scalar — $x_0^{(i)} = 1$ ~ intercept

$$y^{(i)} = \vec{\theta}^T \vec{x}^{(i)} + \epsilon^{(i)}$$

vector of features

Assumption: $\epsilon^{(i)} = y^{(i)} - \theta^T x^{(i)}$ are iid $\mathcal{N}(0, \sigma^2)$

$$\Longrightarrow p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\epsilon^{(i)2}}{2\sigma^2}}$$

$$p(y^{(i)} \mid x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

distribution of $y^{(i)}$'s!

$$L(\theta) = p(\vec{y} \mid \underbrace{X}_{\text{design matrix}}; \theta)$$

y

$$= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y^{(i)} - \theta^T x^{(i)})^2\right)$$

Chose θ to maximize $L(\theta)$!

Note:

$$\log L(\theta) = n \log \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{\sigma^2}\frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - \theta^T x^{(i)})^2$$

Max. likelihood $\Longleftrightarrow$ OLS

Logistic Regression

Suppose we have a Boolean variable y ~ pass/fail

which we want to model using $\vec{x}$ ~ explanatory/features.

Idea: Make a parameterized function $h_\theta(x)$ s.t.

$$P(y=1 \mid x; \theta) = h_\theta(x)$$

$$P(y=0 \mid x; \theta) = 1 - h_\theta(x)$$

$$\implies p(y \mid x; \theta) = h_\theta(x)^y \left(1 - h_\theta(x)\right)^{1-y}$$

design matrix of n observations $x^{(i)}$

$$L(\theta) = p(\vec{y} \mid X; \theta)$$

$$= \prod_{i=1}^{n} p(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{n} h_\theta(x)^{y^{(i)}} \left(1 - h_\theta(x)\right)^{1-y^{(i)}}$$

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

$\theta_0 + x_1^{(i)}\theta_1 + \ldots$

Take $h_\theta(x^{(i)}) = \sigma(\theta^T x^{(i)}) = \dfrac{1}{1 + e^{-\theta^T x^{(i)}}}$

Better yet; consider log-odds ~ $\log\left(\frac{p}{1-p}\right)$

assuming log-odds depend on a linear predictor η

$$\log\left(\frac{p}{1-p}\right) = \eta = \theta^T x^{(i)}$$

$p = (1-p)e^{\eta}$

$p = \frac{e^{\eta}}{1+e^{\eta}}$

$$\Longrightarrow p = \frac{e^{\eta}}{1+e^{\eta}} = \frac{1}{1+e^{-\eta}} = \sigma(\eta)$$

Define $h_\theta(x) = \sigma_\theta(x)$; $\sigma_\theta'(\eta) = \sigma_\theta(1-\sigma_\theta)$

$$\frac{\partial}{\partial\theta_j}\sigma(\theta^T x^{(i)}) = \sigma(1-\sigma)\, x_j^{(i)}$$

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

$$\Longrightarrow \frac{\partial}{\partial\theta_j}\ell(\theta) = \sum_{i=1}^{n}\left[y^{(i)}\frac{1}{h_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right]\frac{\partial h_\theta(x^{(i)})}{\partial\theta_j}$$

$$= \sum_{i=1}^{n}\left[y^{(i)}\frac{1}{\sigma_\theta(x^{(i)})} - (1-y^{(i)})\frac{1}{1-\sigma_\theta(x^{(i)})}\right]\sigma_\theta(x^{(i)})(1-\sigma_\theta(x^{(i)}))\, x_j^{(i)}$$

$$= \sum_{j=1}^{n}\left[y^{(i)}(1-\sigma(\theta^T x^{(i)})) - (1-y^{(i)})\sigma(\theta^T x^{(i)})\right] x_j^{(i)}$$

$y^{(i)} - y^{(i)}\sigma - \sigma + y\sigma$

$(\sigma(\theta^T x^{(1)}), \ldots, \sigma(\theta^T x^{(n)}))$

$$\Rightarrow \frac{\partial l(\theta)}{\partial \theta_j} = (\vec{y} - \sigma(\theta^T x)) x_j^{(i)}$$

max likelihood

$$\Rightarrow \theta_j := \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

~ looks like least square update step!

BUT! $h_\theta(x^{(i)})$ is non-linear!

Perceptron learning algorithm

$$h_\theta(x^{(i)}) = \begin{cases} 1, & \theta^T x \geq 0 \\ 0, & \theta^T x < 0 \end{cases}$$

- Somehow, this is a much different learning algorithm.
  - not like least squares or logistic

Decision boundary

$$p = \frac{1}{2} = \frac{1}{1 + e^{-\theta \cdot x}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x + \theta_2 y)}}$$

$$1 + e^{-\theta x} = \frac{1}{p} \qquad e^{-\theta x} = \frac{1}{p} - 1 = \frac{1-p}{p}$$

$$\theta x = \log\left(\frac{p}{1-p}\right)$$

for $p = 1/2$; $\theta x = 0$

$$\theta_0 + \theta_1 x + \theta_2 y = 0$$

$$y = -\frac{1}{\theta_2}(\theta_0 + \theta_1 x)$$