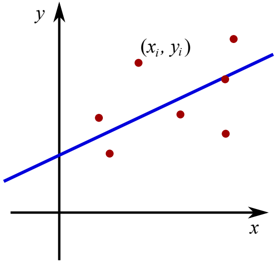
Logistic Regression

Foundations of Data Analysis

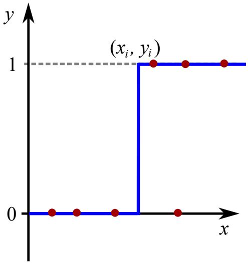
April 14, 2019

Classification as Regression

Regression problem:
Given x (independent variable),
predict y (dependent variable).

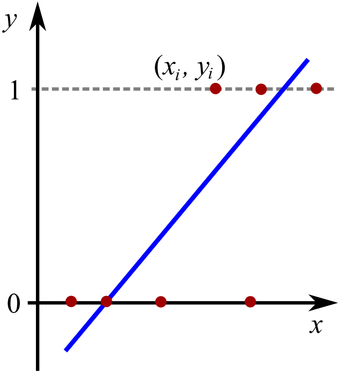


Classification problem:
Given x (features),
predict y (labels).



Classification as Regression

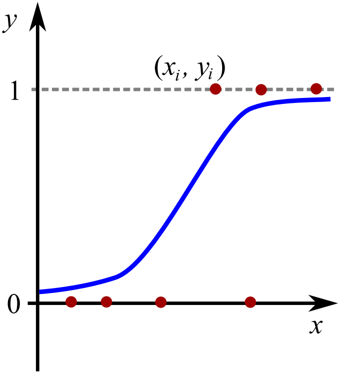
What if given x, we wanted to predict $p(y = 1 \mid x)$?



Linear fit to $p(y = 1 \mid x)$ goes outside $[0, 1]$!

Classification as Regression

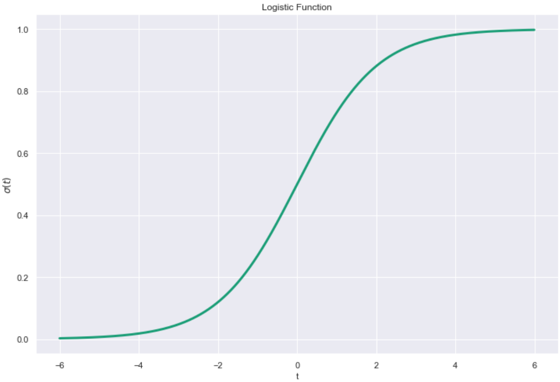
We want to use a nonlinear function with outputs in $[0, 1]$



This is *logistic regression*.

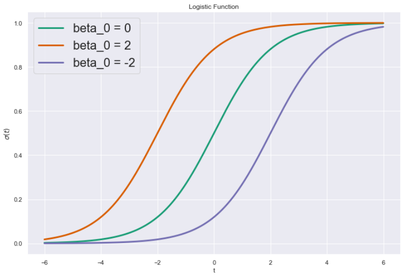
Logistic Function

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

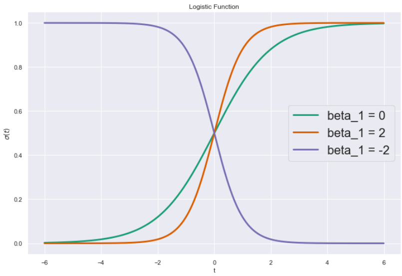


Linear Predictor Inside Logistic Function

$$p(y \mid x) = \sigma(\beta_0 + \beta_1 x) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x}}$$



β_0: “intercept”

