Fundamentals of Data Science

Logistic regression

Ramesh Johari

Regression methods for binary outcomes

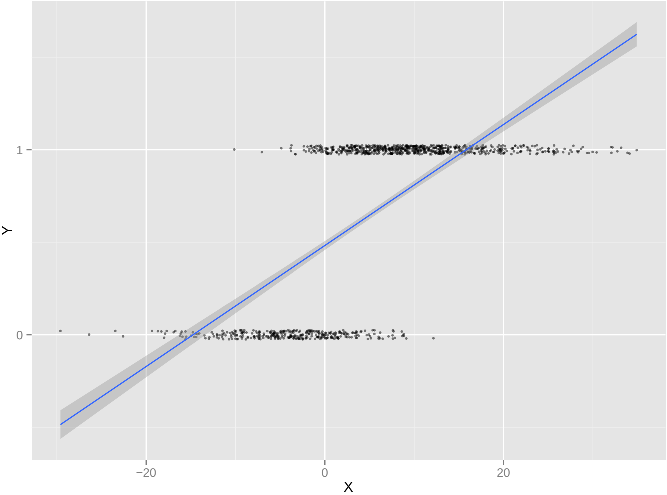
Binary outcomes

For the duration of this lecture suppose the outcome variable $Y_i \in \{0, 1\}$ for each i.

(Much of what we cover generalizes to discrete outcomes with more than two levels, but we focus on the binary case.)

Linear regression?

Why doesn't linear regression work well for prediction? A picture:



Logistic regression

At its core, logistic regression is a method that directly addresses this issue with linear regression: it produces fitted values that always lie in $[0, 1]$.

Input: Sample data $\mathbf{X}$ and $\mathbf{Y}$.

Output: A fitted model $\hat{f}(\cdot)$, where we interpret $\hat{f}(\vec{X})$ as *an estimate of the probability that the corresponding outcome Y is equal to* 1.

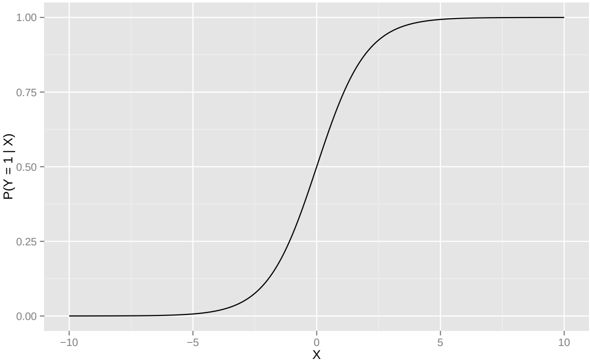
Logistic regression: Basics

The population model

Logistic regression assumes the following population model:

$$\mathbb{P}(Y = 1|\vec{X}) = \frac{\exp(\vec{X}\boldsymbol{\beta})}{1 + \exp(\vec{X}\boldsymbol{\beta})} = 1 - \mathbb{P}(Y = 0|\vec{X}).$$

A plot in the case with only one covariate, and $\beta_0 = 0, \beta_1 = 1$:



Logistic curve

The function g given by:

$$g(q) = \log\left(\frac{q}{1-q}\right)$$

is called the *logit* function. It has the following inverse, called the *logistic curve*:

$$g^{-1}(z) = \frac{\exp(z)}{1+\exp(z)}.$$

In terms of g, we can write the population model as:[1]

$$\mathbb{P}(Y = 1|\vec{X}) = g^{-1}(\vec{X}\boldsymbol{\beta}).$$

[1] This is one example of a *generalized linear model* (GLM); for a GLM, g is called the *link function*.

Logistic curve

Note that from this curve we see some important characteristics of logistic regression:

- The logistic curve is *increasing*. Therefore, in logistic regression, larger values of covariates that have *positive* coefficients will tend to increase the probability that $Y = 1$.
- When $z > 0$, then $g^{-1}(z) > 1/2$; when $z < 0$, then $g^{-1}(z) < 1/2$.
  Therefore, when $\vec{X}\boldsymbol{\beta} > 0$, Y is more likely to be one than zero; and conversely, when $\vec{X}\boldsymbol{\beta} < 0$, Y is more likely to be zero than one.

Interpretations: Log odds ratio

In many ways, the choice of a logistic regression model is a matter of practical convenience, rather than any fundamental understanding of the population: it allows us to neatly employ regression techniques for binary data.

