Logistic Regression

Dan Sheldon

October 1, 2014

Logistic Regression

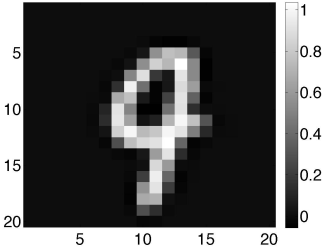
- Classification
- Model
- Cost function
- Gradient descent
- Linear classifiers and decision boundaries

Classification

- Input: $\mathbf{x} \in \mathbb{R}^n$
- Output: $y \in \{0, 1\}$

Example: Hand-Written Digits

Input: 20×20 grayscale image



$$\begin{bmatrix} x_1 & x_{21} & \dots & x_{381} \\ x_2 & x_{22} & \dots & x_{382} \\ & & \vdots & \\ x_{20} & x_{40} & \dots & x_{400} \end{bmatrix}$$

Unroll image into a feature vector $\mathbf{x} \in \mathbb{R}^{400}$

$$\mathbf{x} = (x_1, \dots, x_{400})^T$$

Output:

$$y = \begin{cases} 0 & \text{digit is "four"} \\ 1 & \text{digit is "nine"} \end{cases}$$

Example: Document Classification

Discuss on board.

The Learning Problem

- Input: $\mathbf{x} \in \mathbb{R}^n$
- Output: $y \in \{0, 1\}$
- Model (hypothesis class): ?
- Cost function: ?

Classification as regression?

Discuss on board

The Model

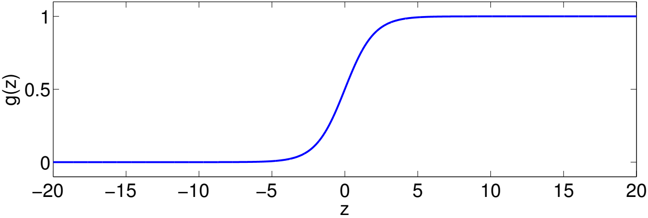
Exercise: fix the linear regression model

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = g(\boldsymbol{\theta}^T \mathbf{x}), \qquad g : \mathbb{R} \to [0, 1].$$

What should g look like?

Logistic Function

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



- This is called the *logistic* or *sigmoid* function

$$g(z) = \text{logistic}(z) = \text{sigmoid}(z)$$

The Model

Put it together

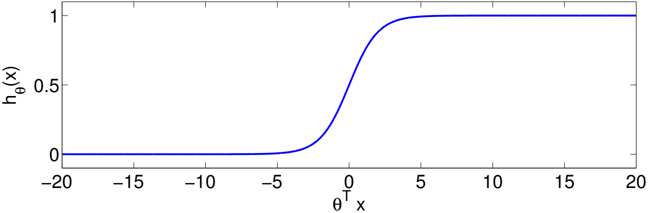
$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{logistic}(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

Nuance:

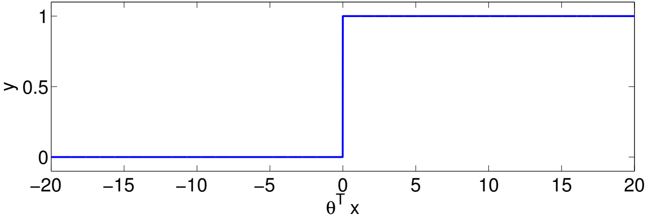
- Output is in $[0, 1]$, not $\{0, 1\}$.
- Interpret as probability

Hypothesis vs. Prediction Rule

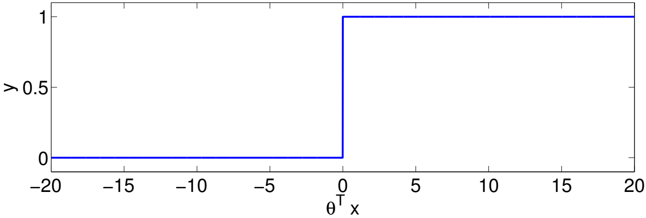
Hypothesis (for learning, or when probability is useful)



Prediction rule (when you need to commit!)



