Classification and Logistic Regression

UNIVERSITY OF SAN FRANCISCO

James D. Wilson

MSAN 601: Linear Regression

Plan for this Lecture

- The classification problem
- Why not regression?
- Assessing model accuracy
  - Mean squared error and accuracy
  - Receiver Operating Curves (ROCs)

Reference: ISL Sections 2.2.3; 4.1; 4.2; 4.4.3

The Classification Setting

Data: Consisting of n observations $(x_1, y_1), \ldots, (x_n, y_n)$ with

- $x_i \in \mathcal{X}$ space of predictors (often $\subseteq \mathbb{R}^p$)
- $y_i \in \mathcal{C}$: response or class label
  - Binary classification: $\mathcal{C} = \{-1, +1\}$ (or equivalently, $\{0, 1\}$)
  - Multi-class classification: $\mathcal{C} = \{0, 1, \ldots, m\}$

Unlike regression, the observed labels are *categorical* or *qualitative*.

Classification

Goal: Given an unlabeled vector x, assign it to class $c \in \mathcal{C}$.

Prediction Rule / Classifier

A prediction rule or classifier is a map

$$\phi : \mathcal{X} \to \mathcal{C}$$

$$\phi(x) = c \in \mathcal{C}$$

Regard $\phi(x) = c \in \mathcal{C}$ as a prediction of the class label associated with the predictor x.

Examples

Medical Tests:

- $x \in \mathbb{R}^p$ contains the (numerical) results of p diagnostic tests
- y = illness / condition

Object Recognition:

- $x \in \mathbb{R}^p$ contains the pixel intensities from a satellite image
- $y = +1$ if image contains a man-made object, $y = -1$ otherwise

Examples

Automatic Spam Recognition:

- x = vector of features extracted from text of email, e.g.,
  - presence of keywords (“cheap”, “cash”, “medicine”)
  - presence of key phrases (“Dear Sir/Madam”)
  - use of words in all-caps (“VIAGRA”)
  - point of origin of email
- $y = +1$ if email is spam, $y = -1$ otherwise

Examples

Credit Card Default

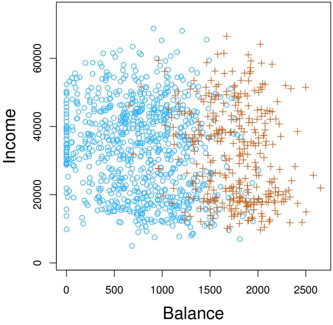


Figure: The annual incomes and monthly credit card balances of a group of individuals. Orange: defaulted on credit card payments; Blue: did not default.

Key Considerations

1. Why not use regression?
2. Measuring the loss/error of a prediction
3. Assessing the overall performance of a prediction rule
4. Identifying the optimal prediction rule

Why Not Use Regression?

Consider a simple example where Doctors are trying to predict the medical condition of a patient. Here,

$$y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$

- Regression assumes that there is a meaning behind the *ordering* of y and that a change in levels above suggest the *same* change.
- Typically, however, categorical variables have no natural order and there is no way to quantify a "jump" from one level to another.

Why Not Use Regression?

Consider a simple example where Doctors are trying to predict the medical condition of a patient. Here,

$$y = \begin{cases} 1 & \text{if stroke} \\ 2 & \text{if drug overdose} \\ 3 & \text{if epileptic seizure} \end{cases}$$

- Regression models y *directly* – therefore, estimates will be continuous values in $(-\infty, \infty)$
- Prediction rules are often concerned with *the probability* of each value of y

Measuring the Loss of a Prediction

Let $\phi : \mathcal{X} \to \mathcal{C}$ be a prediction/classification rule of interest

Question: Given a pair (x, y), how do we compare $\phi(x)$ and y?

Namely, how do we measure the accuracy of $\phi(x)$?

Common to use the Zero-One Loss Function $\ell(\phi(x), y)$:

$$\ell(\phi(x), y) = \begin{cases} 1 & \text{if } \phi(x) \neq y \\ 0 & \text{if } \phi(x) = y \end{cases}$$

Note: Two types of errors $\phi(x) = 1, y = 0$ and $\phi(x) = 0, y = 1$ given equal weight

Expected Loss

Given: Zero-one loss of prediction rule $\phi : \mathcal{X} \to \mathcal{C}$ given by

$$\ell(\phi(x), y) = \mathbb{I}(\phi(x) \neq y)$$

We typically measure performance of ϕ by its expected loss (risk)

$$R(\phi) = \mathbb{E}[\ell(\phi(x), y)]$$

Important: Note that

$$R(\phi) = \mathbb{E}[\mathbb{I}(\phi(x) \neq y)] = \mathbb{P}(\phi(x) \neq y)$$

is just the probability that ϕ misclassifies a sample.

