Classification

Maryclare Griffin

2024-09-26

This material is based on Chapters 2 and 4 of Introduction to Statistical Learning (ISL) and parts of Chapter 4 of Elements of Statistical Learning (ESL). We will tend to follow ISL more closely, and look to ESL for occasional additional higher level material. There may be some slight changes to the notation.

For the past several lectures, we have been focused on the supervised learning problem where where we observe inputs $\boldsymbol{X}$ and a quantitative output $\boldsymbol{y}$ and assuming

$$Y = f(X) + \epsilon,$$

where f is a fixed but unknown function of $X = X_1, \dots, X_p$ and ϵ is a mean zero random error term which is independent of X.

In practice, we often encounter a **qualitative** or **categorical** output $\boldsymbol{y}$, with elements that each take on one of M distinct values. Sometimes these distinct values are also called **classes**, and the problem of predicting a future output value or class is referred to as **classification**.

If we want to predict future values of the output, it no longer makes sense to think about minimizing

$$E[(Y - \hat{f}(X))^2]$$

when Y is qualitative. Rather, our goal will be to obtain predictions $\hat{Y}$ based on estimated functions of the inputs $\hat{f}(X)$ that correctly identify the value the observed outpout is most likely to take on, i.e. we want to obtain predictions $\hat{Y}$ from $\hat{f}(X)$ that satisfy $\hat{Y} = Y$ as often as possible. We refer to this as predictions that minimize the **error rate** and we refer to the estimated function $\hat{f}(X)$ mapping from $\hat{f}(X)$ to $\hat{Y}$ as a **classifier**.

Let $I(y_i \neq \hat{y}_i)$ refer to an indicator function that is equal to 0 if $y_i = \hat{y}_i$, i.e. if the prediction for the i-th training data point matches the observed value of the output, and 1 otherwise. We can quantify the performance of a classifier on the training data via the **training error rate**,

$$\sum_{i=1}^{n} I(y_i \neq \hat{y}_i).$$

Analagously to the setting where the output is quantitative, we can also quantify the performance of a classifier on test data that were not used to fit the classifier. Letting (x_0, y_0) refer to an arbitrary data point that has not been seen before, we define the **test error rate** as

$$I(y_0 \neq \hat{y}_0). \tag{1}$$

Again, we can imagine averaging this over all possible observations that we haven't seen before but may in the future. The best classifier will minimize the average test error rate over test data that were not used to fit the classifier.

So far the definitions of the error rate, training error rate, and test error rate are a bit awkward, because they do not explicitly depend on the $\hat{f}(X)$, rather they depend on $\hat{f}(X)$ implicitly through $\hat{Y}$. Relatedly, thinking of the output as the sum of an unknown function of the inputs and a mean zero error is no longer meaningful - what is $E[Y|X] = f(X)$ if Y is qualitative? It doesn't make sense!

Accordingly, we will make the relationship between $\hat{f}(X)$ and $\hat{Y}$ more explicit. It can be proven that (1) is minimized on average by a classifier that predicts the class that is most likely given the inputs, i.e. if $v_j = 1, \dots, v_M$ represents all of the possible values that Y can take on and if the probability that Y takes on value v_j given inputs X is known $\Pr(Y = v_j|X)$, this classifier assigns predictions $\hat{y}_0$ according to

$$\hat{y}_0 = \text{argmax}_j \Pr(Y = v_j|x_0),$$

Note that when $M = 2$, this corresponds to predicting whichever class has probability $\Pr(Y = v_j|x_0) > 0.5$.

This suggests defining

$$f_j(X) = \Pr(Y = v_j|X).$$

We now have a subscript j associated with $f(X)$ that reflects the fact that we need to define these probabilities for every possible value of Y.

Note that because $j = 1, \dots, M$ indexes all of the possible values that the output Y could take on, $\sum_{j=1}^{M} \Pr(Y = v_j) = \sum_{j=1}^{M} f_j(X) = 1$. This means that given $M - 1$ probabilities, we can always reconstruct the remaining probability. For this reason, we will sometimes models for qualitative output only specify $\Pr(Y = v_j) = f_j(X)$ and estimate $f_j(X)$ for $M - 1$ values of j, most frequently for $j = 2, \dots, M$ or $j = 1, \dots, M - 1$.