Prediction Rule



Rule

$$y = \begin{cases} 0 & \text{if } h_{\boldsymbol{\theta}}(\mathbf{x}) < 1/2 \\ 1 & \text{if } h_{\boldsymbol{\theta}}(\mathbf{x}) \geq 1/2 \end{cases}$$

Equivalent rule

$$y = \begin{cases} 0 & \text{if } \boldsymbol{\theta}^T \mathbf{x} < 0 \\ 1 & \text{if } \boldsymbol{\theta}^T \mathbf{x} \geq 0. \end{cases}$$

The Model—Big Picture

Illustrate on board: $\mathbf{x} \to z \to p \to y$

MATLAB visualization

Cost Function

Can we used squared error?

$$J(\boldsymbol{\theta}) = \sum_i (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})^2$$

This is sometimes done. But we want to do better.

Cost Function

Let's explore further. For squared error, we can write:

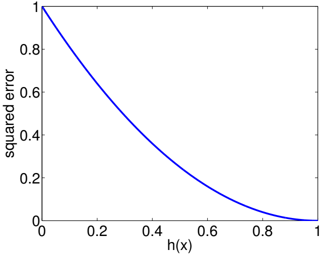
$$J(\boldsymbol{\theta}) = \sum_{i=1}^{m} \text{cost}(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}), y^{(i)})$$

$$\text{cost}(p, y) = (p - y)^2$$

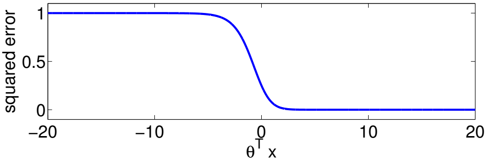
$\text{cost}(p, y)$ is cost of predicting $h_{\boldsymbol{\theta}}(\mathbf{x}) = p$ when the true value is y

Cost Function

Suppose $y = 1$. For squared error, $\mathrm{cost}(p, 1)$ looks like this

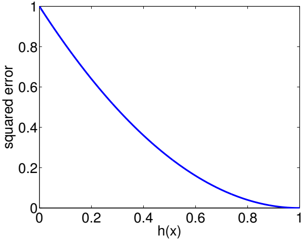


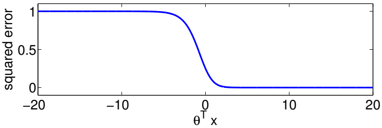
If we undo the logistic transform, it looks like this



Cost Function

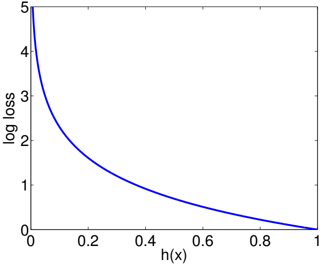
Exercise: fix these

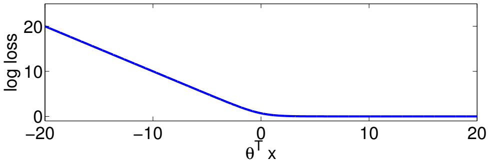




Log Loss $(y = 1)$

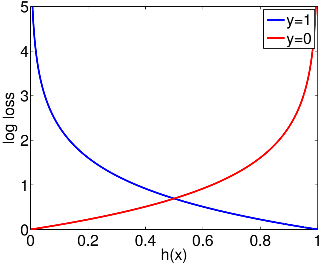
$$\text{cost}(p, 1) = -\log p$$

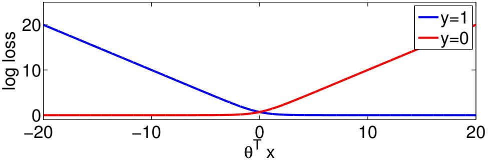




Log Loss

$$\text{cost}(p, y) = \begin{cases} -\log p & y = 1 \\ -\log(1-p) & y = 0 \end{cases}$$





