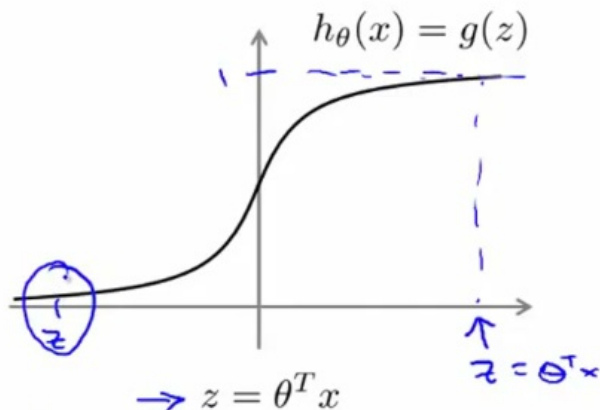


Alternative view of logistic regression

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



If $y = 1$, we want $h_{\theta}(x) \approx 1$, $\theta^T x \gg 0$

If $y = 0$, we want $h_{\theta}(x) \approx 0$, $\theta^T x \ll 0$

支持向量机具有和逻辑回归相似的代价函数，只不过是直线：

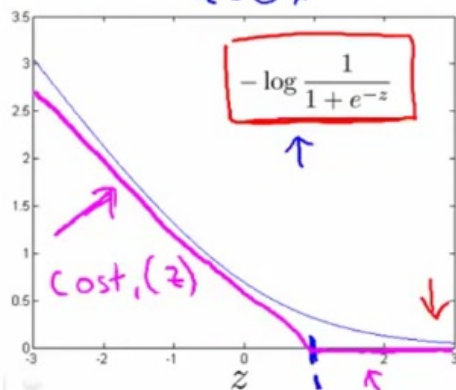
Alternative view of logistic regression (x, y)

Cost of example: $-(y \log h_{\theta}(x) + (1 - y) \log(1 - h_{\theta}(x)))$ ←

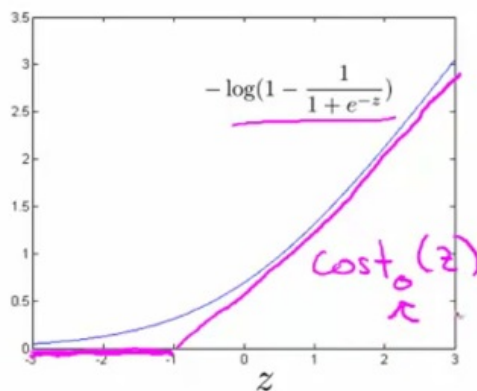
$$= -y \log \frac{1}{1 + e^{-\theta^T x}} - (1 - y) \log \left(1 - \frac{1}{1 + e^{-\theta^T x}}\right)$$

If $y = 1$ (want $\theta^T x \gg 0$):

$$z = \theta^T x$$



If $y = 0$ (want $\theta^T x \ll 0$):



SVM代价函数（注意去掉了常数项1/m，这样不影响最小化效果）：

Support vector machine

Logistic regression:

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

Support vector machine:

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

$\min_u (u-5)^2 + 1 \rightarrow u=5$
 $\min_u 10(u-5)^2 + 10 \rightarrow u=5$

$A + \lambda B \leftarrow$
 $\rightarrow C \quad A + B \leftarrow$
 $C = \frac{1}{\lambda}$