Just as in the regression setting where we could decompose the performance of an estimated classifier into reducible and irreducable, we can do something similar in the classification setting. The equivalent to irreducible error in the classification setting is the **Bayes error rate**,

$$1 - E[\max_j f_j(X)], \tag{2}$$

where the expectation is taken with respect to X. This describes the error rate even if the true probablities $f_j(X)$ were known. The Bayes error rate is equal to 0 when the **Bayes decision boundary**, which describes the set of values of X for which all possible values of the outcome are equally likely perfectly separates the outputs. This corresponds to the setting where $\max_j Pr(Y = v_j|X) = 1$ for all X and the outputs are deterministic functions of the inputs. In real life (and furthermore in this *statistical* learning class in which we are studying random outputs), the outputs are rarely deterministic functions of the inputs and the Bayes error rate is rarely 0.

Now we will introduce our first classifier which is very closely related to based on K-nearest neighbors regression, **K-nearest neighbors classification**. Like its regression counterpart, KNN classification is one of the simplest non-parametric methods for classification.

Let $\mathcal{N}_i^{(K)}$ refer to the set of K indices of observed values of the predictor that are closest to x_i. Then the KNN estimate of $f_j(x_i)$ is

$$\hat{f}_j(x_i) = \frac{1}{K} \sum_{k \in \mathcal{N}_i^{(K)}} I(y_k = v_j)$$

and the KNN classifier assigns the prediction $\hat{y}_i = v_j$ where v_j is the value of the output associated with the highest estimated probability $\hat{f}_j(x_i)$

Again, the value of K is chosen by the user and determines the bias and variance of the estimate of f. Smaller K correspond to less biased but more variable estimates, whereas larger K correspond to (potentially) more biased and less variable estimates.

Just as in the regression setting, the performance of KNN classifiers depends on how K is chosen and if K is chosen to be too small, KNN classifiers can overfit the data. Furthermore, the performance of KNN classifiers also deteriorates rapidly as p, the number of predictors and/or dimension of X, increases. This is because it is harder to find neighbors in high dimensions. This is related to the idea of the **curse of dimensionality**.

This leads us to parametric classifiers. In what follows we will mainly consider the **binary** setting When Y is **binary** with $M = 2$ levels, we will refer to the levels of Y as equal to $v_1 = 0$ or $v_2 = 1$, specify

$\Pr(Y = 1) = f_2(X)$, and drop the subscript on $f_2(X)$, letting $f(X) = f_2(X)$. We can recover $f_1(X)$ from $f(X)$ according to $f_1(X) = 1 - f(X)$. Note that the KNN classifier's estimate of $f(X)$ in this case is identical to the estimate of $f(X)$ obtained from KNN regression.

Similarly, the simplest parametric classifier is obtained by using linear regression to estimate $f(X)$. We assume $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ and estimate $\beta_0, \beta_1, \ldots, \beta_p$ by finding the values $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ that minimize the least squares criterion

$$\sum_{i=1}^{n} (y_i - b_0 - x_{i1} b_1 - \cdots - x_{ip} b_p)^2,$$

with respect to $b_0, b_1, \ldots, b_p$. Then predictions $\hat{Y}$ can be obtained by setting $\hat{Y} = I(\hat{\beta}_0 + \hat{\beta}_1 X_1 + \cdots + \hat{\beta}_p X_p > 0.5)$. As in the regression case, there is only a unique minimizer of the least squares criterion when $p < n$, i.e. we have more observations than predictors, and when the predictors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$ aren't too correlated with each other. We emphasize that this is only reasonable in the binary setting with $M = 2$ when the output is defined to take on values 0 and 1.