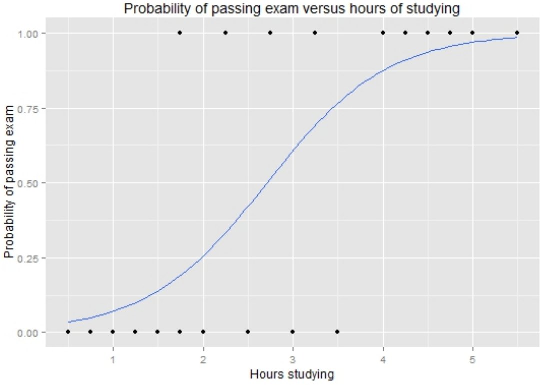


β_1: “slope”

Example (from Wikipedia)

Pass/fail of exam (y) vs. Hours spent studying (x)

Hours	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



Multivariate Predictor

If x is multivariate: $x = (x^{(1)}, x^{(2)}, \dots, x^{(d)})$,

$$\begin{aligned} p(y \mid x) &= \sigma(\beta_0 + \beta_1 x^{(1)} + \beta_2 x^{(2)} + \cdots + \beta_d x^{(d)}) \\ &= \frac{1}{1 + e^{-\beta_0 - \beta_1 x^{(1)} - \beta_2 x^{(2)} - \cdots - \beta_d x^{(d)}}} \end{aligned}$$

(Note: just multivariate linear regression inside σ)

Multivariate Predictor

Data matrix X with n data points (rows):

$$X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1d} \\ 1 & x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & & \vdots & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nd} \end{pmatrix}$$

Logistic regression evaluated for the ith data point (ith row vector):

$$p(y \mid X_{i\bullet}) = \sigma(X_{i\bullet}\beta)$$

(Note: $X_{i\bullet}\beta$ is the dot product btwn ith row and β)

How To Estimate Parameter β?

Maximize likelihood:

1. Compute derivative (gradient) of likelihood w.r.t. β
2. Solve for β that makes this derivative zero

Likelihood Function

Use Bernoulli likelihood:

$$L(\beta; X, y) = \prod_{i=1}^{n} \sigma(X_{i\bullet}\beta)^{y_i} \left(1 - \sigma(X_{i\bullet}\beta)\right)^{1-y_i}$$

Log-Likelihood Function

$$
\begin{aligned}
\ell(\beta; X, y) &= \ln L(\beta; X, y) \\
&= \sum_{i=1}^{n} (y_i - 1) X_{i\bullet}\beta - \ln(1 + e^{-X_{i\bullet}\beta})
\end{aligned}
$$

Gradient of Log-Likelihood Function

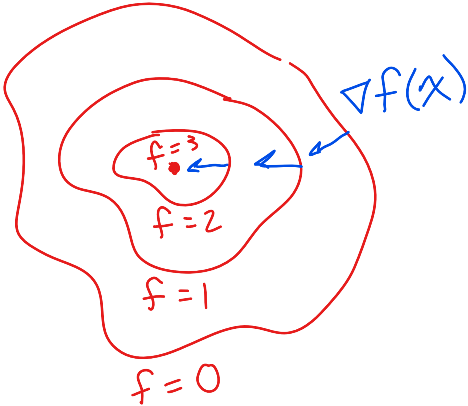
$$\nabla \ell(\beta; X, y) = \begin{bmatrix} \frac{\partial \ell}{\partial \beta_0} \\ \frac{\partial \ell}{\partial \beta_1} \\ \vdots \\ \frac{\partial \ell}{\partial \beta_d} \end{bmatrix}$$

$$\frac{\partial \ell}{\partial \beta_k} = \sum_{i=1}^{n} \left[(y_i - 1) + \frac{e^{-X_{i\bullet}\beta}}{1 + e^{-X_{i\bullet}\beta}} \right] X_{ik}$$

Problem! Can't solve for β that makes this zero!

Gradient Ascent

- Take a small step in the gradient direction
- Repeat until the gradient is zero



Algorithm for Logistic Regression

Set $\epsilon =$ small threshold
Set $\delta =$ step size along gradient
Initialize β

While $\|\nabla \ell\| > \epsilon$
    Update $\beta \leftarrow \beta + \delta \nabla \ell(\beta)$