15
Logistic Regression

Let's return to the problem of binary classification where a feature vector $\mathbf{x}$ is used to predict the class y of a sample. We've already used the Support Vector Machine to solve the binary classification problem. Recall that the SVM uses optimization to find a hyperplane $\mathbf{a} \cdot \mathbf{x} = b$ that separates the two classes. Although the SVM works well, it is difficult to understand how the algorithm predicts the class of new data. We could try to examine the support vectors that lie nearest the separating hyperplane, but in general we cannot directly interpret SVM models.

By contrast, we've seen how straightforward it is to interpret linear statistical models. We are able to assign meaningful interpretations to the fitted coefficients, and the relative importance of the predictor variables is quantified by the statistical outputs of the `fitlm` function. Ideally we would use linear models for the binary classification problem. However, there are two problems:

- The predictions of a classification algorithm are binary, while linear models make continuous predictions.
- Even if we force a linear model to make discrete predictions, we must also force the outputs of a linear model to stay within the set of classes.

In this chapter we develop a variant of linear regression called *logistic regression*. Logistic regression uses a linear model to predict binary outcomes. Rather than predict the class of a sample directly, a logistic regression model predicts the probability that the sample is in each class. We will show how a *link function* can be used to map the output of a linear model into a bounded range, like the interval $[0, 1]$ for probabilities.

Predicting the Odds

You're probably familiar with probabilities as the long-run expectation

Pun intended.

of an uncertain process. Logistic regression uses a related concept called the *odds*. You may have used the term “odds" interchangeably with “probability," but they are not the same. Let’s assume that a random variable y has two possible outcomes, 0 and 1. The odds of y is the ratio of the probability that y equals 1 to the probability that y equals 0, or

$$\text{odds}(y) = \frac{P(y=1)}{P(y=0)}.$$

For example, if $\text{odds}(y) = 2$ then the probability that $y = 1$ is twice as large as the probability that $y = 0$. We can convert between probabilities and odds by remembering that probabilities sum to one, or P(y=0) + P(y=1) = 1. Then

Odds are usually expressed as a proportion, so an odds of 2 is written as 2:1, or “two to one".

$$\text{odds}(y) = \frac{P(y=1)}{P(y=0)} = \frac{P(y=1)}{1-P(y=1)} \Rightarrow P(y=1) = \frac{\text{odds}(y)}{1+\text{odds}(y)}.$$

The odds function lives interval $[0,\infty)$. The odds of y become infinite as the probability that $y = 1$ increases. The odds of y go to zero as the probability that $y = 0$ increases. This means that the logarithm of the odds, or the “log odds" is a continuous value in the interval $(-\infty,\infty)$, which is the same range as the predictions of a linear model. We can build a binary classifier by using a linear model to predict the log odds of the response variable y, i.e.

Some people go further and refer to the log odds as the “lods".

$$\log(\text{odds}(y)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

The function $L(y) = \log(\text{odds}(y))$ is called the *logit* function. Because it links the response variable to the linear models, we refer to the logit (and other similar functions) as *link functions*.

To summarize:

$$\begin{aligned} y &\in 0 \text{ or } 1 \\ P(y=1) &\in [0,1] \\ \text{odds}(y) &\in [0,\infty) \\ \log(\text{odds}(y)) &\in (-\infty,\infty) \end{aligned}$$

From Odds to Probabilities

Log odds can be predicted using linear models, but it is difficult for most people to interpret the odds, much less their logarithm. Ideally we would have our logistic regression model predict probabilities. The logistic regression model from above was

$$\log(\text{odds}(y)) = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n.$$

Exponentiating both sides to gives

$$\text{odds}(y) = e^{\beta_0+\beta_1 x_1+\cdots+\beta_n x_n} \equiv e^t,$$

where the placeholder t equals the output of the linear model. We can solve for the probability that y equals 1 using the relationship between probabilities and odds.

$$
\begin{aligned}
P(y = 1) &= \frac{\text{odds}(y)}{1 + \text{odds}(y)} \\
&= \frac{e^t}{1 + e^t} \\
&= \frac{1}{1 + e^{-t}}
\end{aligned}
$$

For the last step we divided both the numerator and denominator by e^t.