There are one obvious issue with using linear regression for classification in this way - it can produce nonsensical estimated probabilities that are less than 0 or greater than 1! This leads us to **logistic regression**, which is a special case of a **generalized linear model**. A generalized linear model is made up of:

- A choice of **link function** $g(\cdot)$ that relates the conditional mean of an output $E[Y|X]$ to a linear function of the inputs $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, satisfying $E[Y|X] = g(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)$;
- An assumed conditional distribution of the output given values of the input.

The parameters of a generalized linear model, including $\beta_0, \beta_1, \ldots, \beta_p$, can then be estimated using maximimum likelihood.

Logistic regression uses the link function

$$\begin{aligned} g(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p) &= \frac{\exp\{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\}}{1 + \exp\{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p\}} \\ &= \frac{1}{1 + \exp\{-\beta_0 - \beta_1 X_1 - \cdots - \beta_p X_p\}}, \end{aligned}$$

which maps the real valued linear function $\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$ to a number between 0 and 1. This is sometimes called the **logistic** or **sigmoid** function. Logistic regression also assumes that the outputs Y are Bernoulli random variables conditional on the inputs X,

$$Y|X \overset{indep.}{\sim} \text{Bernoulli}\left(\frac{1}{1 + \exp\{-\beta_0 - \beta_1 X_1 - \cdots - \beta_p X_p\}}\right).$$

When we use a nonlinear model, we can still interpret nonzero β_j for $j > 0$ as indicating the presence of a relationship between the outpout and the input, but the interpretation of individual β_j for $j > 0$ changes. The interpretation will now describe how the log odds of $Y = 1$ change with changes in individual inputs X_j, where the log odds are

$$\log\left(\frac{\Pr(Y = 1|X)}{1 - \Pr(Y = 1|X)}\right).$$

Personally, I don't find log odds very interpretable beyond the fact that higher log odds correspond to $Y = 1$ being more likely and lower log odds correspond to $Y = 0$ being more likely. Nonetheless, the interpretation of any β_j for $j > 0$ will be the expected change in log odds of $Y = 1$ associated with a one unit change in X_j, holding all other predictors constant. Sometimes, people point to this awkward interpretation of the regression coefficients as a reason not to use logistic regression but that is beyond the scope of this class.

Estimates of the logistic regression coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ can then be obtained by maximizing the log likelihood

$$\begin{aligned}\sum_{i=1}^{n} \log\left(p\left(y_i|x_i\right)\right) = \sum_{i=1}^{n} & y_i \log\left(\frac{1}{1+\exp\{-b_0 - b_1X_1 - \cdots - b_pX_p\}}\right) + \\ & (1-y_i)\log\left(1 - \frac{1}{1+\exp\{-b_0 - b_1X_1 - \cdots - b_pX_p\}}\right) \\ = \sum_{i=1}^{n} & y_i \log\left(\frac{1}{1+\exp\{-b_0 - b_1X_1 - \cdots - b_pX_p\}}\right) + \\ & (1-y_i)\log\left(\frac{\exp\{-b_0 - b_1X_1 - \cdots - b_pX_p\}}{1+\exp\{-b_0 - b_1X_1 - \cdots - b_pX_p\}}\right)\end{aligned}$$

with respect to $b_0, b_1, \ldots, b_p$. This is a nonlinear function of $b_0, b_1, \ldots, b_p$, and we will rely on statistical software to maximize it and find the maximum likelihood estimator $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$.

This may seem new, we have already been using maximum likelihood in this class without explicitly saying it! The least squares estimator is the maximum likelihood estimator obtained by assuming that the outpout is conditionally normal given the inputs with mean $\beta_0 + \beta_1X_1 + \cdots + \beta_pX_p$ and constant variance!

A great think about many commonly used generalized linear models broadly and also for logistic regression is that it has been shown that the asymptotic distribution of the estimated regression coefficients $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$ as $n \to \infty$ can be derived to be normal and centered at the truth $\beta_0, \beta_1, \ldots, \beta_p$ with a variance-covariance matrix that can be estimated from the data. This means that we can construct hypotheses tests of null hypotheses that any individual β_j is equal to a certain value or that subsets of coefficients are equal to a certain value and construct confidence intervals for the estimated regression coefficients.

