Logistic regression:

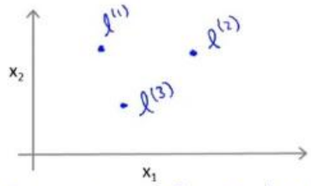
$$\min_{\theta} \underset{C}{\cancel{\frac{1}{m}}} \left[\sum_{i=1}^{m} y^{(i)} \left(-\log h_\theta(x^{(i)}) \right) + (1 - y^{(i)}) \left((-\log(1 - h_\theta(x^{(i)})) \right) \right] + \frac{\overset{1}{\cancel{\lambda}}}{2\cancel{m}} \sum_{j=1}^{n} \theta_j^2$$

Support vector machine:

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \underbrace{\left(-\log h_\theta(x^{(i)}) \right)}_{=} + (1 - y^{(i)}) \underbrace{\left((-\log(1 - h_\theta(x^{(i)})) \right)}_{=} \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} \overbrace{cost_1(\theta^T x^{(i)})} + (1 - y^{(i)}) \overbrace{cost_0(\theta^T x^{(i)})} \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

Kernel



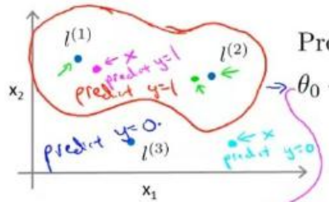
Given x, compute new feature depending on proximity to landmarks $l^{(1)}, l^{(2)}, l^{(3)}$

Given x: $f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$ $\|x - l^{(1)}\|^2$ $\|w\|$

$f_2 = \text{similarity}(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$

$f_3 = \text{similarity}(x, l^{(3)}) = \exp(\ldots)$

Kernel (Gaussian kernels) $k(x, l^{(i)})$



Predict "1" when

$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

x

$\theta_0 = -0.5, \quad \theta_1 = 1, \quad \theta_2 = 1, \quad \theta_3 = 0$

$f_1 \approx 1, \quad f_2 \approx 0, \quad f_3 \approx 0.$

$\rightarrow \theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$

$= -0.5 + 1 = 0.5 \geq 0$

$f_1, f_2, f_3 \approx 0$

$\rightarrow \theta_0 + \theta_1 f_1 + \ldots \approx -0.5 < 0$

Andrew Ng

SVM with Kernels

→ Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,

→ choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Given example x:

→ $f_1 = \text{similarity}(x, l^{(1)})$ $x^{(1)}$

→ $f_2 = \text{similarity}(x, l^{(2)})$

$\ldots$

$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$ $f_0 = 1$

For training example $(x^{(i)}, y^{(i)})$:

$x^{(i)} \rightarrow$ $f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)})$

$f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)})$

$\vdots$ ← $f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp(-\frac{0}{2\sigma^2}) = 1$ $x^{(i)}$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$x^{(i)} \in \mathbb{R}^{n+1}$ (or $\mathbb{R}^n$)

$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix}$

$f_0^{(i)} = 1$

Andrew Ng

SVM with Kernels

Hypothesis: Given x, compute features $f \in \mathbb{R}^{m+1}$ $\theta \in \mathbb{R}^{m+1}$

→ Predict "y=1" if $\theta^T f \geq 0$ $\theta_0 f_0 + \theta_1 f_1 + \cdots + \theta_m f_m$

Training: $n = m$

$$\rightarrow \min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{\cancel{n}\, = m} \theta_j^2$$

$\cancel{\theta^T x^{(i)}}$ $\theta^T f^{(i)}$ $\rightarrow \theta_0$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \left(-log\left(\frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right) + (1-y^{(i)}) \left(-log\left(1 - \frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right) + \frac{1}{2} \sum_{j}^{n} \theta_j^2$$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \left(-log\left(\frac{1}{1+e^{\left(\theta^T exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)\right)}}\right)\right) + (1-y^{(i)}) \left(-log\left(1 - \frac{1}{1+e^{\left(\theta^T exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)\right)}}\right)\right) + \frac{1}{2} \sum_{j}^{n} \theta_j^2$$

$$x^{(j)} = \begin{bmatrix} x_1^{(j)} \\ x_2^{(j)} \\ \vdots \\ x_n^{(j)} \end{bmatrix} \quad l^{(i)} = \begin{bmatrix} l_1^{(i)} \\ l_2^{(i)} \\ \vdots \\ l_n^{(i)} \end{bmatrix} \quad x^{(j)} - l^{(i)} = \begin{bmatrix} x_1^{(j)} - l_1^{(i)} \\ x_2^{(j)} - l_2^{(i)} \\ \vdots \\ x_n^{(j)} - l_n^{(i)} \end{bmatrix}$$

