Machine Learning Formulae

Notations

m = Number of training examples.

n = Number of features.

$x^{(i)}$ = The features of the i_{th} training example, which is an $n+1$ vector.

$y^{(i)}$ = The Target value of the i_{th} training example, which is a number.

$(x^{(i)}, y^{(i)})$ = The i_{th} training example.

$x_j^{(i)}$ = The j_{th} feature of the i_{th} training example, which is a number.

α = Learning rate

Multivariate Linear Regression

Hypothesis

$h_x(\theta) = \sum_{i=0}^{n} \theta_i x_i$ x_0 is always equal to 1.

Cost Function

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

Gradient Desecnt

`Repeat{`

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

$(j = 0, 1, 2, \ldots, n)$, α stands for learning rate.

`}`

Calculate the $\frac{\partial}{\partial \theta_j} J(\theta)$, which means the partial derivative respect to θ_j:

`Repeat{`

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

`}`

Feature Scaling

Replace every x with:

$\frac{x-\mu}{s}$

μ stands for the average of x's, and s stands for the standard deviation of x's.

Normal Equation

Define the feature vector of the i_{th} training example:

$$x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ \\ x_1^{(i)} \\ \\ \cdots \\ \\ x_n^{(i)} \end{bmatrix}$$ $x^{(i)}$ is an $n+1$ vector.

Define ***design matrix*** X:

$$X = \begin{bmatrix} (x^{(1)})^T \\ \\ (x^{(2)})^T \\ \\ \cdots \\ \\ (x^{(m)})^T \end{bmatrix}$$ X is an $m \times (n+1)$ matrix.

Define the target value vector:

$$y = \begin{bmatrix} y^{(1)} \\ \\ y^{(2)} \\ \\ \cdots \\ \\ y^{(m)} \end{bmatrix}$$ y is an $m \times (n+1)$ vector.

Then, we can calculate θ:

$\theta = (X^T X)^{-1} X^T y$ θ Is an $n+1$ vector.

Logistic Regression

Hypothesis

Define:

$$\theta = \begin{bmatrix} \theta_0 \\ \\ \theta_1 \\ \\ \cdots \\ \\ \theta_n \end{bmatrix} x = \begin{bmatrix} x_0 \\ \\ x_1 \\ \\ \cdots \\ \\ x_n \end{bmatrix}$$

Hypothesis:

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

$g(z) = \frac{1}{1+e^{-z}}$ (sigmoid/logistic function)

which means:

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \in (0,1)$

Cost Function

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

$$Cost(h_\theta(x^{(i)}), y^{(i)}) = \begin{cases} -log(1-h_\theta(x)), & y=0 \\ -log(h_\theta(x)), & y=1 \end{cases} \iff Cost(h_\theta(x^{(i)}), y^{(i)}) = -ylog(h_\theta(x)) - (1-y)log(1-h_\theta(x))$$

Gradient Descent

```
Repeat{
```

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

```
}
```

Multiclass Classification

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

Regularized Linear Regression

Cost Function

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

Gradient Descent

```
Repeat{
```

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

$\theta_j := \theta_j - \alpha[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(j)} + \frac{\lambda}{m}\theta_j]$

$(j = 1, 2, 3, \dots, n)$

```
}
```

Normal Equation

$$\theta = (X^TX + \lambda\begin{bmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \dots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix})^{-1}X^Ty$$

Regularized Logistic Regression

Cost Function

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

Gradient Desecnt

`Repeat{`

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

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

$(j = 1, 2, 3, \ldots, n)$

`}`