Gaussian distribution**. This assumes that *each individual predictor follows a 1D normal distribution, with some correlation between each pair of predictors.* Then, X $N(\mu, \sum)$ where $\mu = E(X)$ is the mean of X, a class-specific mean vector with p components or a p-dimensional variable, and $\sum = Cov(X)$ is the *pxp* covariance matrix of X. The multivariate Gaussian density is

$$f(x) = \frac{1}{(2\pi)^{\frac{p}{2}} |\sum|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \sum^{-1} (x-\mu)}$$

In our context where $p > 1$, the LDA classifier assumes that the observations in the k^{th} class are drawn from a multivariate Gaussian distribution $N(\mu_k, \sum)$, where μ_k is the mean vector for class k and $\sum$ is a covariance matrix that is common to all K classes. A lot of math is involved here, but ultimately, we find out that the **Bayes classifier assigns an observation $X = x$ to the class for which**

$$\delta_k(x) = \frac{x^T}{\sum \mu_k} - \frac{\mu_k^T}{2 \sum \mu_k} + log\pi_k$$

is largest. This is the vector/matrix version of the other δ value we explored for $p = 1$.

Quadratic Discriminant Analysis

Like LDA, QDA classifier assumes that the observations from each class are drawn from a Gaussian distribution and plugging estimates for the parameters into Bayes'theorem in order to perform prediction. Unlike LDA, however, **QDA assumes that each class has its own covariance matrix**. Then, X $N(\mu, \sum_k)$ where $\mu = E(X)$ is the mean of X, a class-specific mean vector, and $\sum_k$ is a covariance matrix for the k^{th} class. Under this assumption, **we have a new δ value which we will be using to classify an observation X=x**.

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

The 1st and 4th terms are new. Another thing that is new is **the quantity x appears as a quadratic function**.

Data and Results

For the purposes and scope of this project, we will not be splitting the data set into training and test as we would normally do when we try to train a model for making predictions. Instead, using the data set as whole for both methods will be beneficial in helping us understand the shared as well as different traits and functionality between LDA and Logistic Regression. Thus, prediction power will not be a contributing factor in examining the two methods.

Logistic Regression Model

The task at hand is to predict whether or not an individual will default on the basis of credit card balance and student status, so $p > 1$. We will start with a Logistic Regression Model.

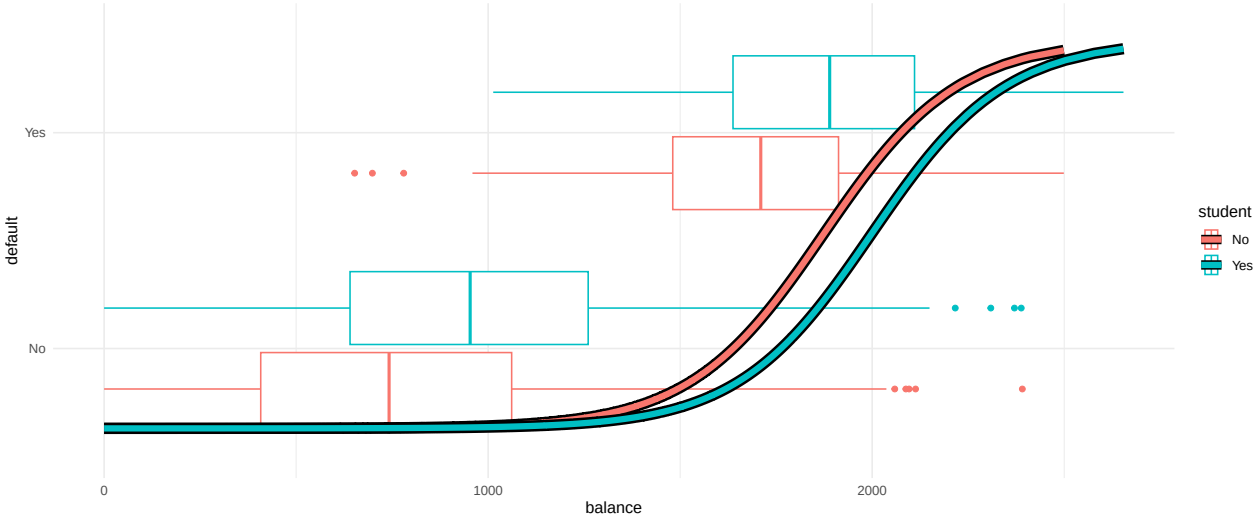

Figure 1: Logistic Regression Model Predictions Compared to Original Information From Data

