Assignment #3

Lindsay Brock, Gustavo Esparza, Brian Schetzsle

September 19, 2019

Logistic Regression

Linear Regression is the prior method in our text for constructing a model for inference and predictions. However, problems arise when data of interest pertains to a qualitative response. Responses of this nature, such as recording whether a patient in a hospital has survived or died in a surgery operation, call for different modeling strategies. Before exploring the particular strategy of *Logistic Regression*, we should briefly expand on why a linear model would not be suitable for this scenario. If the data was applied to a simple linear regression, we have declared our response categories to have a natural ordering (IE surviving a surgery is a greater "quantity" than dying in surgery). This is clearly not the goal set in mind, thus we must modify our desired response to consist of a probability rather than a numerical value.

Now, we will continue our exploration of Logistic Regression by first considering a basic binary response (two categories: yes and no) with a single predictor. When determining a response Y, logistic regression will model the probability that Y belongs to a particular category. As expected, this probability will always be between 0 and 1. Having moved away from linear regression, we can no longer use the classic model of $\hat{Y} = \beta_0 + \beta_1 X$. In order to keep our responses between the range of $[0, 1]$, logistic regression uses the *logistic function* defined as follows:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

We can observe that the inclusion of e ensures that probability values can approach our range of $[0, 1]$, but will never exceed them. It is also of importance to recognize that the probabilities will produce an S-shaped plot that better explains the extreme probability/response values than an ordinary linear regression. Taking our previously stated logistic function and rearranging it to solve for our exponential expression, we obtain the following result:

$$\frac{P(X)}{1 - P(X)} = e^{\beta_0+\beta_1 X}$$

The ratio seen above is referred to as the *odds*, and can now take any value between 0 and ∞. This new odds expression can be more suitable for expressing how much more likely a "Yes" outcome is compared to its counterpart "No". Once again, we can manipulate the above equality via logarithm to obtain the following result:

$$log\left(\frac{P(X)}{1 - P(X)}\right) = \beta_0 + \beta_1 X$$

The left-hand side is referred to as the log-odds, and is a very useful value for explaining the impact of our regression coefficients. In particular, increasing X by one unit changes the log odds by β_1, or multiplies the standard odds by e^{β_1}. It is critical to understand that the one unit increase in our regression coefficient does not directly correspond to the change in our probability $P(X)$, only the odds in the context of our responses. However, we can still observe that a positive β_1 will produce an increasing $P(X)$ as X increases and a negative β_1 will produce a decreasing $P(X)$ as X increases. In other words, our regression coefficients still provide context to how our X values determine our response.

Logistic Regression with Multiple Predictors/K Responses

Now that we have provided a basic coverage of Logistic Regression, we will expand our definitions to more complex instances where there are multiple predictor variables and/or responses with more than two categories.

For the multiple predictor case of size $p \geq 2$, we can redefine the log-odds equation as follows:

$$log\left(\frac{P(X)}{1-P(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

When considering more than one predictor, we may (and likely will) see different responses when compared to the single β_1 responses. This difference is critical in understanding that many logistic regression models can not be fully explained by a single predictor, but rather depend on multiple predictors to aptly provide accurate response categories. In addition, we must also analyze the correlation among the predictors that may lead to confounding results.

Multiple predictors are more likely to be included in a logistic regression model as opposed to more than two response categories, since *discriminate analysis* is a much more popular option for modeling these sets of data.

Estimating Coefficients

As stated before, the logistic regression model attempts to model the posterior probabilities of the K classes of responses via linear functions that ensure they sum to one and remain in $[0, 1]$. Now considering the K-response model, we have the following expanded form:

$$\begin{aligned}
log\frac{Pr(G=1|X=x)}{Pr(G=K|X=x)} &= \beta_{10} + \beta_1^T x \\
log\frac{Pr(G=2|X=x)}{Pr(G=K|X=x)} &= \beta_{20} + \beta_2^T x \\
&\vdots \\
log\frac{Pr(G=K-1|X=x)}{Pr(G=K|X=x)} &= \beta_{(K-1)0} + \beta_{K-1}^T x
\end{aligned}$$

Where the Kth class is defined by the denominator of each equation and the parameter set can be defined by $\theta = \beta_{10}, \beta_1^T, \ldots, \beta_{(K-1)0}, \beta_{K-1}^T$

