CS 5565 - Intro to Statistical Learning

Lecture 3: Classification

Adu Baffour, PhD

Lecture Objectives

- Compare and contrast classification with linear regression.
- Perform classification using logistic regression.
- Perform classification using linear discriminant analysis (LDA).
- Perform classification using naive Bayes.
- Perform classification using K-Nearest Neighbors.
- Identify the strengths and weaknesses of the various classification models.

Classification

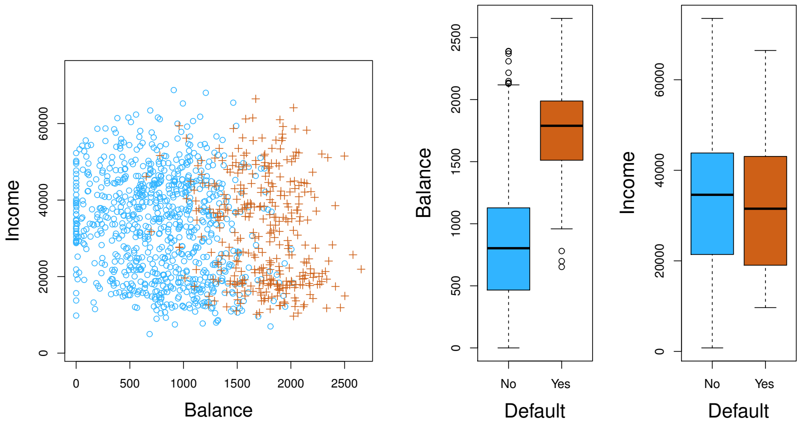
- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:
  `eye color`$\in \{$`brown`, `blue`, `green`$\}$
  `email`$\in \{$`spam`, `ham`$\}$.
- Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y; i.e. $C(X) \in \mathcal{C}$.
- Often we are more interested in estimating the *probabilities* that X belongs to each category in $\mathcal{C}$.

Classification

- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:
  `eye color`$\in \{$`brown`, `blue`, `green`$\}$
  `email`$\in \{$`spam`, `ham`$\}$.
- Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y; i.e. $C(X) \in \mathcal{C}$.
- Often we are more interested in estimating the *probabilities* that X belongs to each category in $\mathcal{C}$.

  For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification fraudulent or not.

Example: Credit Card Default



Can we use Linear Regression?

Suppose for the `Default` classification task that we code

$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of Y on X and classify as `Yes` if $\hat{Y} > 0.5$?

Can we use Linear Regression?

Suppose for the `Default` classification task that we code

$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of Y on X and classify as `Yes` if $\hat{Y} > 0.5$?

- In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to *linear discriminant analysis* which we discuss later.
- Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.

Can we use Linear Regression?

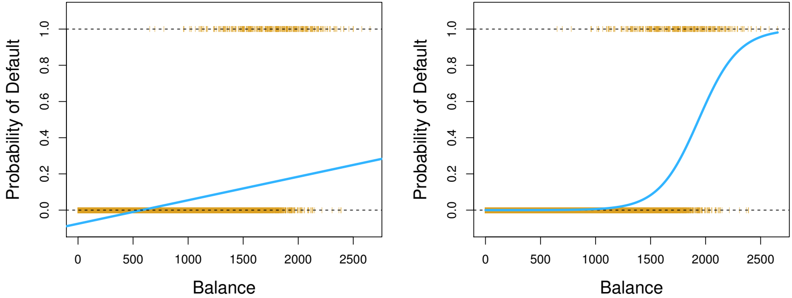
Suppose for the `Default` classification task that we code

$$Y = \begin{cases} 0 & \text{if } \texttt{No} \\ 1 & \text{if } \texttt{Yes}. \end{cases}$$

Can we simply perform a linear regression of Y on X and classify as `Yes` if $\hat{Y} > 0.5$?

- In this case of a binary outcome, linear regression does a good job as a classifier, and is equivalent to *linear discriminant analysis* which we discuss later.
- Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.
- However, *linear* regression might produce probabilities less than zero or bigger than one. *Logistic regression* is more appropriate.

Linear versus Logistic Regression



The orange marks indicate the response Y, either 0 or 1. Linear regression does not estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.

Linear Regression continued

Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

This coding suggests an ordering, and in fact implies that the difference between `stroke` and `drug overdose` is the same as between `drug overdose` and `epileptic seizure`.

Linear Regression continued

Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases}$$

This coding suggests an ordering, and in fact implies that the difference between `stroke` and `drug overdose` is the same as between `drug overdose` and `epileptic seizure`.