Making predictions with linear models is a two-step process. First, we use a linear model to predict the placeholder value t. The value of t is used to calculate the probability that the response is equal to one. If we were interested in classifying the response, we would say that $y = 1$ if $P(y = 1) > 0.5$ and choose $y = 0$ otherwise. Note that the point $P(y = 1) = 0.5$ occurs when $t = 0$. When classifying with a logistic regression model, our response prediction switches from class 0 to class 1 when the output of the linear model $t = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$ switches from negative to positive.

The shape of the output of a logistic regression model is sigmoidal, as shown in Figure 15.1. Logistic regression is used for binary classification, so the model should alternate between predicting class 0 and class 1. The sigmoid shape is a compromise; it is smooth and continuous but still transitions rapidly from 0 to 1. The smoothness of the link function makes it easier to fit logistic regression models.

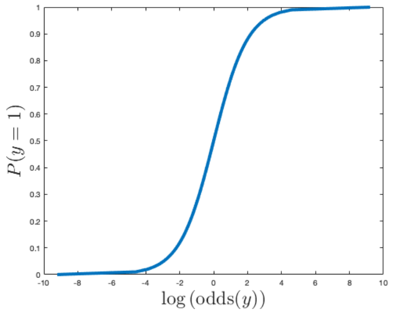
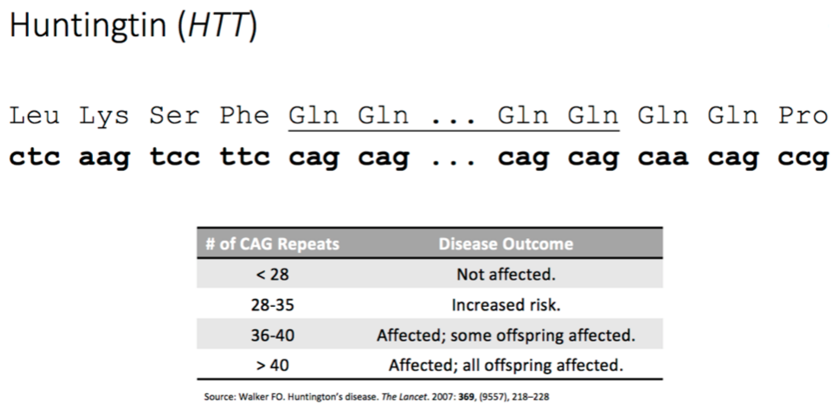


Figure 15.1: The inverse logistic function.

Example: Predicting the risk of Huntington's Disease

Huntington's Disease is an inherited genetic condition caused by repeated CAG sequences in the Huntingtin (*HTT*) gene. Too many CAG repeats create a "glutamine knot" in the protein, causing toxic protein aggregates in neurons. Symptoms of Huntington's appear later life, and an individual's risk for developing the disease correlates with the number of CAG repeats (Figure 15.2).

Let's build a model to predict the probability of developing Huntington's based on the number of CAG repeats. The response variable is binary (Huntington's disease or not) and the predictor variable is continuous (the number of CAG repeats in the *HTT* gene). To train the model we counted the number of CAG repeats in 50 individuals with and without the disease.



Huntingtin (*HTT*)

