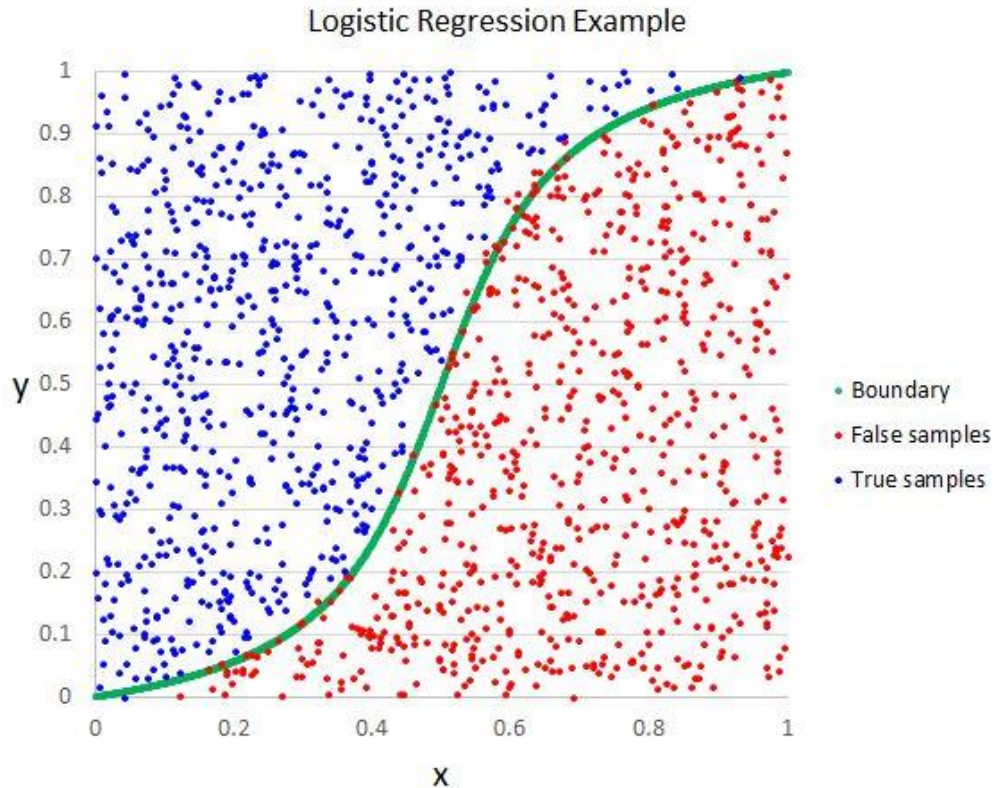# Logistic Regression (Logit)

Logistic Regression Example

Summary: Logistic regression is used in a classification problem. For example: determine whether a tumor is benign or malignant. The output of the classifier is a value 0 or 1 (for a binary classification problem). That means:

$$y \in \{0,1\}$$

0: "negative class" (i. e. benign tumor)

1: "positive class" (i. e. malignant tumor)

If $y \in \{0,1,2,\dots\}$, we call this a multi-class classification problem. In classification problem, the output should be either 0 or 1 (for binary classification problem). It should not output value $0 \leq h(x) \leq 1$.

1. ## Notation
   - m: number of training examples
   - n: number of features
   - x: input variable
   - y: output variable
   - $x^{(i)}$: input features of ith training example
   - $x_j^{(i)}$: value of feature j in ith training example
   - $y^{(i)}$: output of ith training example
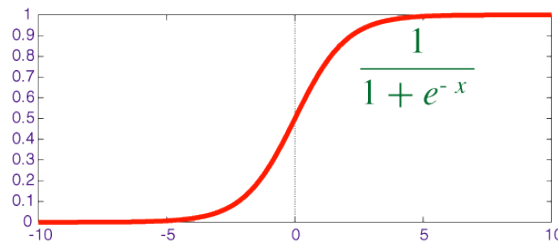   - $(x^{(i)}, y^{(i)})$: ith training example

Logistic Regression (Logit)

2. Hypothesis

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

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

- o The function $g(x)$ is called a sigmoid function.

