Econ 425, Week 5
Classification

Manu Navjeevan

February 2, 2021

1 Binary Classification

Suppose I am interested in predicting a binary variable $Y \in \{0, 1\}$ using some (real-valued) explanatory variables $\mathbf{X} = \big(1, X_1, \ldots, X_p\big)$. For example, in the exercise we will go over in class, we will use the `penguins` dataset from `seaborn`.

A naive approach to this problem would be to use linear regression to predict Y, and interpret the $\widehat{Y}(\mathbf{X})$ value from linear regression as an estimate of the probability that $Y = 1$ give the explanatory variables $\mathbf{X}$. That is, we could try to estimate a model of the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p \tag{1}$$

And estimate $\widehat{\Pr}\big(Y = 1|\boldsymbol{X}\big) = \widehat{\beta}_0 + \widehat{\beta}_1 X_1 + \cdots + \widehat{\beta}_p X_p$. However, if we are given a large (or sufficiently) enough value of, say X_1, then we can make $\widehat{\Pr}(Y = 1|\boldsymbol{X})$ arbitrarily large or small. That is, we can $\widehat{\Pr}(Y = 1|\boldsymbol{X}) > 1$ or $\widehat{\Pr}(Y = 1|\boldsymbol{X}) < 0$, both nonsensical results.

Of course, from a machine learning perspective, this isn't *necessarily* a bad thing. We could still try and make a classification under the rule:

$$\widehat{Y} = \begin{cases} 1 & \text{if } \widehat{\Pr}(Y = 1|\boldsymbol{X}) \geq 0.5 \\ 0 & \text{if } \widehat{\Pr}(Y = 1|\boldsymbol{X}) < 0.5 \end{cases}.$$

However, this suggests that the linear model may not be the best fit for the data.

1.1 Logistic Regression

The linear probability model described in (1) makes the implicit assumption that the true probability $\pi(\boldsymbol{X}) := \Pr(Y = 1|\boldsymbol{X})$ is linear in $\boldsymbol{X}$[1]. As we went over before, this is problematic because if $\boldsymbol{X}$ has an unbounded support (can take on any value in $\mathbb{R}^p$), this will lead to predictions that are either below zero or above one.

Instead we want to specify a functional form for $\pi(\boldsymbol{X})$ that is bounded between zero and one. In logistic regression we will assume that the conditional probability has the functional form given below:

$$\pi(\boldsymbol{X}; \boldsymbol{\beta}) = \frac{1}{1 + \exp\Big(-\big(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\big)\Big)} \tag{2}$$

In class or in prior courses, you may have been introduced to logistic regression as assuming that the log-odds $\ln\left(\frac{\pi(\boldsymbol{X})}{1-\pi(\boldsymbol{X})}\right)$ are linear. That is, you may have been taught:

$$\ln\left(\frac{\pi(\boldsymbol{X})}{1 - \pi(\boldsymbol{X})}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p.$$

[1] That is that $\pi(\boldsymbol{X}) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$

This is an equivalent to the assumption that the conditional probability is of the form given in (2), I just find the given setup easier to understand.

In any case, the specification given in (2) is nice as it is bounded between zero and 1 as well as continuously differentiable with respect to $\boldsymbol{X}$ [2]. What remains is to estimate the parameters $\beta_0, \ldots, \beta_p$ using our data.

1.1.1 Estimating the Parameters via Maximum Likelihood

One way of estimating the parameters of this model is via Maximum Likelihood.

Consider the following "Likelihood Function", for evaluating the "likelihood" or "probability" of observing our data if the data was generated from a model $\pi(\boldsymbol{X};\boldsymbol{\beta})$ with parameters $\boldsymbol{\beta} = \big(\beta_0, \ldots, \beta_p\big)$:

$$L\left(\boldsymbol{\beta} \,\middle|\, \{Y_i, \boldsymbol{X}_i\}_{i=1}^n\right) = \prod_{i=1}^n \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)^{Y_i} \left(1 - \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)\right)^{1-Y_i} \tag{3}$$

Note that for an observation where $Y_i = 1$, $\pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)^{Y_i} \left(1 - \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)\right)^{1-Y_i} = \pi(\boldsymbol{X}_i;\boldsymbol{\beta})$ whereas for an observation where $Y_i = 1$, $\pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)^{Y_i} \left(1 - \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)\right)^{1-Y_i} = 1 - \pi(\boldsymbol{X}_i;\boldsymbol{\beta})$.

