Lecture 12: Logistic Regression

STAT GU4206/GR5206 *Statistical Computing & Introduction to Data Science*

Gabriel Young
Columbia University

December 5, 2019

Lecture Notes

Topics:

- Supervised vs. Unsupervised Learning
- Supervised Learning
- Simple Logistic Regression (Parametric Model)

Supervised & Unsupervised Learning

Supervised vs. Unsupervised Learning

Supervised Learning

- Have access to a set of p predictors $X_1, X_2, ..., X_p$ and a response Y both measured on the same n observations.
- The goal is to predict Y using $X_1, X_2, ..., X_p$ (usually by learning β parameters of a model).

Unsupervised Learning

- *Only* have access to a set of p predictors $X_1, X_2, ..., X_p$ measured on n observations.
- We are not interested in prediction, because we do not have an associated response variable Y.
- The goal is to discover interesting patterns about the measurements on the predictors $X_1, X_2, ..., X_p$.

Supervised vs. Unsupervised Learning

The questions fall into two categories: **supervised learning** and *unsupervised learning*.

Supervised learning:

- Predicting an output
- Understanding the relationship between an input and an output

Unsupervised learning:

- Summarizing the data
- Understanding underlying (hidden) factors

Note: Principle Component Analysis (PCA) is unsupervised learning.

Supervised Learning

Supervised Learning: Regression and Classification

Regression:

- Y has continuous values

$$
\begin{aligned}
X &= (X_1, X_2, \ldots, X_p) && \text{inputs} \\
Y & && \text{output} \\
Y &= f(X) + \epsilon && \text{relationship} \\
Y &= E[Y|X = x] + \epsilon
\end{aligned}
$$

Classification:

- Y has categorical values

$$
\begin{aligned}
X &= (X_1, X_2, \ldots, X_p) && \text{inputs} \\
Y & && \text{output} \\
p_k &= P(Y = y_k|X = x) && \text{relationship}
\end{aligned}
$$

Prediction and Inference

Why estimate f and p?

- Prediction
- Inference

Prediction:

- We have a new product with a set advertising budget (`TV`, `radio` and `newspaper`). What will its sales be?
- Alice has 16 years of education and 0 years of seniority. What will her income be?

Goal:

Accurately estimate output for new inputs.

Inference

Inference

We want to learn about relationships between inputs and outputs:

- How will increasing one input affect the output?
- Is a specific combination of inputs associated with an increase in the output?

Inference vs. Prediction

- Can you think of some inference questions?
- Prediction questions?
- What is the difference between the two?

Fitting f and p

How do I find $\hat{f}$ and $\hat{p}$ using $(x_1, y_1), \ldots, (x_n, y_n)$?

1. select a statistical model
2. select the model parameters using the data MLE, MOM, MCMC

What types of statistical models are there?

- **Parametric:** described by a finite number of parameters, say

$$\beta_1, \beta_2, \ldots, \beta_d$$

- **Non-parametric:** not described by a finite number of parameters

Parametric Models

Parametric Models

A **parametric model** is a statistical model described by a finite number of parameters. Examples include:

- a Gaussian distribution ($N(\mu, \sigma^2)$)
- a Bernoulli distribution (Bern(p))
- a *linear model*

$$Y = \beta_0 + \beta_1 X_d + \cdots + \beta_d X_d + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

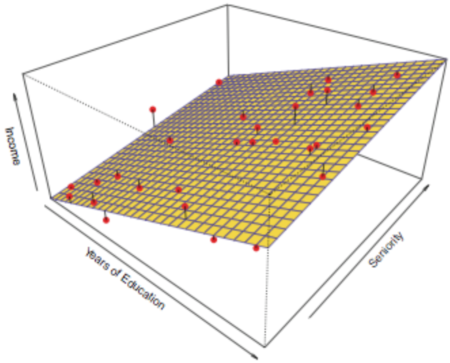
- a logistic model (this will be discussed today)

Parameters

- What are the parameters of the Gaussian?
- What are the parameters of a linear model?
- What are the parameters of a logistic model?

Parametric Regression Model

$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{years of education} + \beta_2 \times \texttt{seniority}$$



Is this model good for prediction? What can it tell us for inference?