SVM hypothesis

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

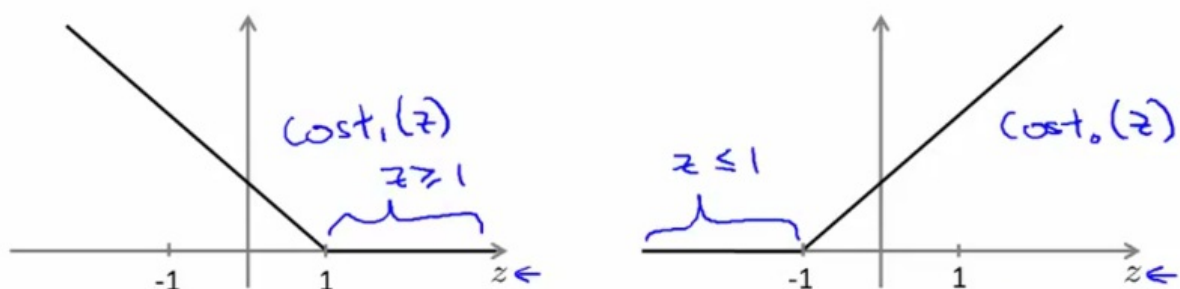
Hypothesis:

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

SVM又称为“大间距分类器”（Large Margin Classifier），直观解释：

Support Vector Machine

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



\rightarrow If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

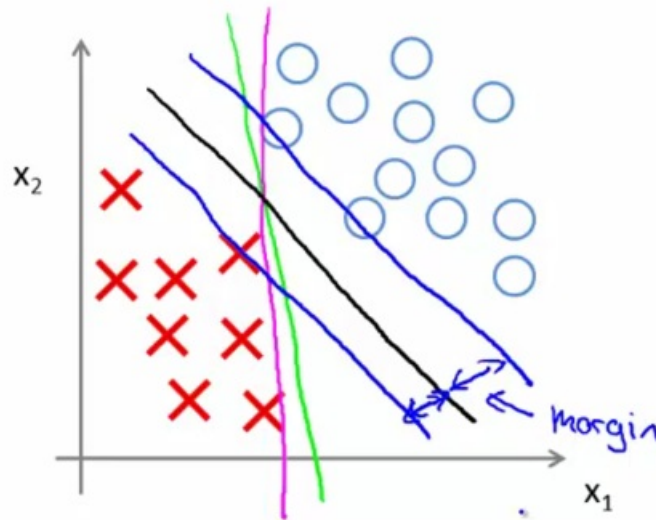
$$\theta^T x \geq 1$$

\rightarrow If $y = 0$, we want $\theta^T x \leq -1$ (not just < 0)