In maximum likelihood estimation, we choose our parameters $\boldsymbol{\beta} = \big(\beta_0, \ldots, \beta_p\big)$ to maximize the likelihood function $L\left(\boldsymbol{\beta} \,\middle|\, \{Y_i, \boldsymbol{X}_i\}_{i=1}^n\right)$. Equivalently, since $\ln(\cdot)$ is a strictly increasing function, we can maximize the log-likelihood $\ell(\boldsymbol{\beta})$:

$$\begin{aligned}
\ell\left(\boldsymbol{\beta}\right) &= \ln L\left(\boldsymbol{\beta} \,\middle|\, \{Y_i, X_i\}_{i=1}^n\right) \\
&= \ln\left\{\prod_{i=1}^n \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)^{Y_i} \left(1 - \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right)\right)^{1-Y_i}\right\} \\
&= \sum_{i=1}^n Y_i \ln \pi\left(\boldsymbol{X}_i;\boldsymbol{\beta}\right) + (1 - Y_i)\ln(1 - \pi(\boldsymbol{X}_i;\boldsymbol{\beta}))
\end{aligned}$$

Note that this is numerically the same approach as using the "log cost function approach" we went over in class. It is just motivated a bit differently (and I believe a bit more intuitively). Moreover, in industry this will be referred to as estimating the parameters via maximum likelihood, so it useful to understand what this means and where it comes from.

After we simplify the log likelihood function more (explained in this YouTube video) and plug in the specific form for $\pi(\boldsymbol{X};\boldsymbol{\beta})$, we will get that the log-likelihood simplifies to:

$$\ell\left(\boldsymbol{\beta}\right) = \sum_{i=1}^n \left[Y_i \boldsymbol{\beta}' \boldsymbol{X}_i - \ln\left\{1 + e^{\boldsymbol{\beta}' \boldsymbol{X}_i}\right\}\right] \tag{4}$$

where $\boldsymbol{\beta}' \boldsymbol{X}_i = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$. This final log-likelihood expression in (4) is what we will numerically try to maximize.

Gradient Descent Equation (4) gives us a fairly simple expression to differentiate and get the gradient of. By direct calculation we find that

$$\nabla \ell(\boldsymbol{\beta}) = \sum_{i=1}^n \big(Y_i - \pi(\boldsymbol{X}_i;\boldsymbol{\beta})\big)\, \boldsymbol{X}_i \tag{5}$$

This can be expressed even more simply if we use the design matrix and the outcome vector. Let $\mathbf{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)' \in \mathbb{R}^{n \times p}$ denote the design matrix (feature matrix) and $\mathbf{Y} = (Y_1, \ldots, Y_n)' \in \mathbb{R}^n$ denote the

[2] It also is useful to verify that the probability is increasing in $\big(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\big)$

outcome vector. Finally, let $\tilde{\pi}(\mathbf{X};\boldsymbol{\beta}) = \big(\pi(\boldsymbol{X}_i;\boldsymbol{\beta}),\ldots,\pi(\boldsymbol{X}_n;\boldsymbol{\beta})\big)'$ denote our vector of predicted probabilities at guess $\boldsymbol{\beta}$. Then we can express the gradient:

$$\nabla\ell(\boldsymbol{\beta}) = \big[\mathbf{Y} - \tilde{\pi}(\mathbf{X};\boldsymbol{\beta})\big] \cdot \mathbf{X} \tag{6}$$

perhaps of greater practical importance, this can be directly calculated in numpy using something like:

```
pHat = logitFunction(Xtrain, beta)
grad = np.dot(Ytrain - pHat, Xtrain)
```

Where the `logitFunction` is some function that would take in $\boldsymbol{\beta}$ and the feature matrix $\mathbf{X}$ (`Xtrain`) are return the vector of predicted probabilities.