- o Interpretation of the output of the hypothesis function:
  - ▪ $h_\theta(x)$ is the estimated probability that $y = 1$ on input x
  - ▪ Example:
    - • If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ tumorSize \end{bmatrix}$ and $h_\theta(x) = 0.7$
    - • It tells the patient that there is a 70% chance the tumor is malignant
  - ▪ $h_\theta(x) = P(y = 1|x; \theta)$ → The probability that $y = 1$, given x, parameterized by θ.

3. Decision Boundary

- o Consider this graph below:

this line is called the decision boundary
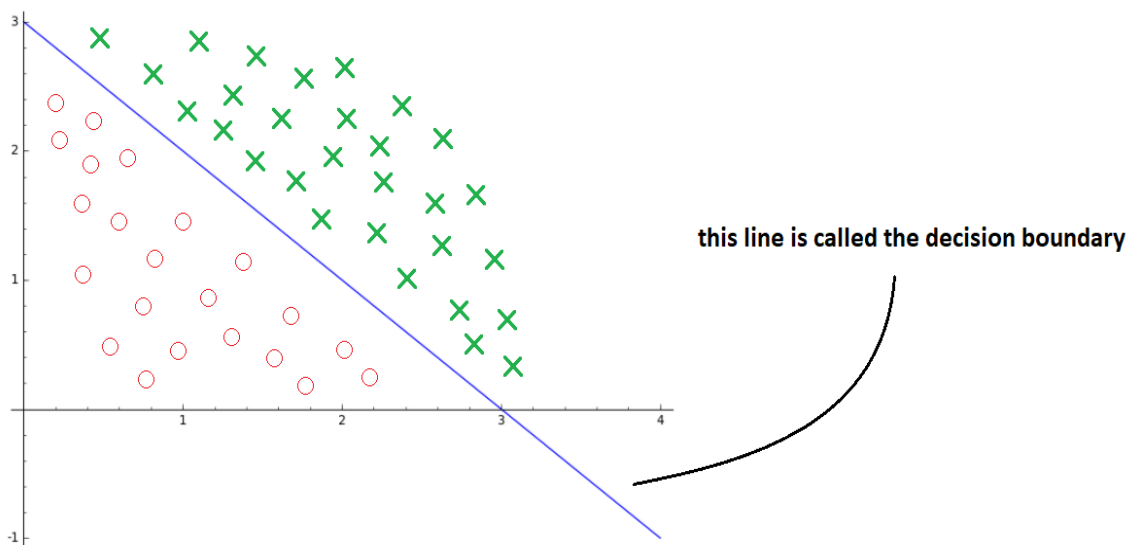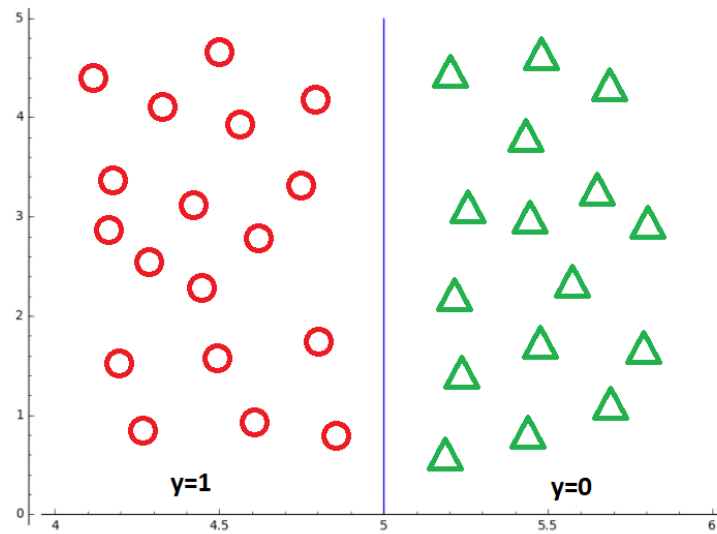
- o We have the decision boundary of the form $x_1 + x_2 = 3$ and the hypothesis $h_\theta = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. That means this graphs is predicting "$y = 1$" if $-3 + x_1 + x_2 \geq 0$
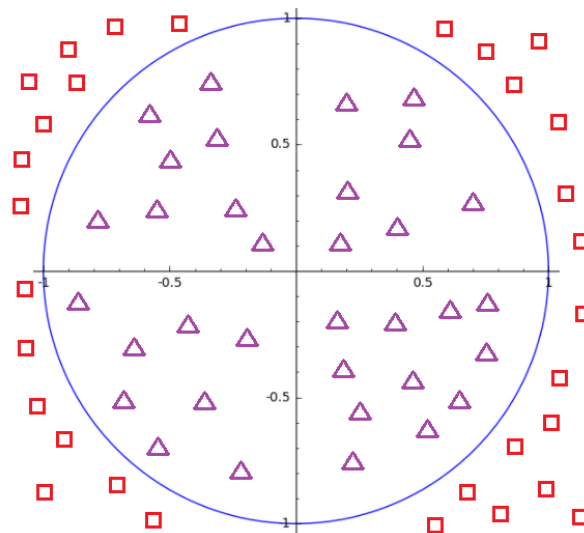
Logistic Regression (Logit)