$$\theta^T x \leq -1$$

SVM决策边界（黑线），与样本有一定的间距，所以鲁棒性比较强：

SVM Decision Boundary: Linearly separable case

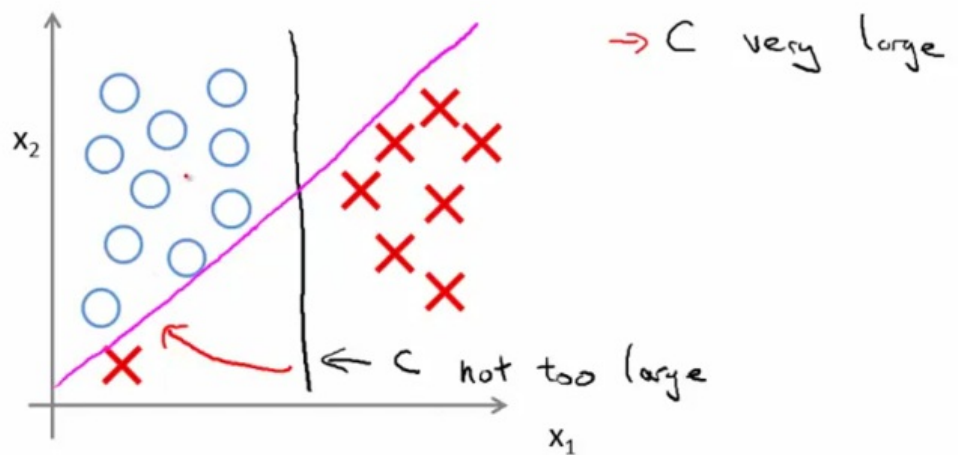


Large margin classifier

我知道我还没有解释这一点

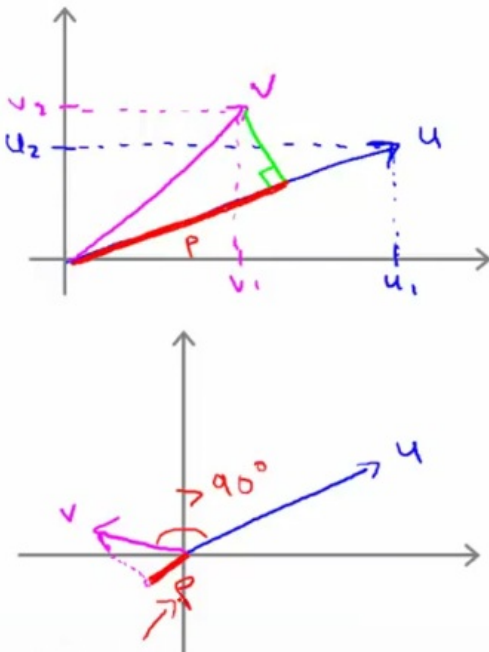
C 是数据集的权重，如果将 C 设置的特别大，那么模型会非常依赖于数据集，那么受单个异常点的影响就特别大，比如，如果 C 设置的特别大，决策边界会是下图粉色线一样：

Large margin classifier in presence of outliers



数学原理：

Vector Inner Product



$$\rightarrow u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \rightarrow v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^T v = ? \quad [u_1 \ u_2] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$\|u\| = \text{length of vector } u \\ = \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$$

$p = \text{length of projection of } v \text{ onto } u.$

$$\text{Signed } u^T v = \underline{p \cdot \|u\|} \leftarrow = v^T u \\ = u_1 v_1 + u_2 v_2 \leftarrow p \in \mathbb{R}$$

$$u^T v = p \cdot \|u\|$$

$$p < 0$$

$$\omega = (\sqrt{\omega})^2$$

SVM Decision Boundary

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

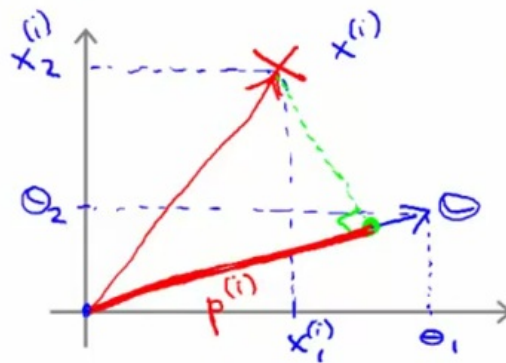
$$\text{s.t. } \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$

$$\rightarrow \theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

$$\text{Simplification: } \underline{\theta_0 = 0.} \quad \underline{n=2}$$

$$= \|\theta\| \\ \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad \theta_0 = 0$$

$$\theta^T x^{(i)} = ? \\ \uparrow \quad \uparrow \\ u^T v$$



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

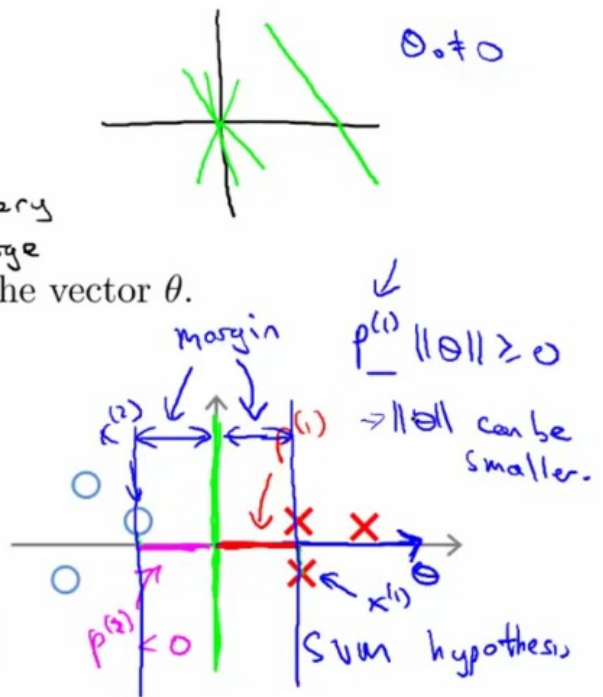
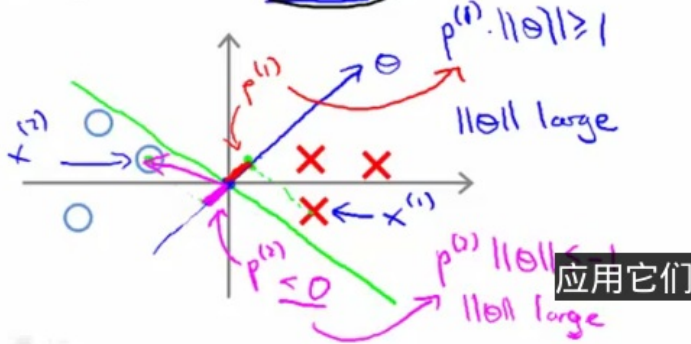
SVM Decision Boundary