Now, when fitting logistic regression models, we utilize *maximumlikelihood* with the condition likelihood of G given X. Thus, for N observations, we have the following log-likelihood:

$$l(\theta) = \sum_{i=1}^{N} \log\, p_{gi}(x_i; \theta)$$

where $p_k(x_i; \theta) = Pr(G = k|X = x_i; \theta)$

Note: For this, we will consider the logistic model consisting of two classes for our response. Now, for a binary response, we can define our log-likelihood functions as

$$l(\beta) = \sum_{i=1}^{N} (y_i log\ p(x_i; \beta) + (1 - y_i) log(1 - p(x_i; \beta))\ ) = \sum_{i=1}^{N} y_i \beta^T x_i - \log(1 + e^{\beta^T x_i})$$

In this instance, $\beta = (\beta_0, \beta_1)$. Now, taking our respective derivatives and setting them to zero, we obtain the following:

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{N} x_i(y_i - p(x_i; \beta)) = 0$$

In order to solve for β, we utilize an iterative algorithm such as the $newton - raphson$ method that required the additional second-derivative Hessian matrix:

$$\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} = -\sum_{i=1}^{N} x_i x_i^T p(x_i; \beta)(1 - p(x_i; \beta))$$

Thus, the iterative process is determined by the following β updates:

$$\beta^{new} = \beta^{old} - \left(\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} \right)^{-1} \frac{\partial l(\beta)}{\partial \beta}$$

This process involves taking the derivative of each β^{new} in order to update our process.

When solving for Logistic Regression coefficients, it is convenient to redefine the previous equations in matrix notation. Thus, we will let $\mathbf{y}$ denote the vector of y_i values, $\mathbf{X}$ the $n_x(p+1)$ matrix of x_i values, p the vector of fitted probabilities with ith element $p(x_i; \beta^{old})$ and $\mathbf{W}$ a NxN diagonal matrix of weights with diagonal element $p(x_i; \beta^{old})(1 - p(x_i; \beta^{old})$. After substituting our new definitions and simplifying, we obtain the following result:

$$\beta^{new} = (X^T W X)^{-1} X^T W z \text{ , where } z = X\beta^{old} + W^{-1}(y - p)$$

This new iterative process is referred to as *iteratively reweighted least squares* (IRLS), and is essential to defining an optimal logistic regression model for any provided set of data.

Making Predictions

Once the logistic coefficients have been determined for a logistic regression model, making predictions is as simple as plugging in predictor values into the logistic function:

$$\hat{p}(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

For any provided predictor values, we can now estimate the probability of producing a response of "Yes" or "No". This is useful for observing how numerical predictors determine our binary response as the predictor's value increases or decreases, but we can also study a qualitative predictor's impact on our probabilities.
Considering a binary predictor, we will examine how a categorical predictor can function in a logistic regression model. By letting the predictor space be determined by $0, 1$ we are able to produce two separate probabilities that deliver an instant comparison of which binary predictor increases or decreases the probability of encountering the defined response.

Regularized Logistic Regression

A closing topic for Logistic regression is the usage of the L_1 penalty found in LASSO for variable selection. We have previously seen how the logistic regression coefficients are determined, but we have yet to witness any methodology for choosing an appropriate amount of predictors. By altering the standard LASSO penalty, we have the following result for Logistic Regression:

$$max_{\beta_0,\beta}\left\{\sum_{i=1}^{n}\left[y_i(\beta_0+\beta^T x_i)-\log(1+e^{\beta_0+\beta^T x_i})\right]-\lambda\sum_{i=1}^{p}|\beta_j|\right\}$$

Linear and Quadratic Discriminant Analysis

This section discusses an alternative approach to logistic regression that is less direct but provides similar results. This new method models the distribution of the predictors X separately in each of the response classes (i.e. given Y), then uses Bayes' theorem to turn the separate models into estimates for $Pr(Y=k|X=x)$. When the distributions are assumed to be normal, the model ends up being quiet similar in form to logistic regression. Some reasons for using this alternative model as opposed to logistic regression are that the parameter estimates are more stable within classes that are well-separated, it is more stable if n is small and the distribution of the predictors is approximately normal in each of the classes, and it is a more popular method when there are two or more response classes.