○ Some other examples:

$$h_\theta(x) = g(5 - x_1)$$

$$\theta^T x = 5 - x_1$$

$$\rightarrow \begin{cases} x_1 \leq 5: \; y = 1 \\ x_1 > 5: \; y = 0 \end{cases}$$

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

$$\theta^T x = x_1^2 + x_2^2 - 1$$

$$\rightarrow \begin{cases} x_1^2 + x_2^2 \geq 1: \; y = 1 \\ x_1^2 + x_2^2 < 1: \; y = 0 \end{cases}$$

Logistic Regression (Logit)

4. ## Cost Function

   o Given a training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ with m examples and:

$$x \in \begin{bmatrix} x_0 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{R}^{n+1}, x_0 = 1$$

$$y \in \{0,1\}$$

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

   o Consider the cost function $J(\theta)$ in linear regression:

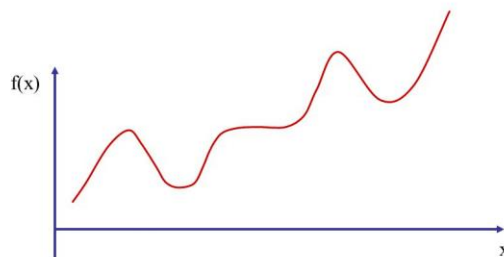$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^i) - y^i\right)^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} cost(h_\theta(x), y)$$

$$\rightarrow cost(h_\theta(x), y) = \frac{1}{2}(h_\theta(x) - y)^2$$

   o If we apply the cost function from simple linear regression to logistic regression, it would result in a non-convex function.
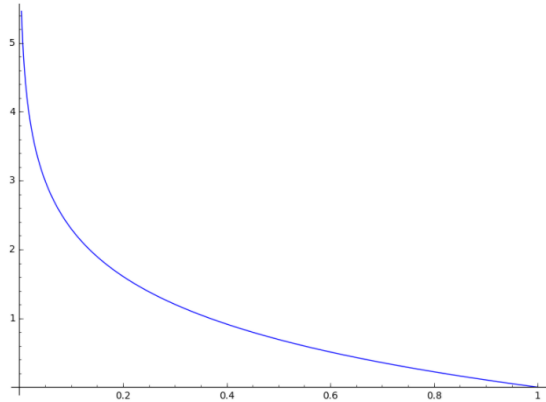
Example of Non-Convex Function

12

   o Therefore, we need to come up with a new cost function.
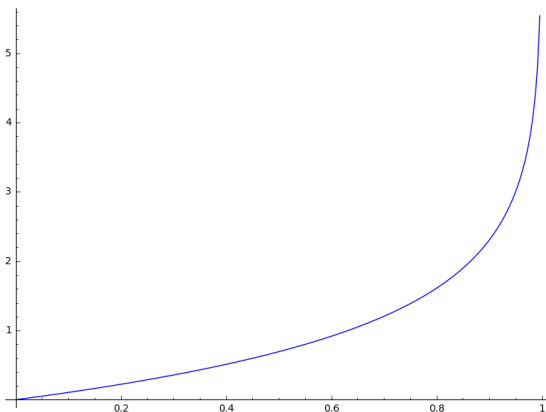
$$cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & if\ y = 1 \\ -\log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$