One way to interpret the model:

- Note that given a probability $0 < q < 1$, $q/(1-q)$ is called the *odds ratio*. The odds ratio lies in $(0, \infty)$.
- So logistic regression uses a model that suggests the *log odds ratio* of Y given $\vec{X}$ is linear in the covariates. Note the log odds ratio lies in $(-\infty, \infty)$.

Interpretations: Latent variables

Another way to interpret the model is the *latent variable* approach:

- Suppose given a vector of covariates $\vec{X}$, a *logistic* random variable Z is sampled independently:

$$\mathbb{P}(Z < z) = g^{-1}(z).$$

- Define $Y = 1$ if $\vec{X}\boldsymbol{\beta} > Z$, and $Y = 0$ otherwise.
- By this definition, the event $Y = 1$ has probability $g^{-1}(\vec{X}\boldsymbol{\beta})$ — exactly the logistic regression population model.

Interpretations: Latent variables

The latent variable interpretation is particularly popular in econometrics, where it is a first example of a *discrete choice modeling*.

For example, in modeling customer choice over whether or not to buy a product, suppose:

- Each customer has a feature vector $\vec{X}$.
- This customer's *reservation* utility level is Z: The customer purchases the item if $\vec{X}\boldsymbol{\beta} > Z$, and does not purchase otherwise.
- The probability the customer purchases the item is exactly the logistic regression model.

This is a very basic example of a *random utility model* for customer choice.

Fitting and interpreting logistic regression models

Finding logistic regression coefficients

So far we have only discussed the population model for logistic regression. But so far we haven't designed an actual *method* for finding coefficients.

Logistic regression uses the population model we've discussed to suggest a way to find the coefficients $\hat{\boldsymbol{\beta}}$ of a fitted model. The basic outline is:

- *Assume* that the population model follows the logistic regression.
- *Find* coefficients (parameters) $\hat{\boldsymbol{\beta}}$ that *maximize the chance of seeing the data* $\mathbf{Y}$ *and* $\mathbf{X}$, given the logistic regression population model.

This is an example of the *maximum likelihood* approach to fitting a model. (We will discuss it more when we discuss inference.)

For now we investigate what the results of this fitting procedure yield for real data.

Example: The CORIS dataset

Recall: 462 South African males evaluated for heart disease.

Outcome variable: Coronary heart disease (`chd`).

Covariates:

- Systolic blood pressure (`sbp`)
- Cumulative tobacco use (`tobacco`)
- LDL cholesterol (`ldl`)
- Adiposity (`adiposity`)
- Family history of heart disease (`famhist`)
- Type A behavior (`typea`)
- Obesity (`obesity`)
- Current alcohol consumption (`alcohol`)
- Age (`age`)

Logistic regression

To run logistic regression in R, we use the `glm` function, as follows:

```
> fm = glm(formula = chd ~ .,
           family = "binomial",
           data = coris)
> display(fm)
            coef.est coef.se
(Intercept) -6.15     1.31
...
famhist      0.93     0.23
...
obesity     -0.06     0.04
...
```

Interpreting the output

Recall that if a coefficient is *positive*, it increases the probability that the outcome is 1 in the fitted model (since the logistic function is increasing).

So, for example, `obesity` has a *negative* coefficient. What does this mean? Do you believe the implication?

Interpreting the output

In linear regression, coefficients were directly interpretable; this is not as straightforward for logistic regression.

Some approaches to interpretation of $\hat{\beta}_j$ (holding all other covariates constant):

- $\hat{\beta}_j$ is the change in the log odds ratio for Y per unit change in X_j.
- $\exp(\hat{\beta}_j)$ is the change in the odds ratio for Y per unit change in X_j.

The "divide by 4" rule

We can also directly differentiate the fitted model with respect to X_j.

By differentating $g^{-1}(\vec{X}\hat{\boldsymbol{\beta}})$, we find that the change in the (fitted) probability $Y = 1$ per unit change in X_j is:

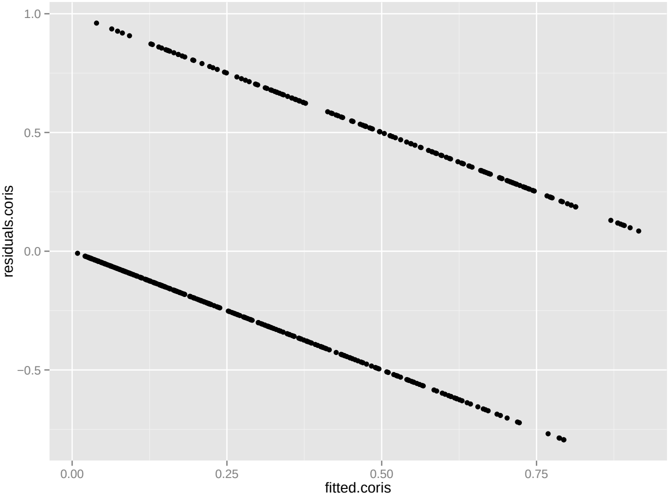
$$\left(\frac{\exp(\vec{X}\boldsymbol{\beta})}{[1 + \exp(\vec{X}\boldsymbol{\beta})]^2} \right) \hat{\beta}_j.$$

Note that the term in parentheses cannot be any larger than $1/4$.

Therefore, $|\hat{\beta}_j|/4$ is an upper bound on the magnitude of the change in the fitted probability that $Y = 1$, per unit change in X_j.

Residuals

Here is what happens when we plot residuals vs. fitted values:

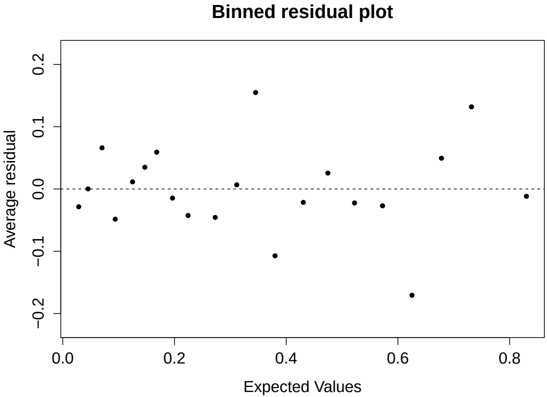


Why is this plot not informative?

Residuals

An alternative approach: *bin* the residuals based on fitted value, and then plot the average residual in each bin.

E.g., when we divide the data into ≈ 50 bins, here is what we obtain:



Logistic regression as a classifier

Classification

Logistic regression serves as a classifier in the following natural way:

- Given the estimated coefficients $\hat{\boldsymbol{\beta}}$, and a new covariate vector $\vec{X}$, compute $\vec{X}\hat{\boldsymbol{\beta}}$.
- If the resulting value is positive, return $Y = 1$ as the predicted value.
- If the resulting value is negative, return $Y = 0$ as the predicted value.

Linear classification

The idea is to directly use the model to choose the *most likely* value for Y, given the model:

When $\vec{X}\hat{\boldsymbol{\beta}} > 0$, the fitted model gives $\mathbb{P}(Y = 1|\vec{X}, \hat{\boldsymbol{\beta}}) > 1/2$, so we make the prediction that $Y = 1$.

Note that logistic regression is therefore an example of a *linear* classifier: the boundary between covariate vectors where we predict $Y = 1$ (resp., $Y = 0$) is linear.

Tuning logistic regression

As with other classifiers, we can *tune* logistic regression to trade off false positives and false negatives.

In particular, suppose we choose a threshold t, and predict $Y = 1$ whenever $\vec{X}\hat{\boldsymbol{\beta}} > t$.

- When $t = 0$, we recover the classification rule on the preceding slide. This is the rule that minimizes average 0-1 loss on the training data.
- What happens to our classifier when $t \to \infty$?
- What happens to our classifier when $t \to -\infty$

As usual, you can plot an ROC curve for the resulting family of classifiers as t varies.

Regularization

Regularized logistic regression [$*$]

As with linear regression, *regularized* logistic regression is often used in the presence of many features.

In practice, the most common regularization technique is to add the penalty $-\lambda \sum_j |\hat{\beta}_j|$ to the maximum likelihood problem; this is the equivalent of lasso for logistic regression.

As for linear regression, this penalty has the effect of selecting a subset of the parameters (for sufficient large values of the regularization penalty λ).

Do you believe the model?

Logistic regression highlights one of the inherent tensions in working with models:

- If we took the population model literally, it is meant to be a probabilistic description of the process that generated our data.
- At the same time, it appears that logistic regression is an estimation procedure that largely exists for its technical convenience. (Do we really believe that the population model is logistic?)

At times like this it is useful to remember:

> *"All models are wrong, but some are more useful than others."*
>
> *– George E.P. Box*