Logistic regression is also straightforward to extend to qualitative outputs with $M > 2$ levels. This extension is called **multinomial logistic regression**. Let β_{kj} for $j = 0, 1, \ldots, p$ refer to the regression coefficient associated with the probability that the output Y is equal to the k-th level v_k for the intercept (if $j = 0$) or predictor X_j (if $j > 0$). Multinomial logistic regression assumes that the probability that the probability that the output Y is equal to the k-th level v_k is

$$\frac{\exp\{\beta_{k0} + \beta_{k1}X_1 + \cdots + \beta_{kp}X_p\}}{\sum_{l=1}^{m}\exp\{\beta_{l0} + \beta_{l1}X_1 + \cdots + \beta_{lp}X_p\}\}}.$$

Going back to the notation we introduced when we first started talking about qualitative outputs, this is $f_k(X)$ or $\Pr(Y = v_k|X)$. Having assumed this link function, multinomial logistic regression then assumes that the outputs Y are multinomial random variables conditional on the inputs X with size 1,

$$Y|X \overset{indep.}{\sim} \text{Multinomial}\left(1, \left(\frac{\exp\{\beta_{10} + \beta_{11}X_1 + \cdots + \beta_{1p}X_p\}}{\sum_{l=1}^{m}\exp\{\beta_{l0} + \beta_{l1}X_1 + \cdots + \beta_{lp}X_p\}\}}, \ldots, \frac{\exp\{\beta_{M0} + \beta_{M1}X_1 + \cdots + \beta_{Mp}X_p\}}{\sum_{l=1}^{m}\exp\{\beta_{l0} + \beta_{l1}X_1 + \cdots + \beta_{lp}X_p\}\}}\right)\right).$$

Interpretation of the estimated regression coefficients is related to the log odds ratio of level k versus level l,

$$\log\left(\frac{\Pr\left(Y = v_k|X\right)}{\Pr\left(Y = v_l|X\right)}\right) = \left(\beta_{k0} - \beta_{l0}\right) + \left(\beta_{k1} - \beta_{l1}\right)X_1 + \cdots + \left(\beta_{kp} - \beta_{lp}\right)X_p.$$

Note that we just see the differences of pairs of regression coefficients for different levels or classes in this expression. This is related to the fact that the multinomial probabilities must sum to 1, which means that the regression coefficients associated with one level/class are a deterministic function of the rest, and reflects the fact that the linear logistic regression model can only identify each regression coefficient up to a constant. To address this in practice, it is common to choose a **reference** or **baseline** level and set the associated regression coefficients to 0. Often the last level, level M, is chosen as a baseline, meaning that $\beta_{M0} = \beta_{M1} = \cdots = \beta_{Mp} = 0$ is assumed. Then the remaining regression coefficients β_{kj} for $k = 1, \ldots, M-1$

and $j > 0$ can be interpreted as the expected change in log odds of $Y = v_k$ relative to the reference level $Y = v_M$ associated with a one unit change in X_j, holding all other predictors constant.

Alternatively, the **softmax** coding, which does not explicitly constrain the regression coefficients can be used. However in this case it is crucial that inference be based on contrasts of the regression coefficients $\beta_{kj} - \beta_{lj}$ to be meaningful.

The probabilities $\Pr(Y = v_k|X)$ are invariant to the coding used.

As with logistic regression for binary outputs, the parameters $\hat{\beta}_{k0}, \hat{\beta}_{k1}, \ldots, \hat{\beta}_{kp}$ are estimated using maximum likelihood and there is theory that supports the construction of normal-based confidence intervals and hypothesis tests of the value(s) of the regression coefficients.

Now we will consider alternative **generative** models for classification that:

- Are more stable than linear or logistic regression based methods when classes or levels of the output are "well separated" by the inputs;
- Can be more accurate if the distribution of the inputs conditional on the output is approximately normal;
- Can also easily be used for qualitative outputs with $M > 2$ levels.