Measuring Accuracy

Accuracy

The accuracy of a classifier $\phi(x)$ is:

$$1 - R(\phi) = \mathbb{P}(\phi(x) = y)$$

Important Notes:

- In practice, we measure the empirical probability of misclassification over a data set with n observations using:

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}(y_i \neq \phi(x_i))$$

- If $y \in \{0, 1\}$, the empirical misclassification rate = MSE(ϕ).
- Training and test set evaluations still apply!

Issues with Measuring Accuracy Only

Example:

Paypal claims that its fraud rate is less than 0.5%. Suppose that you are hired to create a classifier that distinguishes fraudulent transactions from non-fraudulent transactions. How might you classify new transactions?

Issues with Measuring Accuracy Only

Example:

Paypal claims that its fraud rate is less than 0.5%. Suppose that you are hired to create a classifier that distinguishes fraudulent transactions from non-fraudulent transactions. How might you classify new transactions?

Let $y_i = -1$ if the transaction is fraudulent and $y_i = +1$ otherwise. A great classifier (perhaps the best) according to MSE / accuracy is choosing $\phi(x_i) = +1$ for all i. Indeed, your MSE would be ~ 0.005.

Result: You never detect any of the fraudulent transactions!

The above is a typical example of unbalanced data.

Informative Model Assessment

Let $y_i \in \{-1, +1\}$ (binary classification). ϕ = proposed classifier.

- True positives (TP):

$$\sum_{i=1}^{n} \mathbb{I}(y_i = \phi(x_i) = +1)$$

- False positives (FP):

$$\sum_{i=1}^{n} \mathbb{I}(y_i = -1; \phi(x_i) = +1)$$

- True negatives (TN):

$$\sum_{i=1}^{n} \mathbb{I}(y_i = \phi(x_i) = -1)$$

- False negatives (FN):

$$\sum_{i=1}^{n} \mathbb{I}(y_i = +1; \phi(x_i) = -1)$$

Model Assessment Relationships

- Accuracy $= \dfrac{\text{TP} + \text{TN}}{n} \in [0, 1]$
- The sensitivity (or recall) of ϕ is:

$$\frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{TP}}{\sum_{i=1}^{n} \mathbb{I}(y_i = +1)} \in [0, 1]$$

- The specificity of ϕ is:

$$\frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{TN}}{\sum_{i=1}^{n} \mathbb{I}(y_i = -1)} \in [0, 1]$$

- The precision of ϕ is:

$$\frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{TP}}{\sum_{i=1}^{n} \mathbb{I}(\phi(x_i) = +1)} \in [0, 1]$$

Model Assessment

To understand the performance of a classifier, we can use a confusion matrix which portrays the FN, TN, FP, TP rates.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)
	Predicted condition negative	**False negative** (Type II error)	**True negative**

Figure: From Wikipedia.org

Model Choice

Choice of Model: depends on the context and constraints

Back to the Paypal problem: Suppose there are 100K transactions

	$y_i = +1$	$y_i = -1$
$\phi(x_i) = +1$	99500	500
$\phi(x_i) = -1$	0	0

Summary: TN = FN = 0; TP = 99500; FP = 500

Accuracy = precision = 0.995; sensitivity = 1; specificity = 0

Result: If we are concerned with identifying fraud, we want specificity to be close to 1. In this case, our model performs terribly.

Back to Regression

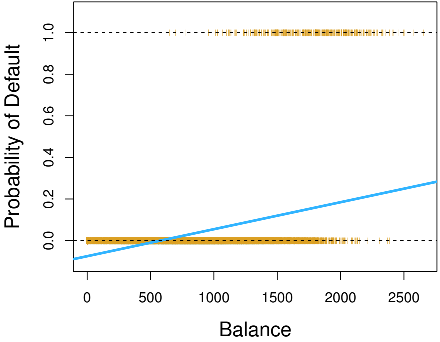
Setting: Y is binary, namely $Y \in \{-1, +1\}$ and fixed predictors $X \in \mathbb{R}^p$

Question: How can we model $\mathbb{P}(Y = +1 \mid X = \mathbf{x})$ as a function of $\mathbf{x}$?