Linear regression is not appropriate here.
Multiclass Logistic Regression or *Discriminant Analysis* are more appropriate.

Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using `balance` to predict `default`. Logistic regression uses the form

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

($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.

Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using `balance` to predict `default`. Logistic regression uses the form

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

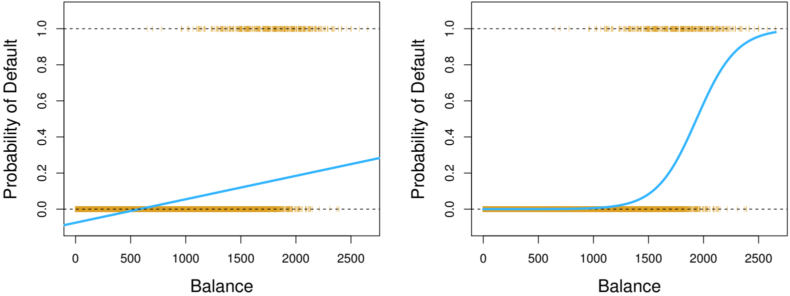
($e \approx 2.71828$ is a mathematical constant [Euler's number.])
It is easy to see that no matter what values β_0, β_1 or X take, $p(X)$ will have values between 0 and 1.

A bit of rearrangement gives

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

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$. (by log we mean *natural log*: ln.)

Linear versus Logistic Regression



Logistic regression ensures that our estimate for $p(X)$ lies between 0 and 1.

Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Maximum Likelihood

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Most statistical packages can fit linear logistic regression models by maximum likelihood. In `R` we use the `glm` function.

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.6513	0.3612	-29.5	< 0.0001
`balance`	0.0055	0.0002	24.9	< 0.0001

Making Predictions

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0 + \hat{\beta}_1 X}}{1 + e^{\hat{\beta}_0 + \hat{\beta}_1 X}} = \frac{e^{-10.6513 + 0.0055 \times 1000}}{1 + e^{-10.6513 + 0.0055 \times 1000}} = 0.006$$

Making Predictions

What is our estimated probability of `default` for someone with a balance of \$1000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of \$2000?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

Lets do it again, using `student` as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-3.5041	0.0707	-49.55	< 0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

Lets do it again, using `student` as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-3.5041	0.0707	-49.55	< 0.0001
`student[Yes]`	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default}=\texttt{Yes}|\texttt{student}=\texttt{No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Logistic regression with several variables

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

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	Z-statistic	P-value
`Intercept`	-10.8690	0.4923	-22.08	< 0.0001
`balance`	0.0057	0.0002	24.74	< 0.0001
`income`	0.0030	0.0082	0.37	0.7115
`student[Yes]`	-0.6468	0.2362	-2.74	0.0062

Why is coefficient for `student` negative, while it was positive before?

Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.

Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.
(The *mathier* students will recognize that some cancellation is possible, and only $K - 1$ linear functions are needed as in 2-class logistic regression.)

Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package `glmnet`) has the symmetric form

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Here there is a linear function for *each* class.
(The *mathier* students will recognize that some cancellation is possible, and only $K - 1$ linear functions are needed as in 2-class logistic regression.)

Multiclass logistic regression is also referred to as *multinomial regression*.

Discriminant Analysis

Here the approach is to model the distribution of X in each of the classes separately, and then use *Bayes theorem* to flip things around and obtain $\Pr(Y|X)$.

When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis.

However, this approach is quite general, and other distributions can be used as well. We will focus on normal distributions.

Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k | X = x) = \frac{\Pr(X = x | Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probabilistic modeling. Here we focus on a simple result, known as Bayes theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

One writes this slightly differently for discriminant analysis:

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

- $f_k(x) = \Pr(X = x|Y = k)$ is the *density* for X in class k. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or *prior* probability for class k.

Why discriminant analysis?

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable. Linear discriminant analysis does not suffer from this problem.
- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is again more stable than the logistic regression model.
- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.

LDA on Credit Data

		True Default Status		
		No	Yes	Total
Predicted	No	9644	252	9896
Default Status	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting.

LDA on Credit Data

		True Default Status		
		No	Yes	Total
Predicted	No	9644	252	9896
Default Status	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since $n = 10000$ and $p = 2$!

LDA on Credit Data

		True Default Status		
		No	Yes	Total
Predicted	No	9644	252	9896
Default Status	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since $n = 10000$ and $p = 2$!
- If we classified to the prior — always to class `No` in this case — we would make 333/10000 errors, or only 3.33%.