Nonparametric Models

Nonparametric Models:

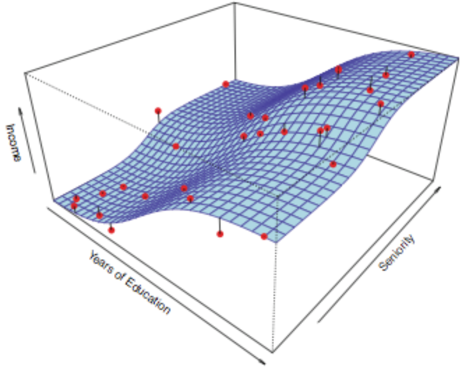
- Nonparametric models are not described by a finite number of parameters.
- So, what does that mean?
- Nonparametric models assume less about the population.
- In the model

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

we let the data decide what the function f looks like.

Nonparametric Model

$$\texttt{income} \approx f(\texttt{years of education}, \texttt{seniority})$$



Is this model good for prediction? What can it tell us for inference?

Problem Types

	Continuous	**Categorical**

Supervised	Regression
	`Parametric`
	Linear reg,...
	`Nonparametric`
	kNN,..
		Classification
		`Parametric`
		Logistic,...
		`Nonparametric`
		kNN,..

Unsupervised	Dimension Reduction PCA,..	Clustering k-means,..

- The above table gives a rough summary of basic ML methods. Each cell is not mutually exclusive.
- E.g., you could run a PCA on the the features $(X_1, X_2, \dots, X_p)$ of both linear regression and logistic regression.

Parametric Classification

Logistic Regression

Logistic Regression

Question:

How do we define a simple (one covariate x) regression model that allows for a categorical (binary) response variable Y?

- To answer this question, first recall the Bernoulli random variable.
- Any rv whose possible values are 0 and 1 is called a **Bernoulli random variable**.
- Also recall that the **expected value** (or true mean) of a Bernoulli random variable is its success probability. That is, if $Y \sim Bern(p)$, then

$$E[Y] = p$$

Answer:

Regress a sigmoidal function $p = f(x)$ on covariate x.

- A sigmoidal function has an s shape and is bounded between 0 and 1 ($0 < f(x) < 1$).

The Simple Logistic Regression Model

not identical

Let $Y_1, Y_2, \ldots, Y_n$ be independently distributed Bernoulli random variables with respective success probabilities $p_1, p_2, \ldots, p_n$ (where the x_i's are fixed). Then the **logistic regression model** is:

$$E[Y_i] = p_i = F_L(\beta_0 + \beta_1 x_i) = \frac{e^{(\beta_0 + \beta_1 x_i)}}{1 + e^{(\beta_0 + \beta_1 x_i)}}, \quad i = 1, 2, \ldots, n.$$

The Estimated Simple Logistic Model

$$\hat{p}_i = \frac{e^{(\hat{\beta}_0 + \hat{\beta}_1 x_i)}}{1 + e^{(\hat{\beta}_0 + \hat{\beta}_1 x_i)}}, \quad i = 1, 2, \ldots, n.$$

- The quantities $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimated intercept and slope.
- Maximum likelihood estimation is used to estimate the logistic model parameters β_0 and β_1.

Maximum Likelihood

- Usually we think of parameters, θ, as fixed and consider the probability of different outcomes $f(y, \theta)$ with θ constant and x changing.
- **Likelihood** of a parameter value is given by $L(\theta)$: what probability does θ give the data?
  - For continuous variables, use the probability density.
  - Calculate $f(y, \theta)$ letting θ change with data constant.
  - *Not* the probability of θ.
- **Maximum likelihood** is the guess that the parameter is whatever makes the data most likely.
- Most likely parameter value is the **maximum likelihood estimate** or the **MLE**.

Coding the Likelihood Function

- With independent data points $x_1, x_2, \ldots, x_n$ the likelihood is

$$L(\theta|y_1, \ldots, y_n) = \prod_{i=1}^{n} f(y_i, \theta).$$

- Multiplying lots of small numbering is bad, so we usually take the log:

$$\ell(\theta|y_1, \ldots, y_n) = \sum_{i=1}^{n} \log f(y_i, \theta).$$

- Note the maximizer is the same for both (though the maximum value will be different).

Maximum Likelihood Logistic

Recall the Bernoulli pmf

$$f(y|p) = P(Y = y) = p^y(1-p)^{1-y}$$

Joint pmf

$$f(y_1, y_2, \ldots, y_n | p_1, p_2, \ldots, p_n) = \prod_{i=1}^{n} p_i^{y_i}(1-p_i)^{1-y_i}$$

The objective function is:

$$\ell(\beta_0, \beta_1 | y_1, \ldots, y_n) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} \log(1 + \exp(\beta_0 + \beta_1 x_i)).$$

- Note: $E[Y_i] = p_i = p_i(x_i)$, x_i is fixed.