In order to understand how this model works, it is essential to understand what Bayes' theorem really is and the section provides this foundation through a simple example. We want to classify a qualitative response into one of K unordered classes ($K \geq 2$). The *prior* probability that a randomly chosen observation comes from the k^{th} class is represented by π_k and $f_k(x) \equiv Pr(X=x|Y=k)$ is the *density function* of X for an observation that also comes from the k^{th} class. All of this comes together in Bayes formula below:

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

$Pr(Y=k|X=x)$ is the *posterior* probability that an observation belongs to the k^{th} class given the predictor value for that observation. If we have a random sample of Ys from the population, we can estimate π_k by computing the fraction of the training observations that belong to the k^{th} class. $f_k(X)$ can be more of a challenge to estimate but it is necessary in order to approximate Bayes' classifier, which classifies an observation to the class where $p_k(X)$ is largest and in turn has the lowest possible error rate of all classifiers. This section goes on to describe the methods to estimate $f_k(X)$, Linear and Quadratic Discriminant Analysis. Although LDA and QDA are the main focus of this chapter there are other techniques based on models for class densities such as more flexible mixtures of Gaussians that allow for nonlinear decision boundaries, general nonparametric density estimates for each class density which allow for the most flexibility, and *Naive Bayes* models that assume each of the class densities are products of marginal densities.

Before describing the two methods above a simple example is given to show how a Bayes' classifier might be estimated if given all of the necessary information and using only one predictor. Assume $f_k(x)$ is *Gaussian* where μ_k and σ_k^2 are the mean and variance parameters for the k^{th} class and that $\sigma_1^2=\cdots=\sigma_k^2$:

$$f_k(x)=\frac{1}{\sqrt{2\pi\sigma_k^2}}exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

Plugging this into the Bayes' formula we get:

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

This is the Bayes' classifier which assigns an observation to the class for which the result is largest. Taking the log of the equation and rearranging, it is equivalent to:

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

If $K = 2$ and $\pi_1 = \pi_2$, then the Bayes classifier assigns an observation to class 1 if $2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$, to class 2 otherwise. In this example, the Bayes decision boundary corresponds to the point:

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 - \mu_2}{2}$$

Since much of the information above would need to be assumed or guessed in order to make this method work, the section goes on to explore another method of approximation called the Linear Discriminant Analysis. It assumes that the observations within each class come from a normal distribution with a class-specific mean vector and a common variance. Using the same example above, this method works by estimating the unknown parameters π_k, μ_k and σ^2 using:

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

Where n is the total number of training observations and n_k is the number of training observations in the k^{th} class. Plugging these into the $\delta_k(x)$ above we get the equation below and can then compute Bayes' decision boundary.

$$\hat{\delta}_k(x) = x\frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + log(\hat{\pi}_k)$$

Next we consider the LDA classifier that involves multiple predictors. It is assumed that $X = (X_1, \ldots, X_p)$ is drawn from a *multivariate Gaussian* distribution with the same class-specific mean vector and a common covariance matrix that is common to all K classes (density below).

$$f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} exp\left(-\frac{1}{2}(x-\mu)^T\mathbf{\Sigma}^{-1}(x-\mu)\right)$$

Plugging this into the Bayes' theorem and rearranging we find that the Bayes' classifier assigns and observation $X = x$ to the class for which the equation below is largest.

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

To better understand Bayes decision boundaries and classifiers in this instance let's take a look at three Gaussian classes, each with class-specific mean vectors and a common covariance matrix. Bayes decision boundaries represent the set of values x for which $\delta_k(x) = \delta_l(x)$:

$$x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k = x^T\mathbf{\Sigma}^{-1}\mu_l - \frac{1}{2}\mu_l^T\mathbf{\Sigma}^{-1}\mu_l \qquad for \quad k \neq l$$

This gives us three boundaries because there are three *pairs of classes* among the three classes which divides the predictor space into three regions. The Bayes classifier will classify an observation according to the region in which it is located.

Just like in the single predictor example, $\mu_1, \ldots, \mu_k$, $\pi_1, \ldots, \pi_k$, and $\mathbf{\Sigma}$ will need to be estimated in much the same way.

On the topic of training error rate, the section provides two warnings. One, training error rates will usually be lower than test error rates because we specifically adjust the parameters of our model to do well on training data. The higher the ratio of parameters p to number of samples n, the more it is expected to overfit. Two, the *null* classifier will achieve an error rate that is only slightly higher than the LDA training set error rate.

In a binary classifier setting, the LDA can also create two types of errors by either incorrectly assigning an observation to one category or incorrectly assigning an observation to the other category. To determine which of these two errors has been made, a *confusion matrix* can be created. Modifying the *threshold* can also change the error rate and can be shown graphically using the *ROC curve*.

The next method of approximation is Quadratic Discriminant analysis. It is very much similar to LDA but instead assumes that each class has its own covariance matrix, i.e. $X \sim N(\mu_k, \mathbf{\Sigma}_k)$. In this case, Bayes classifier assigns an observation $X = x$ to the class for which the equation below is largest. Just like before, $\mathbf{\Sigma}_k, \mu_k$, and π_k will need to be estimated.