Logistic Regression (Logit)

When $y = 1$:
- $if\ h_\theta(x) = 1, cost = 0$
- As $h_\theta(x) \to 0, cost \to \infty$

When $y = 0$:
- $if\ h_\theta(x) = 0, cost = 0$
- As $h_\theta(x) \to 1, cost \to \infty$

o We can re-write the function above as following:

$$cost(h_\theta(x), y) = -y \log(h_\theta(x)) - (1 - y)\log(1 - h_\theta(x))$$

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

o To fit parameters θ, we need to minimize $J(\theta)$
o To make a prediction on a new input x:

$$output\ h_\theta(x) = P(y = 1|x; \theta) = \frac{1}{1 + e^{-\theta^T x}}$$

5. Gradient Descent Algorithm
o We have:

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

$$\to \frac{\partial J}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \qquad for\ j = 0 \dots n$$

o We then have the gradient descent algorithm as follow:

Logistic Regression (Logit)

Repeat {

$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} \qquad for\ j = 0 \dots n$$

}

6. <u>Vectorized Implementation</u>
   o A strategy to optimize gradient descent:

$$\theta := \theta - \frac{\alpha}{m}X^T(g(X\theta) - Y)$$

$$\theta \in \mathbb{R}^{n+1} \qquad Y \in \mathbb{R}^m$$

$$X \in M_{m\times(n+1)} \quad \rightarrow \quad X^T \in M_{(n+1)\times m}$$

7. <u>Optimization Algorithm</u>
   o Some available optimization algorithms:
     ▪ Gradient descent
     ▪ Conjugate gradient
     ▪ BFGS
     ▪ L-BFGS
   o Advantages of the last 3 algorithms:
     ▪ No need to manually pick learning rate α. They automatically pick the best learning rate for each iteration
     ▪ Often faster than gradient descent
   o Disadvantages:
     ▪ More complex

8. <u>Multiclass Classification</u>
   o Examples:
     ▪ Email foldering/tagging: work, friends, family, etc
     ▪ Weather: sunny, cloudy, rain, snow
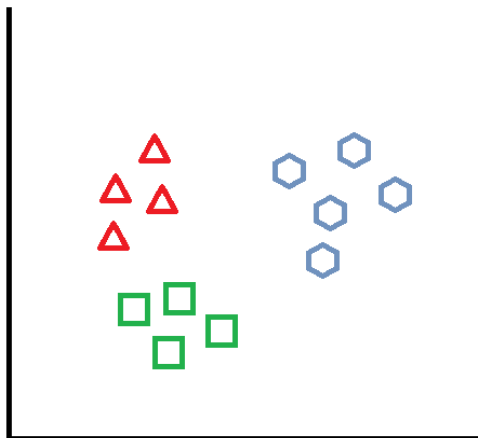
Logistic Regression (Logit)