Log-odds and the Logit Link

$p_i = \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}$ $L(p) = \prod p_i^{y_i}(1-p_i)^{1-y_i}$

$l(p) = \sum[y_i \log p_i + (1-y_i)\log(1-p_i)] = \sum[y_i \log\frac{p_i}{1-p_i} + \log(1-p_i)]$

$= \sum y_i(\beta_0+\beta_1 x_i) - \sum \log(1+e^{\beta_0+\beta_1 x_i})$

Odds

$$\frac{p_i}{1-p_i} = e^{\beta_0+\beta_1 x_i}, \quad i = 1, 2, \ldots, n.$$

- The equation above relates the *odds* of event $\{Y = 1\}$ occurring to a deterministic *exponential function*.

Logit-Link and Log-Odds

$$F_L^{-1}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_i, \quad i = 1, 2, \ldots, n.$$

- The link function "links" the mean ($E[Y] = p$) to a linear function.

Logistic logit-link $g(u) = \log(\frac{u}{1-u})$

Linear identity-link $g(u) = u$

Example

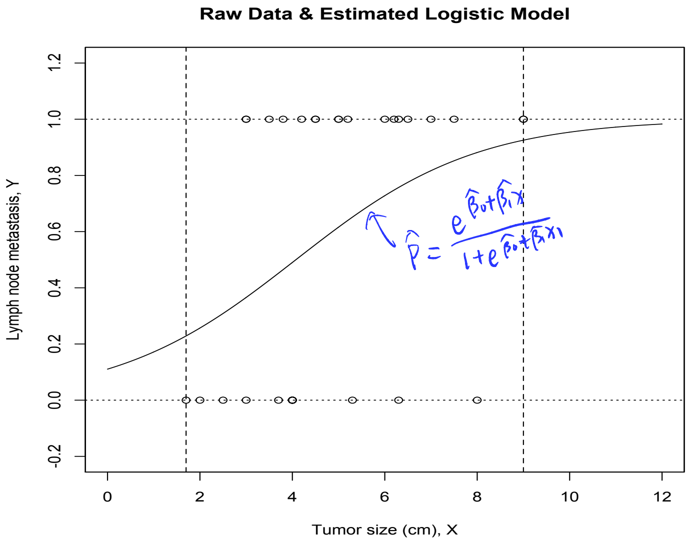
Esophageal cancer is a serious and very aggressive disease. Scientists conducted a study of 31 patients with esophageal cancer in which they studied the relationship between the size of the tumor that a patient had and whether or not the cancer had spread (metastasized) to the lymph nodes of the patient. In this study the response variable is dichotomous: $Y = 1$ if the cancer had spread to the lymph nodes and $Y = 0$ if not. The predictor variable is the size (recorded as the maximum dimension, in cm) of the tumor found in the esophagus.

```
> cancer <- read.table("logistic.txt")
```

Data Table

Patient number	Tumor Size (cm), X	Lymph node metastasis, Y
1	6.5	1
2	6.3	0
3	3.8	1
4	7.5	1
5	4.5	1
6	3.5	1
7	4.0	0
8	3.7	0
9	6.3	1
10	4.2	1
⋮	⋮	⋮
30	3	1
31	1.7	0

Plots


Raw Data & Estimated Logistic Model
Lymph node metastasis, Y
Tumor size (cm), X
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
2
4
6
8
10
12
$\hat{p} = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 x}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 x_1}}$

Estimating the Logistic Model - Gradient Descent

Process

- Define log-likelihood by:

$$\ell(\beta_0, \beta_1) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \sum_{i=1}^{n} \log(1 + \exp(\beta_0 + \beta_1 x_i)).$$

- Define negative-log-likelihood in R.

$$-\ell(\beta_0, \beta_1)$$

- Use `dbinom()` to set this up. Look up the argument `prob`. No need to define the neg-log-likelihood using the long expression from above.
- Use Newton's Method (Gradient Descent).
- Also try `nlm()`. glm( )
- Use literately reweighted least squares (**most common method**)!