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

You may well be wondering why we would pick LDA or QDA and the answer lies in the bias-variance trade-off. With p predictors, estimating the covariance matrix requires estimating $p(p+1)/2$ parameters. QDA estimates a separate covariance matrix for each class, or K$\times$(estimated parameters). With LDA, there are K$\times$p linear coefficients to estimate. LDA is also a much less flexible classifier with lower variance than QDA, which can lead to improved prediction performance. On the other hand, if LDA's assumption that the K classes share a common covariance matrix is hugely off then it can suffer from high bias. Generally speaking, LDA can be the better option if there are few training observations which reduces variance. QDA can be better if the training set is large allowing for the variance of the classifier to be less of a concern or if common covariance can't be proven.

Now that both methods have been discussed, it is also important to note a compromise between the two proposed by Friendman (1989). It is similar to ridge regression and allows the user to shrink the separate covariances of QDA toward a common covariance as in LDA. The regularized covariance matrices have the form below, where $\hat{\mathbf{\Sigma}}$ is the pooled covariance matrix as used in LDA and $\alpha \in [0, 1]$ which allows a continuum of models between LDA and QDA. This α needs to be specified and can be chosen by evaluating the performance of the model on validation data or the tried and true cross-validation.

$$\hat{\mathbf{\Sigma}}_k(\alpha) = \alpha\hat{\mathbf{\Sigma}}_k + (1-\alpha)\hat{\mathbf{\Sigma}}$$

$\hat{\mathbf{\Sigma}}$ can also be shrunk towards a scalar covariance using the formula below, where γ now belongs to $[0, 1]$. Plugging in $\hat{\mathbf{\Sigma}}(\boldsymbol{\gamma})$ to the original $\hat{\mathbf{\Sigma}}$ equation leads to a more general family of covariances $\hat{\mathbf{\Sigma}}(\boldsymbol{\alpha}, \boldsymbol{\gamma})$ that are indexed by a pair of parameters.

$$\hat{\mathbf{\Sigma}}_k(\gamma) = \gamma\hat{\mathbf{\Sigma}} + (1-\gamma)\hat{\sigma}^2\boldsymbol{I}$$

The section on QDA and LDA concludes with a discussion of Reduced-Rank Linear Discriminant Analysis, or an LDA with reduced dimensions specifically less than the total number of classes minus one. Summarizing the mathematics within this section we find that Gaussian classification with common covariances leads to linear decision boundaries. When we have higher dimensions, classification can be achieved by sphering the data with respect to W (the within-class covariance), and classifying to the closest centroid ($log\pi_k$) in the sphered space. Since this specific distance from the centroid is the focus of this classification, the data can be confined to this sphered subspace. This subspace can then be further decomposed into successively optimal subspaces in terms of centroid separation.

Algorithms 6.1, 6.2, and 6.3

The goal with these three algorithms is to reduce the number of predictors used in a model. This often leads to a reduction in the variance of the estimated predictors and can also yield a model with better predictive accuracy. Fewer predictors in a model is also easier to interpret. Best Subset Selection, Forward Stepwise Selection, and Backward Stepwise Selection are methods to iteratively evaluate models that have different subsets of possible predictors and choose the model with the "best" performance.

Algorithm 6.1: Best Subset Selection

The Best Subset Selection is an algorithm that will select the best subset of all possible predictors in a model. This is the brute force method and evaluates all 2^p possible combinations of the p predictors. The methods of evaluating each model's performance are typically cross validated prediction error, Mallows' C_p, Akaike's Information Criterio (AIC), Bayesian Information Criterion (BIC), or Adjusted R^2. This algorithm can be computationally intensive because it must fit all 2^p models. The other two algorithms are refinements on this algorithm to achieve a potentially less optimal model faster.

Algorithm 6.2: Forward Stepwise Selection

This algorithm starts by considering the degenerate model with no predictors and then iteratively adds the predictor that improves the model's performance. It uses the same evaluation methods as the previous algorithm. This algorithm is not guaranteed to reach the best possible of the 2^p models because a predictor that gets chosen early in the algorithm may be less important in a complex model later in the algorithm. But this algorithm is faster because it only fits $\frac{1+p(p+1)}{2}$ models.

Algorithm 6.3: Backward Stepwise Selection

This algorithm works in the opposite direction as Forward Stepwise Selection. It starts with the model with all p predictors and iteratively removes predictors that contribute the least to performance. This uses the same evaluation methods as Best Subset and Forward Stepwise. Similar to Forward Stepwise, it is not guaranteed to find the best model containing a subset of p predictors and it also only fits $\frac{1+p(p+1)}{2}$ models.