Standard Regression setting: We could use a linear model

$$\mathbb{P}(Y = +1 \mid X = \mathbf{x}) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

but...



...using a linear model provides some values of $\mathbb{P}(Y = +1 \mid X = \mathbf{x})$ outside of 0 to 1!

The Logistic Function

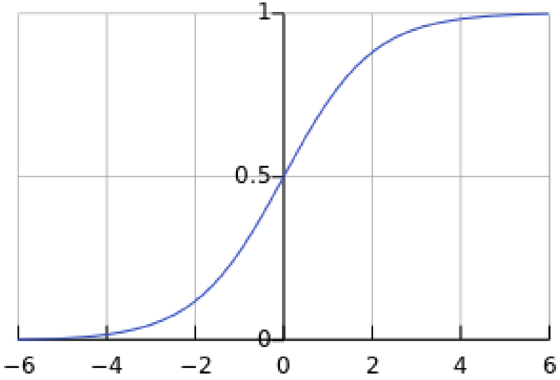
Instead, we can use a different model to ensure $\mathbb{P}(Y = +1 \mid X = \mathbf{x})$ is between 0 and 1:

$$\mathbb{P}(Y = +1 \mid X = \mathbf{x}) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}$$

Here, $f(x) = \dfrac{e^x}{1 + e^x} \in (0, 1)$ is called the logistic function of x.

The Logistic Function

$$f(x) = \frac{e^x}{1 + e^x}$$



The model

$$\mathbb{P}(Y = +1 \mid X = \mathbf{x}) = \frac{\exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}{1 + \exp\left(\beta_0 + \sum_{j=1}^{p} \beta_j x_j\right)}$$

can be rearranged and equivalently stated as:

$$\log\left(\frac{P(Y = +1 \mid X = \mathbf{x})}{1 - P(Y = +1 \mid X = \mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

The above model is called the logistic regression of Y on $X = \mathbf{x}$

Logistic Regression

Model:

$$\log\left(\frac{P(Y=+1 \mid X=\mathbf{x})}{1-P(Y=+1 \mid X=\mathbf{x})}\right) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

Features:

- $\dfrac{P(Y=+1 \mid X=\mathbf{x})}{1-P(Y=+1 \mid X=\mathbf{x})}$ is known as the odds of Y taking value $+1$
- $\log($odds$)$ is known as the logit or log-odds of Y taking value $+1$.
- The right hand side is linear in $\mathbf{x}$

Logistic Regression: Interpretation

Holding all other variables constant, increasing x_j by one unit changes the log odds of $Y = +1$ by β_j. Equivalently, increasing x_j by one unit multiplies the odds of $Y = +1$ by e^{β_j}.

Inference:

- $\beta_j < 0$: the odds of $Y = +1$ is decreased $\Rightarrow$ the probability of $Y = +1$ is decreased
- $\beta_j > 0$: the odds of $Y = +1$ is increased $\Rightarrow$ the probability of $Y = +1$ is increased
- $\beta_j = 0$: no effect on chances of $Y = +1$

Estimation via Maximum Likelihood

Goal:

- Estimate $\beta_0, \ldots, \beta_p$ via maximum likelihood
- Estimate $\mathbb{P}(Y = +1 \mid X = \mathbf{x})$ by plugging in the above estimates into the logistic function

Methodology: Identify $\hat{\beta}_0, \ldots, \hat{\beta}_p$ that maximizes the likelihood:

$$L(\beta \mid Y = y) = \prod_{i=1}^{n} \mathbb{P}(Y = +1 \mid X = \mathbf{x}_i)^{y_i} \mathbb{P}(Y = 0 \mid X = \mathbf{x}_i)^{1-y_i}$$

Important Fact: The maximum likelihood estimate (MLE) of $\hat{\beta}_j$ has an *approximate* Gaussian distribution with mean β_j. Therefore, statistical inference can be conducted the same as OLS.