Equivalent Expression for Log-Loss

$$\text{cost}(p, y) = \begin{cases} -\log p & y = 1 \\ -\log(1-p) & y = 0 \end{cases}$$

$$\text{cost}(p, y) = -y \log p - (1 - y) \log(1 - p)$$

$$\text{cost}(h_{\boldsymbol{\theta}}(\mathbf{x}), y) = -y \log h_{\boldsymbol{\theta}}(\mathbf{x}) - (1 - y) \log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}))$$

Review so far

- Input: $\mathbf{x} \in \mathbb{R}^n$
- Output: $y \in \{0, 1\}$
- Model (hypothesis class)

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{logistic}(\boldsymbol{\theta}^T \mathbf{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \mathbf{x}}}$$

- Cost function:

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{m} \Big(- y^{(i)} \log h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - (1 - y^{(i)}) \log(1 - h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})) \Big)$$

TODO: optimize $J(\boldsymbol{\theta})$

Gradient Descent for Logistic Regression

1. Initialize $\theta_0, \theta_1, \ldots, \theta_d$ arbitrarily
2. Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}), \qquad j = 0, \ldots, d.$$

Partial derivatives for logistic regression (exercise):

$$\frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta}) = 2 \sum_{i=1}^{m} (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

(Same as linear regression! But $h_{\boldsymbol{\theta}}(\mathbf{x})$ is different)

Decision Boundaries

Example from R&N (Fig. 18.15).

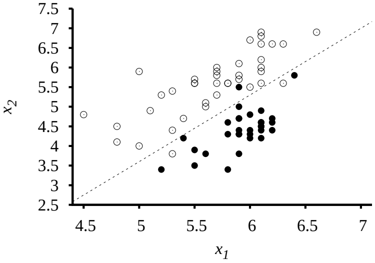
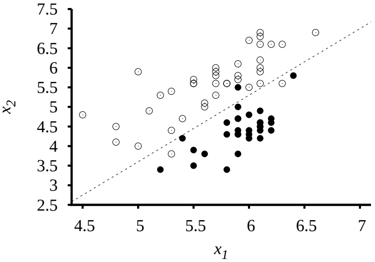


Figure : Earthquakes (white circles) vs. nuclear explosions (black circles) by body wave magnitude ($x1$) and surface wave magnitude ($x2$)

Decision Boundaries



E.g., suppose hypothesis is

$$h(x_1, x_2) = \text{logistic}(1.7x_1 - x_2 - 4.9)$$

Predict nuclear explosion if:

$$1.7x_1 - x_2 - 4.9 \geq 0$$
$$x_2 \leq 1.7x_1 - 4.9$$

Linear Classifiers

Predict

$$y = \begin{cases} 0 & \text{if } \boldsymbol{\theta}^T \mathbf{x} < 0, \\ 1 & \text{if } \boldsymbol{\theta}^T \mathbf{x} \geq 0. \end{cases}$$

Watch out! Hyperplane!

Many other learning algorithms use linear classification rules

- Perceptron
- Support vector machines (SVMs)
- Linear discriminants

Nonlinear Decision Boundaries by Feature Expansion

Example (Ng)

$$(x_1, x_2) \mapsto (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2),$$
$$\boldsymbol{\theta} = \begin{bmatrix} -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}^T$$

Exercise: what does decision boundary look like in (x_1, x_2) plane?

Note: Where Does Log Loss Come From?

$$\text{probability of } y \text{ given } p = \begin{cases} p & y = 1 \\ 1 - p & y = 0 \end{cases}$$

$$\mathrm{cost}(p, y) = -\log \mathrm{probability} = \begin{cases} -\log p & y = 1 \\ -\log(1 - p) & y = 0 \end{cases}$$

Find $\boldsymbol{\theta}$ to minimize cost $\longleftrightarrow$ Find $\boldsymbol{\theta}$ to maximize probability