Maximum Likelihood Logistic

Set up the neg-log-likelihood in R

```
> logistic.NLL <- function(beta,data=cancer) {
+
+   beta_0 <- beta[1]
+   beta_1 <- beta[2]
+   y <- data$y
+   x <- data$x
+   linear.component <- beta_0 + beta_1*x
+   p.i <- exp(linear.component)/(1+exp(linear.component))
+   return(-sum(dbinom(y,size=1,prob=p.i,log=TRUE)))
+ }
> logistic.NLL(beta=c(-1,.5),data=cancer)
[1] 21.72804
```

vector vector

nlm()

nlm(fn, θ_0, ...) → additional arg for fn

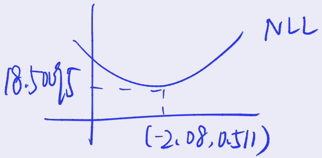
```
> nlm(logistic.NLL,p=c(-1,.5),data=cancer)

$minimum
[1] 18.50095

$estimate
[1] -2.0857732  0.5116513

$gradient
[1] 3.936344e-06 1.677591e-05
```

$\approx (0,0)$




```
$code
[1] 1

$iterations
[1] 15
```

Interpretation of the slope parameter β_1

Consider a 1 unit increase in the covariate:

- The odds of event $\{Y = 1\}$ occurring when the covariate is fixed at x is

$$odds_1 = \frac{p_1}{1 - p_1} = e^{\beta_0 + \beta_1 (x)}$$

- The odds of event $\{Y = 1\}$ occurring when the covariate is fixed at $x + 1$ is

$$odds_2 = \frac{p_2}{1 - p_2} = e^{\beta_0 + \beta_1 (x+1)}$$

- Thus

$$\text{"odds ratio"} = \Theta = \frac{odds_2}{odds_1} = \frac{e^{\beta_0 + \beta_1 (x+1)}}{e^{\beta_0 + \beta_1 x}} = e^{\beta_1}$$

- Equivalently $odds_2 = e^{\beta_1} \cdot (odds_1)$

"The odds of event $\{Y = 1\}$ occurring are multiplied by $e^{\hat{\beta}_1}$ for every 1 unit increase in x."

glm function in R

logistic

```
> cancer <- read.table("logistic.txt")
> model <- glm(y~x,data=cancer,family=binomial(link="logit"))
> model

Call:  glm(formula = y ~ x, family = binomial(link = "logit"),

Coefficients:
(Intercept)            x
    -2.0858       0.5117

Degrees of Freedom: 30 Total (i.e. Null);  29 Residual
Null Deviance:          42.17
Residual Deviance: 37         AIC: 41
```

Estimation in R

Summary

```
> summary(model)

Call:
glm(formula = y ~ x, family = binomial(link = "logit"), data =

Deviance Residuals:
    Min        1Q    Median        3Q       Max
-2.0657   -1.1288   0.5657    0.9844    1.4185

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.0858     1.2256  -1.702   0.0888 .
x             0.5117     0.2561   1.998   0.0457 *
---
Signif. codes:
0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ â
```

The `predict()` function in R

The `predict` function always predicts the "linear" component

.

$$\hat{\beta}_0 + \hat{\beta}_1 x$$

R code

```
> x.test <- data.frame(x=7)
> linear.pred <- predict(model,newdata = x.test)
> linear.pred

       1
1.495793

> exp(linear.pred)/(1+exp(linear.pred))

        1
0.8169462
```

Iteratively Reweighted Least Squares

(IRLS)

Iteratively Reweighted Least Squares (IRLS): Set-up

- X is the design matrix:

$$X = \begin{pmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1p} \\ 1 & X_{21} & X_{22} & \cdots & X_{2p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{np} \end{pmatrix}$$

- y is the response vector $y = [y_1 \ y_2 \ \ldots y_n]^T$
- p is the probability vector $p = [p_1 \ p_2 \ \ldots p_n]^T$ with

$$p_i = \frac{e^{(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip})}}{1 + e^{(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip})}}, \qquad i = 1, 2, \ldots, n.$$

- W is the diagonal matrix of variances:

$$W = \text{diag}\{p_1(1-p_1), p_2(1-p_2), \ldots, p_n(1-p_n)\}$$

Iterative Reweighted Least Squares (IRLS): Newton's

- The gradient of the log-likelihood can be written as:

$$\nabla[\ell(\beta_0, \beta_1)] = X^T(y - p) = \text{“score function”}$$

- The Hessian of the log-likelihood can be written as:

$$H_\ell(\beta_0, \beta_1) = -X^T WX \quad \text{(related to Fisher information)}$$