This graph essentially shows us visually that the logistic regression model is successful in detecting the trends of defaulting give the 2 explanatory variables in question as well as the differences between them.

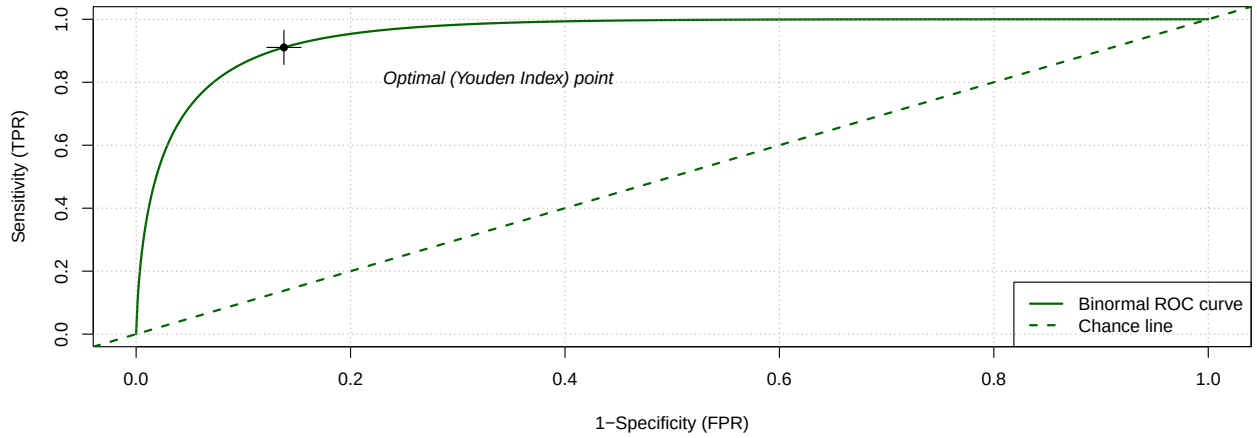

Figure 2: ROC Curve for Logistic Regression Model

The ROC curve traces out 2 types of error:

- The True Positive Rate (TPR) is the Sensitivity: the fraction of defaulters that are correctly identified

- The False Positive Rate is 1-Specificity: the fraction of non-defaulters that we classify incorrectly as defaulters.

The ideal ROC curve hugs the top left corner, indicating a high true positive rate and a low false positive rate. The dotted line represents the "no information" classifier; this is what we would expect if student status and credit card balance are not associated with probability of default.

LDA Model

Table 1: Confusion Matrix for LDA Model With Default Threshold

	No	Yes
No	9644	252
Yes	23	81

LDA predicted that a total of 104 people would default. 81 of those actually defaulted and 23 did not. Hence *only 23/9667 of non-defaults were incorrectly labeled.* However, of the 333 defaults, 252 (or 75.7 %) were missed by LDA. So while the overall error rate is low, the error rate among defaults is very high. In this case, *the Sensitivity (percentage of true defaults identified) is 24.3%. The Specificity (percentage of non-defaults correctly identified) is (1 − 23/9667) = 99.8 %.*

The overall error rate is 2.75%: LDA made incorrect predictions for 23 individuals who did not default and for 252 individuals who did default.