Estimation via Maximum Likelihood

Methodology, continued... Equivalently, we can maximize the log-likelihood of $\beta_0, \beta \mid Y = y$, which we can simplify as follows:

$$\begin{aligned}
\ell(\beta_0, \beta \mid Y = y) &= \log(L(\beta_0, \beta \mid Y = y)) \\
&= \sum_{i=1}^{n}[y_i \log(\mathbb{P}(Y = +1 \mid X = \mathbf{x}_i)) \\
&+ (1 - y_i)\log(\mathbb{P}(Y = 0 \mid X = \mathbf{x}_i))] \\
&= \sum_{i=1}^{n} \exp(\beta_0 + \mathbf{x}_i^T \beta) + \sum_{i=1}^{n}[y_i(\beta_0 + \mathbf{x}_i^T \beta)]
\end{aligned}$$

Estimation via Maximum Likelihood

There is no analytical form for $\hat{\beta}$ that maximizes the log-likelihood (unlike OLS for standard regression). So, we must resort to a computational means using methods like:

- Gradient descent methods for each β_j or
- Fisher scoring algorithm

Once we obtain $\hat{\beta}$, we can calculate:

$$\hat{\mathbb{P}}(Y = +1 \mid X = \mathbf{x}) = \frac{\exp\left(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j\right)}{1 + \exp\left(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_j\right)}$$

Logistic Regression

The binary classifier is defined as

$$\phi(x) = \text{argmax}_j\{\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x})\}$$

Or, equivalently,

$$\phi(x) = \text{argmax}_j\{\text{logit}\left(\hat{\mathbb{P}}(Y = j \mid X = \mathbf{x})\right)\}$$

Hence, the discriminant function is given by

$$\delta_j(\mathbf{x}) = (-1)^{j-1}(\hat{\beta}_0 + \sum_{i=1}^{p} \hat{\beta}_i x_i)$$

Conclusion: Logistic regression gives a linear discriminant!

Summary of Logistic Regression

- Inference-based: β_j describes the multiplicative effect of x_j on the odds of $Y = +1$
- For binary classification only! (though there are multi-class extensions)
- Provides linear discriminants $\delta_j = \log(\mathbb{P}(Y = j \mid X = \mathbf{x}))$
- Estimates found via maximum likelihood + gradient descent / Fisher scoring algorithms

Classification via Thresholding

Issue: In binary classification settings, we often choose a class using a threshold τ. For example, we choose $Y = +1$ if

$$P(Y = +1 \mid X = \mathbf{x}) > \tau$$

So far, we've typically used $\tau = 0.5$.

Point: The error rate will change based on the threshold value τ that we choose.

Question: How can we assess the performance of a method based on τ?

Receiver Operating Characteristics (ROC)

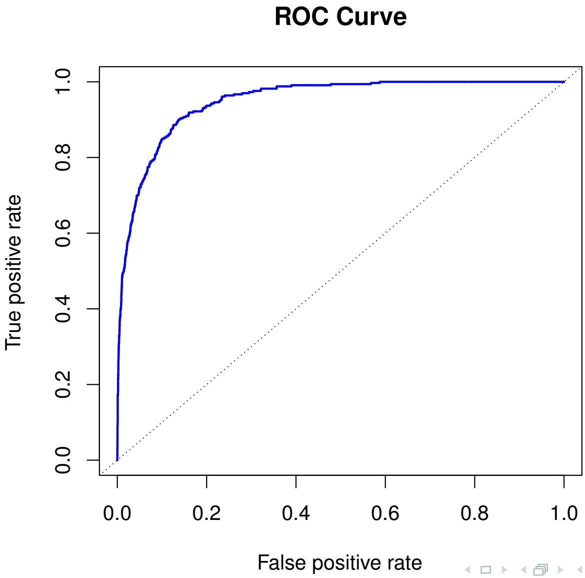
The receiver operating characteristics (ROC) of a binary classifier are the

- true positive rate (sensitivity)
- false positive rate (1 - specificity)

for the classifier across a grid of the threshold τ.

The ROC curve plots the comparison of these two quantities across τ.

ROC curve


ROC Curve
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

The Area Under the Curve (AUC)

The area under the curve (AUC) is the area under the ROC curve.

Features:

- AUC $\in [0, 1]$
- Measures the overall performance of a classifier
- The higher the better.
- We expect a classifier that performs no better than chance to have an AUC of 0.5 on an independent test set.