- One strategy to solve a multiclass classification problem is <u>One-vs-all (one-vs-rest) classification</u>. In the example above, we can set:
  - Class 1: triangle
  - Class 2: hexagon
  - Class 3: square
- Then break the problem above into 3 smaller binary classification problems
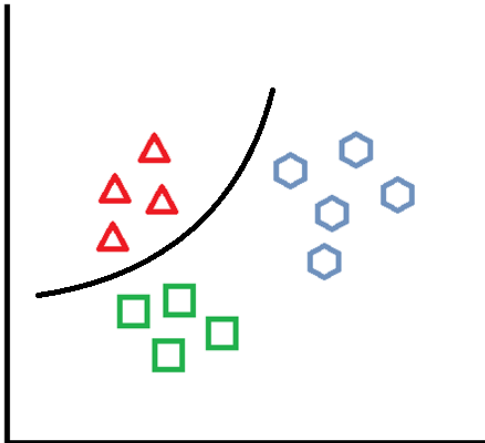
$$Classifier: h_\theta^{(1)}(x)$$
$$h_\theta^{(1)}(x) = P(y = 1 | x, \theta)$$
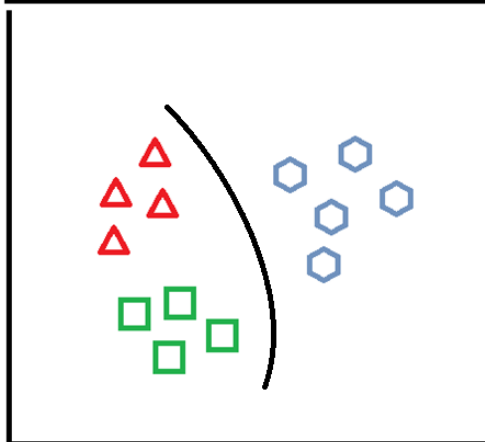
$$Classifier: h_\theta^{(2)}(x)$$
$$h_\theta^{(2)}(x) = P(y = 2 | x, \theta)$$
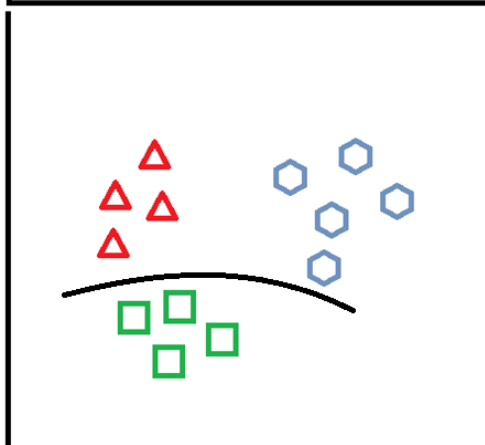
$$Classifier: h_\theta^{(3)}(x)$$
$$h_\theta^{(3)}(x) = P(y = 3 | x, \theta)$$

- In one-vs-al classification, we train logistic regression classifiers $h_\theta^{(i)}$ for each class i to predict the probability that $y = i$

Logistic Regression (Logit)

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

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

- o For a logistic regression problem with n classes, using one-vs-all classification, we'll need to break it into n binary classification problems.

9. Problem of Overfitting

- o Underfitting: poor performance on the training data, high bias, and poor performance on test data
- o Overfitting: fit the training data very well but has a poor performance on the test set, which fail to generalize new examples. Overfitting has a high variance.
- o There are 2 options to solve overfitting:
  - ▪ Reduce the number of features (manually or automatically)
  - ▪ Regularization
    - • Keep all the features, but reduce the magnitude/values of the parameters θ
    - • Work well when we have lots of features, each of which contributes a bit to the prediction of y

10.   Regularization – Cost Function

- o The ideas of regularization:
  - ▪ Penalize the parameters θ (make the values small)
  - ▪ → Get a "simpler" hypothesis
  - ▪ → Less prone to overfitting

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

$$\lambda: regularization\ parameter$$

- o If λ is set too large, we'll have $\theta_1, \theta_2, \theta_n \approx 0$
  - ▪ → $h_\theta(x) \approx 0$
  - ▪ → We'll encounter <u>underfitting</u> problem
- o Note:
  - ▪ We only penalize $\theta_1, \theta_2, \dots, \theta_n$ (Since θ_0 is set to be 1 by default)

11.   Regularized Linear Regression

- o Consider the regularized cost function $J(\theta)$ that we want to minimize

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

Logistic Regression (Logit)

- Gradient descent for linear regression is written as follows:

Repeat until convergence {

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

$$\theta_j := \theta_j - \alpha \left\{ \frac{1}{m} \sum_{i=1}^{m} \left[\left(h_\theta(x^{(i)}) - y^{(i)} \right) \left(x_j^{(i)} \right) \right] + \frac{\lambda}{m} \theta_j \right\} \qquad for \ j := 1,2,\dots,n$$

$$\frac{\partial}{\partial \theta_j} \left(regularized \ J(\theta) \right) \longrightarrow$$

}

- Consequently, we can re-write the gradient descent algorithm for regularized linear regression as follows:

Repeat until convergence {

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

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m} \right) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left[\left(h_\theta(x^{(i)}) - y^{(i)} \right) \left(x_j^{(i)} \right) \right] \qquad for \ j := 1,2,\dots,n$$

}

- Notes:
  - $1 - \alpha \frac{\lambda}{m} < 1$ → which means this term will shrink θ_j
- Consider normal equation method for linear regression, applying regularization to this method, we'll have:

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \right)^{-1} X^T Y$$

Is a $(n + 1) \times (n + 1)$ identity matrix with first value on diagonal line being 0

- Regularization will take care of the non-invertibility issue in normal equation method, making the matrix become an invertible.