LDA is trying to approximate the Bayes classifier, which will yield the smallest possible **total** number of misclassifications, regardless of the class. In our context, a credit card company might particularly want to avoid incorrectly classifying an individual who will default, whereas incorrectly classifying an individual who will not default, though still to be avoided, is less problematic. It is possible to modify LDA in order to develop a classifier that better meets the credit card company's needs.

The Bayes classifier works by assigning an observation to the class for which the posterior probability $p_k(X)$ is greatest. In the two-class case, this amounts to assigning an observation to the default class if $Pr(default = Yes|X = x) > 0.5$. Thus, the Bayes classifier, and by extension LDA, uses a threshold of 50 % for the posterior probability of default. However, if we are concerned about missing actual positives, then we can consider lowering this threshold. For instance, $Pr(default = Yes|X = x) > 0.2$.

Table 2: Confusion Matrix for LDA Model With Threshold 0.2

	No	Yes
No	9432	138
Yes	235	195

Now LDA predicts that 430 individuals will default. Of the 333 individuals who default, LDA correctly predicts all but 138, or 41.4 %. This is a vast improvement over the error rate of 75.7% that resulted from using the threshold of 50%. However, this improvement comes at a cost: now 235 non-defaults are incorrectly classified. The overall error rate has increased slightly to 3.73 %. But a credit card company may consider this slight increase in the total error rate to be a small price to pay for more accurate identification of individuals who do indeed default.

There is a trade-off between error rate and the threshold value. To decide which threshold value is best, we need domain knowledge, such as detailed information about the costs associated with default.

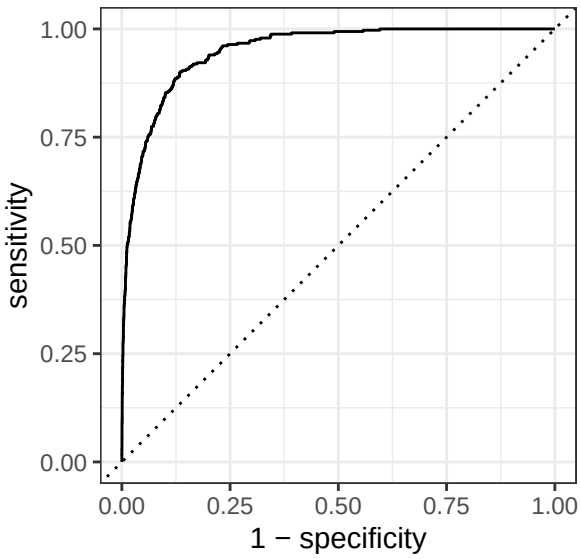

Figure 3: ROC Curve for LDA Model With Threshold 0.2

You might notice that this ROC Curve is exactly the same as the one we found for Logistic Regression!

Discussion/Conclusion

LDA vs Logistic Regression

As our example shows, the use of these 2 methods in our context gave similarly successful prediction models. Instead of thinking about them one over the other, we can think of them as 2 methods that are both at our disposal, and choosing to use one over another is up to us based on the context we might be in or what we are trying to investigate.

LDA, in this context, introduced the flexibility and concept of setting a threshold, which could be useful in certain situations. However, as we also discovered, the default threshold of 0.5 actually posed a problem for us which required a closer look and some modifications to the model. Both LDA and Logistic Regression, as well as any other method, have their own positives and negatives. It is our job to understand what each of them are doing and how they work so that we can make the least wrong model to our knowledge.

QDA vs LDA

Why would we pick one over another? In other words, why does it matter that each class has its own common covariance matrix versus every class sharing the same one? The answer lies in **the bias-variance trade-off**.

When there are p predictors, then estimating a covariance matrix requires estimating $\frac{p(p+1)}{2}$ parameters. QDA estimates a separate covariance matrix for each class, so we get $K\frac{p(p+1)}{2}$ parameters. *Remember that these are the number of parameters and not the value we are trying to estimate.* With $p = 50$, then, we have

some multiple of 1,275 parameters, which is a lot. By instead assuming that the K classes share a common covariance matrix, the LDA model becomes linear in x, which means there are $K * p$ linear coefficients to estimate.

