Logistic Regression

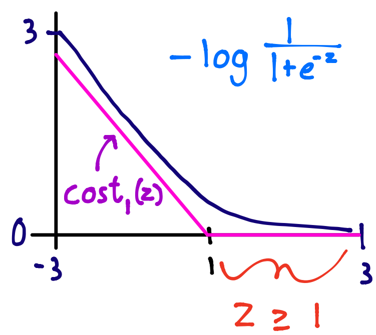
Sid

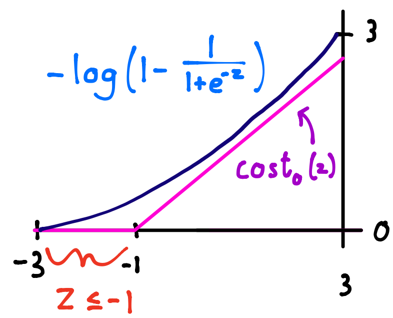
$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

$$y=1 : h_\theta(x) \approx 1,\ \theta^T x \gg 0$$
$$y=0 : h_\theta(x) \approx 0,\ \theta^T x \ll 0$$

$$\text{Cost}(x,y) : -\left(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x))\right)$$
$$= -y \log \frac{1}{1+e^{-\theta^T x}} - (1-y)\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)$$

if $y=1$ (want $\underbrace{\theta^T x}_{(z)} \overset{\geq 1}{\gg} 0$)   if $y=0$ (want $\theta^T x \overset{\leq -1}{\ll} 0$)





Logistic :

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \underbrace{\left(-\log h_\theta(x^{(i)})\right)}_{\text{cost}_1(z)} + (1-y^{(i)})\underbrace{\left(\left(-\log(1-h_\theta(x^{(i)}))\right)\right)}_{\text{cost}_0(z)}\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

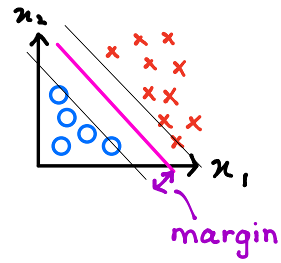
(convex)

SVM :

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$\text{hypothesis } (h_\theta(x)) : \begin{cases} 1 & \text{if } \theta^T x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

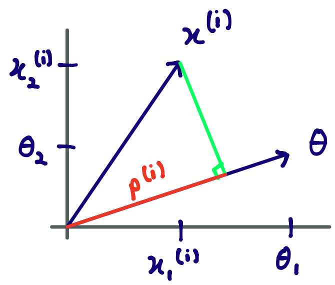
SVM: Large Margin Classifier



$(\theta_0 = 0, n = 2)$

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n} \theta_j^2 = \frac{1}{2}\left(\sqrt{\theta_1^2+\theta_2^2}\right)^2 = \frac{1}{2}||\theta||$$

$$p^{(i)}||\theta|| \begin{cases} \underline{\theta^T x^{(i)}} \geq 1, & y^{(i)} = 1 \\ \underline{\theta^T x^{(i)}} \leq -1, & y^{(i)} = 0 \end{cases}$$



$$\theta^T x^{(i)} = p^{(i)} \cdot ||\theta||$$
$$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

Kernels : used to make non-linear classifiers using SVM

Given x $\longrightarrow$ compute new features (f) depending on proximity to landmarks (l)

$$f^{(i)} = \underbrace{\overset{(k)}{\text{similarity}}}_{}(x, l^{(i)}) = \exp\left(\frac{-\|x - l^{(i)}\|^2}{2 \cdot \sigma^2}\right)$$

Kernel (Gaussian kernal)

$$= \exp\left(-\frac{\sum_{j=1}^{n}(x_j - l_j^{(i)})^2}{2\sigma^2}\right)$$

- $x \approx l^{(i)} \longrightarrow f_i = \exp\left(-\frac{\approx 0^2}{2\sigma^2}\right) \approx 1$
- x far from $l^{(i)} \longrightarrow f_1 = \exp\left(\frac{-\text{large } \#^2}{2\sigma^2}\right) \approx 0$

$$h_\theta(x) = \theta_1 f_1 + \theta_2 + f_2 + \cdots$$

Given $(x^{(1)}, y^{(1)}), \ldots (x^{(m)}, y^{(m)})$

$\downarrow$

Choose $l^{(1)} = x^{(1)}, \ldots l^{(m)} = x^{(m)}$

$$x^{(i)} \longrightarrow \begin{bmatrix} f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)}) \\ \vdots \\ f_m^{(i)} = \text{sin}(x^{(i)}, l^{(m)}) \end{bmatrix}$$

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} = 1 \\ \vdots \\ f_m^{(i)} \end{bmatrix}$$

$$\begin{cases} 1 : \theta^T f \geq 0 \\ 0 : \theta^T f < 0 \end{cases}$$

$= \theta^T\theta$

$\downarrow$

$\theta^T M \theta$

$$\min_\theta \; C\sum_{i=1}^{m} y^{(i)} \text{cost}_1(\underline{\theta^T f^{(i)}}) + (1-y^{(i)})\text{cost}_0(\underline{\theta^T f^{(i)}}) + \frac{1}{2}\underbrace{\sum_{j=1}^{n=m}\theta_j^2}_{}$$

$C \ (\approx \frac{1}{\lambda})$

- large C : Low bias, high variance
- small C : High bias, low variance

σ^2

- Large : Features f_i vary smoothly ($\uparrow$ Bias, $\downarrow$ Variance)
- Small : f_i vary less smoothly ($\downarrow$ Bias, $\uparrow$ Variance)

Using a SVM

- use SVM library (liblinear, libsvm, ...)

• Choose C

• Choose kernel (sim. function)

- No kernel ("linear") $\rightarrow$ standard linear classifier
  (n is large, m is small) or logistic regression
- Gaussian kernel (n is small, m is large)
  • choose σ^2

if too large, add features

function f = kernel (x_1, x_2)

$x^{(i)}$ $\quad l^{(j)} = x^{(j)}$

$$f = exp\left(-\frac{||x_1 - x_2||^2}{2\sigma^2}\right)$$

end

☆ feature scaling before using Gaussian kernel

Multi-class classification using SVM

$y \in \{1, 2, \dots K\}$

$\rightarrow$ Train k SVMs to distinguish $y = i$ $\rightarrow$ $\theta^{(1)}, \theta^{(2)}, \dots \theta^{(K)}$

$\rightarrow$ Pick class i w/ largest $(\theta^{(i)})^T x$