Gaussian Kernel $= exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)$

$$C = \frac{1}{\lambda}$$

$$l^{(i)} = x^{(i)}$$

$$similarity(x, l^{(i)}) = kernel(x, l^{(i)}) = k(x, l^{(i)}) = f_i = exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right)$$

$$k(x, l^{(i)}) \quad \forall i \in m, m = |x| \Rightarrow \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

HINT: x is the Matrix containing all m training examples of x
$l^{(i)}$ is the i'th landmark, one landmark is actually one specific training example of x.

$l^{(i)} \in x$

$$\|x - l^{(i)}\| = \begin{bmatrix} \sqrt{(x_1 - l^{(i)})_1^2 + ... + (x_1 - l^{(i)})_n^2} \\ \sqrt{(x_2 - l^{(i)})_1^2 + ... + (x_2 - l^{(i)})_n^2} \\ \vdots \\ \sqrt{(x_m - l^{(i)})_1^2 + ... + (x_m - l^{(i)})_n^2} \end{bmatrix} = \begin{bmatrix} \sqrt{\sum_{j=1}^n (x_1 - l^{(i)})_j^2} \\ \sqrt{\sum_{j=1}^n (x_2 - l^{(i)})_j^2} \\ \vdots \\ \sqrt{\sum_{j=1}^n (x_m - l^{(i)})_j^2} \end{bmatrix}$$

HINT: $\|x - l^{(i)}\|$ simply measures the distance of $l^{(i)}$ to all other m training examples of x
NOTE: $\|x - l^{(i)}\|^2$ will result in elements without square root!

$$\|x - l^{(i)}\|^2 = \begin{bmatrix} \sqrt{(x_1 - l^{(i)})_1^2 + ... + (x_1 - l^{(i)})_n^2}^2 \\ \sqrt{(x_2 - l^{(i)})_1^2 + ... + (x_2 - l^{(i)})_n^2}^2 \\ \vdots \\ \sqrt{(x_m - l^{(i)})_1^2 + ... + (x_m - l^{(i)})_n^2}^2 \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n (x_1 - l^{(i)})_j^2 \\ \sum_{j=1}^n (x_2 - l^{(i)})_j^2 \\ \vdots \\ \sum_{j=1}^n (x_m - l^{(i)})_j^2 \end{bmatrix}$$

$$cost_1(\theta^T f^{(i)}) = \left(-log\left(h_\theta(f^{(i)})\right)\right) = \left(-log\left(\frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right)$$

$$cost_0(\theta^T f^{(i)}) = \left(-log\left(1 - h_\theta(f^{(i)})\right)\right) = \left(-log\left(1 - \frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right)$$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_j^n \theta_j^2 =$$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \left(-log\left(h_\theta(f^{(i)})\right)\right) + (1 - y^{(i)}) \left(-log\left(1 - h_\theta(f^{(i)})\right)\right) + \frac{1}{2} \sum_j^n \theta_j^2 =$$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \left(-log\left(\frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right) + (1 - y^{(i)}) \left(-log\left(1 - \frac{1}{1+e^{\left(\theta^T f^{(i)}\right)}}\right)\right) + \frac{1}{2} \sum_j^n \theta_j^2 =$$

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} \left(-log\left(\frac{1}{1+e^{\left(\theta^T exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)\right)}}\right)\right) + (1 - y^{(i)}) \left(-log\left(1 - \frac{1}{1+e^{\left(\theta^T exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)\right)}}\right)\right) + \frac{1}{2} \sum_j^n \theta_j^2$$

Example, calculating the gaussian kernel for an entire Matrix X.
Since Landmarks are actually also X, I simply show the Landmarks Matrix in blue X.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix}$$

