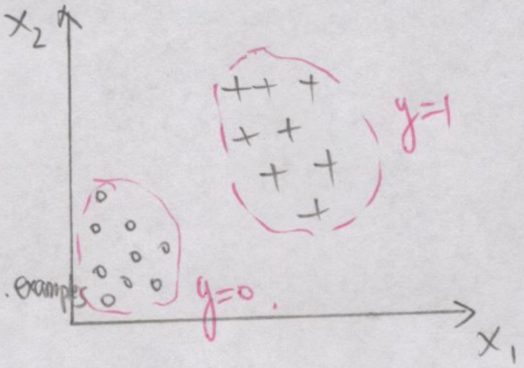
Logistic Regression.

- Logistic Regression: $0 \le h_\theta(x) \le 1$, using sigmoid function: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

  if $h_\theta(x) \ge 0.5$, $y=1$, otherwise, $y=0$.



e.g.:

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2).$$

Training set: $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ m. examples

And $x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ $x_0 = 1$, $y \in \{0,1\}$.

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

Logistic Regression Cost Function:

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

since $y=0$ or 1 always,

$$\text{Cost}(h_\theta(x), y) = -y \cdot \log(h_\theta(x)) - (1-y) \cdot \log(1-h_\theta(x)).$$

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

Gradient Descent:

Repeat {

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

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

Multiclass Logistic Regression:

i) One vs all

ii) One vs Rest.

Problem of Overfitting: if have too many features, $J(\theta) \approx 0$ in training set, but may fail to fit the test data.

"underfitting → high bias".

"Overfitting → High variance".

i). Reduce number of features.

ii). Regularization:

① Keep all the features, but reduce magnitude/values of parameters θ_j.

② Work well when we have a lot of features, each of which contributes a bit to predicting y.

Regularization: linear Regression

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

↓ Regularization parameter.

Notice: if λ is too large, algorithm results in underfitting, and may fail to fit even the training set).

Since all $\theta_1 \cdots \theta_n$ would be very close to 0, results in $h_\theta(x) = \theta_0$.

Now the Gradient descent becomes:

Repeat {

$$\theta_0 = \theta_0 - \alpha \cdot \frac{1}{m}\cdot\sum_{i=1}^{m} h_\theta(x^{(i)}) - y^{(i)})\, x_0^{(i)};$$

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

(for $j = 1, \cdots, n$)

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

}

$1 - \alpha\frac{\lambda}{m} < 1$, always. since $m > 1$, $\alpha > 0, \lambda > 0$.

Normal Equation with Regularization.

Suppose $m \le n$, $\qquad$ $\underline{\text{pinv}}$ to calculate $\boxed{\text{invert}}$

$$\theta = (X^T \cdot X)^{-1} \cdot X^T \cdot y.$$

if $\lambda > 0$,

$$\theta = \left(X^T \cdot X + \lambda \underbrace{\begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots 1 \end{bmatrix}}_{(n+1) \times (n+1)} \right)^{-1} \cdot X^T \cdot y.$$

Logistic Regression with Regularization.

Without Regularization: $J(\theta) = -\frac{1}{m} \cdot \sum_{i=1}^{m} [y^{(i)} \cdot \log(h_\theta(x^{(i)})) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))]$

With Regularization: $J(\theta) = J(\theta) + \frac{\lambda}{2m} \cdot \sum_{j=1}^{n} \theta_j^2$

So the Gradient Descent with Regularization is:

Repeat {

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

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

}.

gradient $\dfrac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m} \cdot \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \right) + \frac{\lambda}{m} \cdot \theta_j$