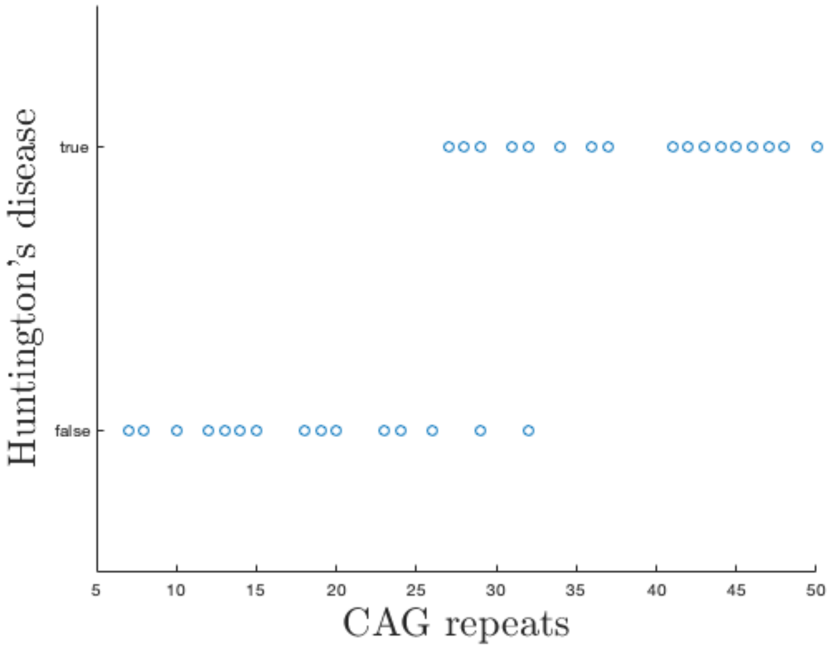
```
Leu Lys Ser Phe Gln Gln ... Gln Gln Gln Gln Pro
ctc aag tcc ttc cag cag ... cag cag caa cag ccg
```

# of CAG Repeats	Disease Outcome
< 28	Not affected.
28-35	Increased risk.
36-40	Affected; some offspring affected.
> 40	Affected; all offspring affected.

Source: Walker FO. Huntington's disease. *The Lancet*. 2007: **369**, (9557), 218–228

Figure 15.2: The risk of developing Huntington's Disease correlates with the number of CAG repeats in the *HTT* gene.

```
load huntington.mat
scatter(hunt.CAGs,categorical(hunt.disease))
xlabel('CAG repeats',axargs{:})
ylabel("Huntington's disease",axargs{:})
```



We see from these data that predicting disease status with low (<25) or high (>35) CAG repeats is straightforward. However, there is a region between 25 and 35 CAG repeats where disease status is ambiguous. Let's build a logistic regression model to predict Huntington's status. We use the MATLAB function `fitglm`, for "fit generalized linear model". The `fitglm` function is similar to `fitglm`; the first argument is a table of data, and the second argument is a formula describing the model. However, `fitglm` can use a wide range of link functions and datatypes when fitting linear models. For logistic regression using binary responses we need to specify the logit link function and a binomial distribution.

```
model = fitglm(hunt,'disease ~ CAGs','link','logit', ...
   'Distribution','binomial')
```

```
model =
Generalized linear regression model:
logit(disease) ~ 1 + CAGs
Distribution = Binomial

Estimated Coefficients:
            Estimate      SE      tStat     pValue
            ________   _______   _______   ________
(Intercept)  -14.032    5.7832   -2.4263   0.015252
CAGs.        0.50558   0.20395    2.4789   0.013179

50 observations, 48 error degrees of freedom
Dispersion: 1
Chi^2-statistic vs. constant model: 55, p-value = 1.18e-13
```

Remember that the model we're fitting is

$$\log(\text{odds}(\text{disease})) = \beta_0 + \beta_1[\text{CAGs}].$$

We know the best fit values of β_0 and β_1 from the output of the `fitglm` model:

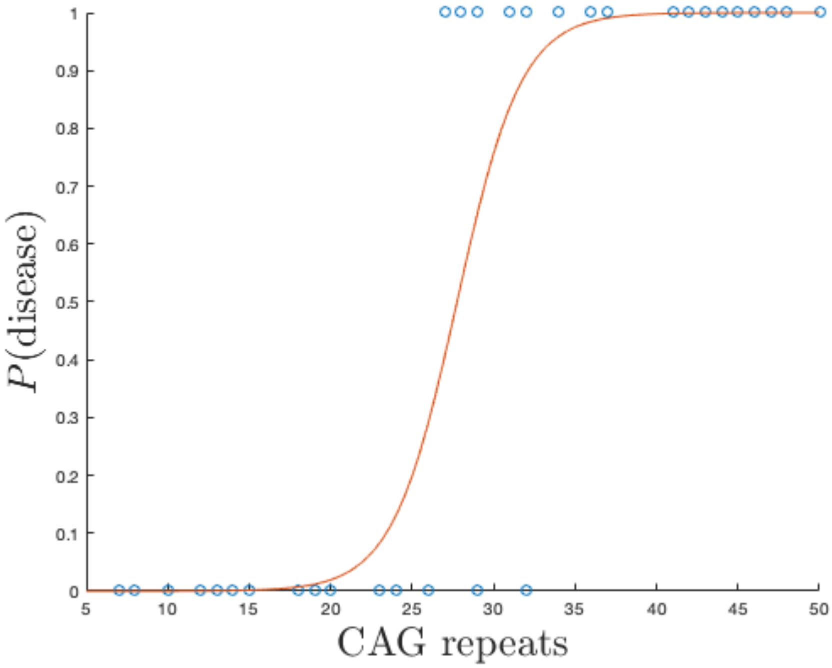
$$\log(\text{odds}(\text{disease})) = -14.032 + 0.50558[\text{CAGs}].$$