LDA on Credit Data

		True Default Status		
		No	Yes	Total
Predicted	No	9644	252	9896
Default Status	Yes	23	81	104
	Total	9667	333	10000

$(23 + 252)/10000$ errors — a 2.75% misclassification rate!

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since $n = 10000$ and $p = 2$!
- If we classified to the prior — always to class `No` in this case — we would make 333/10000 errors, or only 3.33%.
- Of the true `No`'s, we make $23/9667 = 0.2\%$ errors; of the true `Yes`'s, we make $252/333 = 75.7\%$ errors!

Types of errors

False positive rate: The fraction of negative examples that are classified as positive — 0.2% in example.

False negative rate: The fraction of positive examples that are classified as negative — 75.7% in example.

We produced this table by classifying to class `Yes` if

$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq 0.5$$

We can change the two error rates by changing the threshold from 0.5 to some other value in $[0, 1]$:

$$\widehat{\Pr}(\texttt{Default} = \texttt{Yes}|\texttt{Balance}, \texttt{Student}) \geq \textit{threshold},$$

and vary *threshold*.

K-Nearest Neighbors

K-Nearest Neighbors (KNN) is a popular nonparametric classifier method.

Given a positive integer K and some test observation, the KNN classifier identifies the K points in the training data that are closest to the test observation. These closest K points are represented by N_0.

Then, it estimates the conditional probability for a class as the fraction of points in N_0 that represent that specific class.

Lastly, KNN will apply the Bayes' rule and classify the test observation to the class with the largest probability.

However, the choice of the K value is very important. Lower values are more flexible, whereas higher values are less flexible but have more bias. Similar to the regression setting, a bias-variance tradeoff exists.

Summary

- Qualitative variables, such as gender, are known as categorical variables. Predicting qualitative responses is known as classification.
- The logistic function is used to model the relationship between the probability (Y) and some predictor (X) because the function falls between 0 and 1 for all X values.
- The coefficients are estimated through the maximum likelihood method.
- Linear discriminant analysis is an alternative approach to classification that models the distributions of the different predictors separately in each of the response classes (Y), and then uses Bayes' theorem to flip these around into estimates.
- K-Nearest Neighbors (KNN) is a popular nonparametric classifier method.

Thank you.

Any questions?

Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative

Evaluation Metrics

- Accuracy: the proportion of correct predictions among the total number of test samples.
  Accuracy = (TP+TN)/(TP+FP+FN+TN)

- Precision: explains how many of the correctly predicted samples turned out to be positive.
  Precision = TP/TP+FP

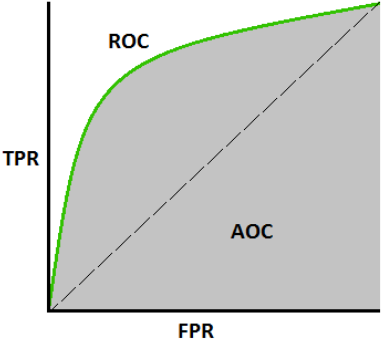
- Recall: explains how many of the actual positive cases we were able to predict correctly with our model.
  Recall = TP/TP+FN

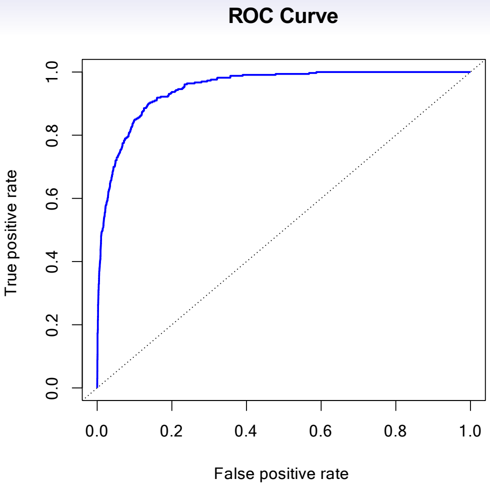
- F1-Score: It is the harmonic mean of precision and recall.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Evaluation Metrics

AUC-ROC: The Receiver Operator Characteristic (ROC) is a probability curve that plots the TPR(True Positive Rate) against the FPR(False Positive Rate) at various threshold values.





The *ROC plot* displays both simultaneously.

Sometimes we use the *AUC* or *area under the curve* to summarize the overall performance. Higher *AUC* is good.