- The update function of Newton's is:

$$\theta_t = \theta_{(t-1)} - [H_\ell(\beta_0, \beta_1)]^{-1} \nabla[\ell(\beta_0, \beta_1)]$$

- Newton's searches for the extremum (min or max). Thus the update function of Newton's method for optimizing the logistic regression log-likelihood (or negative log-likelihood) is:

$$\theta_t = \theta_{(t-1)} + [X^T W_{(t-1)} X]^{-1} X^T (y - p_{(t-1)})$$

Iterratively Reweighted Least Squares (IRLS)

Summary

- Assume logistic regression model
- X is the design matrix
- y is the response vector
- p is the probability vector $p = [p_1 \ p_2 \ \dots p_n]^T$ with

$$p_i = \frac{e^{(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip})}}{1 + e^{(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip})}}, \quad i = 1, 2, \dots, n.$$

- W is the diagonal matrix of variances:

$$W = \mathrm{diag}\{p_1(1 - p_1), p_2(1 - p_2), \dots, p_n(1 - p_n)\}$$

- The IRLS update function to optimize the logistic regression:

$$\theta_t = \theta_{(t-1)} + [X^T W_{(t-1)} X]^{-1} X^T (y - p_{(t-1)})$$

Iterative Reweighted Least Squares: Example

```
> n <- nrow(cancer)
> X <- cbind(rep(1,n),cancer$x)
> y <- cancer$y
> # Iterations
> R <- 10
> # Starting values
> theta <- matrix(0,nrow=2,ncol=R+1)
> beta0 <- log(mean(y)/(1-mean(y)))
> theta[,1] <- c(beta0,0)
> for (i in 1:R) {
+   theta.i <- theta[,i]
+   linear.term.i <- theta.i[1]*X[,1]+theta.i[2]*X[,2]
+   p.i <- exp(linear.term.i)/(1+exp(linear.term.i))
+   W.i <- diag(p.i*(1-p.i))
+   theta[,i+1] <- theta[,i] + solve(t(X)%*%W.i%*%X)%*%(t(X)%*
+ }
```

intercept

initial guess

$$\hat{p} = \frac{e^{\beta_0+0}}{1+e^{\beta_0+0}} = \bar{y}$$

Iteratively Reweighted Least Squares: Example

```
> # Iteratively Reweighted Least Squares
> theta
          [,1]       [,2]       [,3]       [,4]       [,5]
[1,] 0.3254224 -1.7286746 -2.0543086 -2.0855252 -2.0857858
[2,] 0.0000000  0.4197561  0.5035851  0.5115873  0.5116542
           [,6]       [,7]       [,8]       [,9]      [,10]
[1,] -2.0857859 -2.0857859 -2.0857859 -2.0857859 -2.0857859
[2,]  0.5116542  0.5116542  0.5116542  0.5116542  0.5116542
          [,11]
[1,] -2.0857859
[2,]  0.5116542

> # Base R code
> model$coefficients

(Intercept)           x
 -2.0857859   0.5116542
```

Iteratively Reweighted Least Squares (IRLS)

IRLS in linear regression

- Recall the multiple regression model:

$$Y = X\beta + \epsilon, \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$$

- The ordinary least squares estimator:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

- The non-constant variance multiple linear regression model

$$Y = X\beta + \epsilon, \qquad \epsilon_i \overset{ind}{\sim} N(0, \sigma_i^2)$$

- The **weighted** least squares estimator:

$$\hat{\beta}_w = (X^T \Omega X)^{-1} X^T \Omega Y, \quad \text{where} \quad \Omega = \text{diag}\{1/\sigma_i^2\}$$

Iteratively Reweighted Least Squares (IRLS)

Big Picture

- The **ordinary** and **weighted** least squares estimators look similar to the inverse of the Hessian times the gradient of objective function f. In this case, f is the negative log-likelihood.

$$(H_f(\beta))^{-1}\,\nabla f(\beta) \;\text{ compare }\; (X^TX)^{-1}X^TY$$

$$(H_f(\beta))^{-1}\,\nabla f(\beta) \;\text{ compare }\; (X^T\Omega X)^{-1}X^T\Omega Y$$

Some closing thoughts

- IRLS often simplifies the Newton's method algorithm.
- IRLS is one of the most widely used algorithms for estimating `general linear models`.
- IRLS is especially useful for exponential families.
- Our example is a case of Fisher scoring (I dropped the details).

Optional Reading

- Chapter 2 (Optimization and Solving Nonlinear Equations) in Computational Statistics.