12.   Regularized Logistic Regression
- Consider the cost function $J(\theta)$ for logistic regression in section 4 together with the regularized cost function $J(\theta)$ in section 10:

Logistic Regression (Logit)

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

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

o We can come up with a regularized cost function for logistic regression as follows:

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

o The gradient descent algorithm for regularized logistic regression can be written as follows:

Repeat until convergence {

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

$$\theta_j := \theta_j\left(1 - \alpha\frac{\lambda}{m}\right) - \alpha\frac{1}{m}\sum_{i=1}^{m}\left[(h_\theta(x^{(i)}) - y^{(i)})(x_j^{(i)})\right] \quad for\ j := 1,2,\dots,n$$

}

o Note: the algorithm looks identical to the gradient descent algorithm for regularized linear regression, however, the hypothesis h_θ for logistic regression is not the same.

13. **Mathematical Interpretation of The Partial Derivative of The Cost Function $J(\theta)$**

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

$$= -y\log(h_\theta(x)) + (y - 1)(\log(1 - h_\theta(x)))$$

$$= -y\log(h_\theta(x)) + y[\log(1 - h_\theta(x))] - \log(1 - h_\theta(x))$$

$$= y[\log(1 - h_\theta(x)) - \log(h_\theta(x))] - \log(1 - h_\theta(x))$$

$$= y\left[\log\left(\frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}\right) - \log\left(\frac{1}{1 + e^{-\theta^T x}}\right)\right] - \log\left(\frac{e^{-\theta^T x}}{1 + e^{-\theta^T x}}\right)$$

$$= y[\log(e^{-\theta^T x}) - \log(1 + e^{-\theta^T x}) + \log(1 + e^{-\theta^T x})] - \log(e^{-\theta^T x}) + \log(1 + e^{-\theta^T x})$$

Logistic Regression (Logit)

$$= y \log(e^{-\theta^T x}) - \log(e^{-\theta^T x}) + \log(1 + e^{-\theta^T x})$$

$$= \log(e^{-\theta^T x})(y - 1) + \log(1 + e^{-\theta^T x})$$

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

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\log\left(e^{-\theta^T x^{(i)}}\right)(y^{(i)} - 1) + \log\left(1 + e^{-\theta^T x^{(i)}}\right) \right]$$

$$\rightarrow \frac{\partial J}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} \left[\frac{\left(e^{-\theta^T x^{(i)}}\right)\left(-x_j^{(i)}\right)}{e^{-\theta^T x^{(i)}}}(y^{(i)} - 1) + \frac{\left(e^{-\theta^T x^{(i)}}\right)\left(-x_j^{(i)}\right)}{1 + e^{-\theta^T x^{(i)}}} \right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\left(-x_j^{(i)}\right)(y^{(i)} - 1) + \frac{\left(e^{-\theta^T x^{(i)}}\right)\left(-x_j^{(i)}\right)}{1 + e^{-\theta^T x^{(i)}}} \right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\left(x_j^{(i)}\right)\left(1 - y^{(i)} - \frac{e^{-\theta^T x^{(i)}}}{1 + e^{-\theta^T x^{(i)}}}\right) \right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\left(x_j^{(i)}\right)\left(\frac{1 + e^{-\theta^T x^{(i)}}}{1 + e^{-\theta^T x^{(i)}}} - \frac{e^{-\theta^T x^{(i)}}}{1 + e^{-\theta^T x^{(i)}}} - y^{(i)}\right) \right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\left(x_j^{(i)}\right)\left(\frac{1}{1 + e^{-\theta^T x^{(i)}}} - y^{(i)}\right) \right]$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\left(x_j^{(i)}\right)\left(h_\theta(x^{(i)}) - y^{(i)}\right) \right] \ for \ all \ j = 0 \ldots n$$