Logistic Regression

- Hypothesis Representation

Want

$$0 \leq h_\theta(x) \leq 1$$

Sigmoid function:

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

Logistic function:

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

Interpretation:

$$h_\theta(x) = estimated\ probability\ that\ y = 1\ on\ input\ x$$

- Decision Boundary

$$h_\theta(x) = g(\theta^T x)$$

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

Suppose

$$predict\ \mathrm{y} = 1, if\ h_\theta(x) \geq 0.5$$

$$predict\ y = 0, if\ h_\theta(x) < 0.5$$

Decision boundary

$$\theta^T x = 0$$

- Non-linear decision boundaries

$$h_\theta(x) = g(-1 + x_1^2 + x_2^2)$$

$$Predict\ y = 1, if - 1 + x_1^2 + x_2^2 \geq 0$$

$$Predict\ y = 0, if - 1 + x_1^2 + x_2^2 < 0$$

Decision boundary

$$x_1^2 + x_2^2 = 1$$

- Cost function

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta\left(x^{(i)}\right), y^{(i)})$$

$$Cost(h_\theta(x), y) = \begin{cases} -\log\left(h_\theta(x)\right), if\ y = 1 \\ -\log\left(1 - h_\theta(x)\right), if\ y = 0 \end{cases}$$

$$\rightarrow J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right))$$

To filter parameters θ:

$$minJ(\theta)$$

To make a prediction given new x:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

- Gradient descent

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}(y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right))$$

Want $min_\theta J(\theta)$:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

$$= \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

- One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

On a new input x, to make a prediction, pick the class i that maximizes

$$max h_\theta^{(i)}(x)$$

- Regularization --- Linear Regression

$$J(\theta) = \frac{1}{2m} (\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{i=1}^{n} \theta_j^2)$$

What if λ is set to an extremely large value?

--- Algorithm works fine; setting λ to be very large cannot hurt it.

--- Algorithm fails to eliminate overfitting.

--- Algorithm results in underfitting.

--- Gradient descent fails to converge.

Gradient descent:

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha (\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} + \frac{\lambda}{m} \theta_j)$$

}

- Regularization --- Logistic Regression

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(-y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) - \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right) + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_i^2$$