We can also rewrite this model to predict the probability of having Huntington's disease

$$P(\text{disease}) = \frac{1}{1 + e^{-14.032+0.50558[\text{CAGs}]}},$$

which we plot below along with the training data.

```
scatter(hunt.CAGs,hunt.disease)
hold on
cag_range = linspace(5,50,100);
beta = model.Coefficients.Estimate;
plot(cag_range, 1./(1+exp(-(beta(1)+beta(2)*cag_range))))
hold off
xlabel('CAG repeats',axargs{:});
ylabel('$$P(\mathrm{disease})$$',axargs{:});
```



We are often interested in the point where $P(\text{disease}) = 0.5$, as this is the threshold number of CAG repeats where a person is equally likely to have or not have Huntington's. The logistic function reaches its midpoint when the linear model moves from negative to positive. Thus we can simply solve for when $\beta_0 + \beta_1[\text{CAGs}] = 0$.

$$\begin{aligned} -14.03 + 0.51[\text{CAGs}] = 0 \Rightarrow [\text{CAGs}] &= 14.03/0.51 \\ &\approx 28 \text{ CAG repeats} \end{aligned}$$

When the output of the linear model is zero,

$$P(y=1) = \frac{1}{1+e^0} = \frac{1}{2}$$

.

Interpreting coefficients as odds ratios

The coefficients of the linear part of a logistic regression equation are not directly interpretable. The coefficients describe how the linear model changes given a unit change in the input variables, but the outputs of the linear model undergo a nonlinear transformation before becoming a probability. Instead, we interpret logistic regression models by calculating the change in odds that accompany a unit change in an input variable. This change is odds is called the *odds ratio*. For example, we can define the odds ratio that corresponds to increasing variable x_i by 1 as

$$\text{odds ratio}(x_i) = \frac{\text{odds}(x_i + 1)}{\text{odds}(x_i)}.$$

Let's calculate the odds ratio for Huntington's disease that accompanies an increase of one CAG repeat.

$$\begin{aligned}\text{odds ratio}([\text{CAGs}]) &= \frac{\text{odds}([\text{CAGs}] + 1)}{\text{odds}([\text{CAGs}])} \\ &= \frac{e^{\beta_0 + \beta_1([\text{CAGs}]+1)}}{e^{\beta_0 + \beta_1[\text{CAGs}]}} \\ &= \frac{e^{\beta_0} e^{\beta_1[\text{CAGs}]} e^{\beta_1}}{e^{\beta_0} e^{\beta_1[\text{CAGs}]}} \\ &= e^{\beta_1}\end{aligned}$$

Since $\beta_1 = 0.51$ in out model, having one more CAG repeat increases the odds of developing Huntington's disease by $e^{0.51} = 1.67$-fold. For any logistic regression model, the odds ratio for variable x_i is the exponential of the corresponding coefficient β_i.

$$\text{odds ratio}(x_i) = \frac{\text{odds}(x_i + 1)}{\text{odds}(x_i)} = e^{\beta_i}$$

If β_i is negative the odds ratio e^{β_i} will be less than one and the odds will decrease.

You may have heard news reports that "doing X increases your risk of Y". Researchers performing this type of study often use logistic regression models to predict the odds of developing condition Y based on input variable X. The reported increase in risk is simply the odds ratio associated with the coefficient of X.