LDA is a much less flexible classifier than QDA, and so has substantially lower variance. This can potentially lead to improved prediction performance. But there is a trade-off: if LDA's assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.

Roughly speaking, LDA tends to be a better bet than QDA if there are relatively few training observations and so reducing variance is crucial. In contrast, QDA is recommended if the training set is very large, so that the variance of the classifier is not a major concern, or if the assumption of a common covariance matrix for the K classes is clearly untenable.

References

- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York:Springer, 2013.
- (No date) Discriminant Analysis. Available at: https://online.stat.psu.edu/stat508/book/export/html/645 (Accessed: 27 May 2024)
- Khan, M.R.A. (2024) ROCit: An R package for performance assessment of binary classifier with visualization. Available at: https://cran.r-project.org/web/packages/ROCit/vignettes/my-vignette.html (Accessed: 27 May 2024).
- Create a border around `geom_line` in GGPLOT2 (1968) Stack Overflow. Available at: https://stackoverflow.com/questions/73528730/create-a-border-around-geom-line-in-ggplot2 (Accessed: 27 May 2024).
- Humboldt-Universität zu Berlin | Geography Department (no date) LDA and model assessment in R. Available at: https://pages.cms.hu-berlin.de/EOL/gcg_quantitative-methods/Lab11_LDA_Model-assessment.html (Accessed: 27 May 2024).
- Ranvir (2018) Linear Discriminant Analysis, RPubs. Available at: https://rpubs.com/ranvirkumarsah/LDA (Accessed: 27 May 2024).
- Dunn, T. (2022) An introduction to statistical learning with the Tidyverse and Tidymodels, 4 Classification. Available at: https://bookdown.org/taylordunn/islr-tidy-1655226885741/classification.html#linear-discriminant-analysis-for-p-1-1 (Accessed: 27 May 2024).

Appendix

I genuinely wish that I was better at story-telling

```
library(ISLR2)
data("Default")

# Fit logistic regression model
glmmodel <- glm(default ~ balance + student, data = Default, family = binomial)
grid <- expand.grid(
  balance = seq(min(Default$balance), max(Default$balance), length.out = 100),
  student = levels(Default$student)
)
grid$pred <- predict(glmmodel, newdata = grid, type = "response")
```

```
#Logistic Regression Model Predictions Compared to Original Information From Data
ggplot() +
  geom_boxplot(data = Default, aes(x = balance, y = default , color = student), height = 0.4) +
  geom_borderline(data = grid, aes(x = balance,
                                   y = (pred+0.3)*1.9, #scaling the predictions to fit the boxplot
```

```
                                    color = student), linewidth=2, bordercolour = "black")+
  theme_minimal()
```

```
#ROC Curve for Logistic Regression Model
class <- glmmodel$y
score <- qlogis(glmmodel$fitted.values)

myroc <- rocit(score = score,
                   class = class,
                   method = "bin")

plot(myroc, col = "darkgreen")
```

```
#confusion matrix for default 0.5 threshold

lda_model <- lda(default ~ balance + student, data=Default)
lda_prediction <- predict(lda_model, newdata = Default)
conf <- table(list(predicted=lda_prediction$class, observed=Default$default))
kable(conf)
```

```
#confusion matrix for 0.2 threshold
lda_prediction <- ifelse(lda_prediction$posterior[, 2] > .2, "Yes", "No")
conf <- table(list(predicted=lda_prediction, observed=Default$default))
kable(conf)
```

```
#ROC Curve for LDA Model With Threshold 0.2
lda_prediction <- bind_cols(
  default = Default$default,
  posterior_prob_default = lda_prediction$posterior[,2]
) %>%
  mutate(
    pred_default = ifelse(posterior_prob_default > 0.2, "Yes", "No")
  )

lda_roc <-
  yardstick::roc_curve(
    lda_prediction,
    posterior_prob_default, truth = default,
    event_level = "second"
  )
autoplot(lda_roc)
```