Remember $\|x - l^{(i)}\|^2$
this is actually the same as $\|x^{(i)} - x^{(j)}\|^2$
NOTE:
$x^{(i)}$ denotes the $i'th$ ROW from X.
$x^{(j)}$ denotes the $j'th$ ROW from X.
If
$i = 1 \Rightarrow x^{(1)}$
and
$j = 1 \Rightarrow x^{(1)}$
then the affected rows would be as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix}$$

Step 1: Calculate the distance d_{11} of $x_1$and x_1
NOTE: Each row has $\boldsymbol{n}$ entries, to be more precise, in the Machine Learning Context these are $\boldsymbol{n}$ features (dimensions).

$$\sqrt{\sum_{k=1}^{n}(x_{1k} - x_{1k})^2}^{\,2} = \sum_{k=1}^{n}(x_{1k} - x_{1k})^2 = d_{11}$$

Step 2: Continue calculating the distance d_{12} of $x^{(1)}$and $x^{(2)}$ usw.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix}$$

$$\sqrt{\sum_{k=1}^{n}(x_{1k} - x_{2k})^2}^{\,2} = \sum_{k=1}^{n}(x_{1k} - x_{2k})^2 = d_{12}$$

Step 3: Continue calculating the distance d_{1m} of $x^{(1)}$and $x^{(m)}$ usw.

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix}$$

$$\sqrt{\sum_{k=1}^{n}(x_{1k} - x_{mk})^2}^{\,2} = \sum_{k=1}^{n}(x_{1k} - x_{mk})^2 = d_{1m}$$

Step 4: Continue calculating the distance d_{21} of $x^{(2)}$and $x^{(1)}$ and so on...

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & x_{12} & \dots \\ \vdots & \ddots & \\ x_{m1} & & x_{mn} \end{bmatrix}$$

Continue until all distances $x^{(i)} \in X$ have been measured to all elements $x^{(j)} \in X$.

Step 5: You will end up with an $m \times m$ Distance D matrix. Each element in the matrix denotes the distance between an element from X and X .
The index of each element d_{ij} shows the origin of the element.
The red i shows which element from X is considered and the blue j shows which element from X is considered.

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots \\ \vdots & \ddots & \\ d_{m1} & & d_{mm} \end{bmatrix}$$

With the distance matrix D $\|x - x\|^2$ it remains to calculate the remaining parts from the Gaussian Kernel formula $f_i = exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)$ in order to receive the Kernel Matrix K.

Sigma $\boldsymbol{\sigma}$ is usually a parameter which can be adjusted, a good starting point is to set $\sigma = 1$ and to see how the outcome will be.

Note, a large $\boldsymbol{\sigma}$ widens the output (increases the bias and lowers variance), a smaller σ makes it pointier (increases the variance and lowers the bias).

Step 6: Calculating the Kernel for each element $d \in D$ results in a Kernel Matrix.

$$\forall d \in D : exp\left(-\frac{d}{2\sigma^2}\right) \Rightarrow F = \begin{bmatrix} f_{11} & f_{12} & \dots \\ \vdots & \ddots & \\ f_{m1} & & f_{mm} \end{bmatrix}$$

Since a Kernel is considered as a feature f, the resulting Kernel Matrix can be denoted as F

x is **destributed as** Gaussian Normal distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma^2}$
Note, σ is the "standard deviation".

$x \sim \mathcal{N}(\mu; \sigma^2)$

$p(x; \mu, \sigma^2)$ is the normalized probability density as parameterized by the feature vector x. Therefore ϵ is a threshold condition obn

$$p(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Multivariate Gaussian

NOTE:

$\Sigma \in \mathbb{R}^{n \times n}$, $X \in \mathbb{R}^{m \times n}$,$x \in \mathbb{R}^n$, $\mu \in \mathbb{R}^n$
$i \in \{1, 2, ..., m\}$
$x^{(i)} = (x_{i1}, x_{i2}, ..., x_{in}) \in X$

$$X = \{x^{(1)}, x^{(2)}, ..., x^{(m)}\} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

Σ is the Covariance Matrix in this case, it has nothing to do with a Sum!!!
Remember the calculation:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (X^{(i)})(X^{(i)})^T = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T = \frac{1}{m} * X' * X$$

$$\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$$

Σ^{-1} is the inverse Covariance Matrix
Remember the calculation:

$$\Sigma \cdot \Sigma^{-1} = I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

$$p(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right)$$