$$\rightarrow \min_{\theta} \frac{1}{2} \sum_{j=1}^n \theta_j^2 = \frac{1}{2} \|\theta\|^2 \leftarrow$$

$$\text{s.t. } \begin{cases} p^{(i)} \cdot \|\theta\| \geq 1 & \text{if } y^{(i)} = 1 \\ p^{(i)} \cdot \|\theta\| \leq -1 & \text{if } y^{(i)} = -1 \end{cases} \quad \left. \begin{array}{l} C \text{ vary} \\ \text{large} \end{array} \right\}$$

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ .

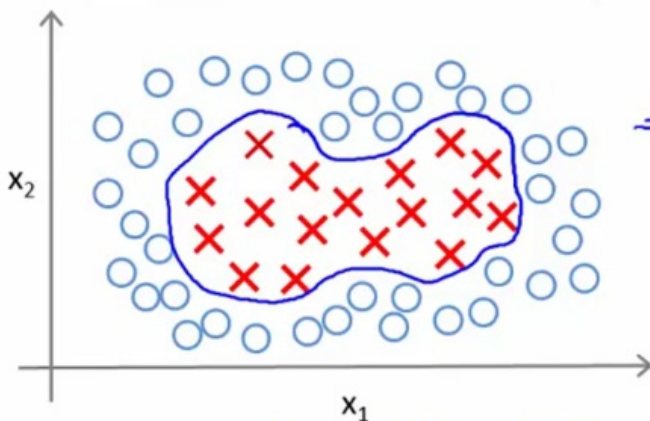
Simplification: $\theta_0 = 0$



Andrew Ng

核函数 (Kernels) : 非线性问题中如果选择更好的特征 :

Non-linear Decision Boundary



Predict $y = 1$ if

$$\rightarrow \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \dots \geq 0$$

$$h_{\theta}(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \dots \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

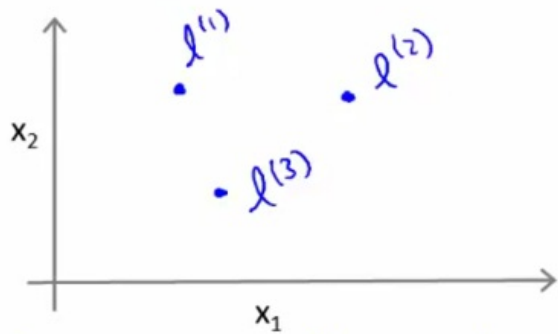
$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \dots$$

$$f_1 = x_1, f_2 = x_2, f_3 = x_1 x_2, f_4 = x_1^2, f_5 = x_2^2, \dots$$

我们之前谈到

Is there a different / better choice of the features f_1, f_2, f_3, \dots ?

Kernel



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

Given x :

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

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

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

它们都称作核函数
kernel (Gaussian kernels)

上图中相似性函数即核函数。上图所用的为高斯核函数。

Kernels and Similarity

$$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right) = \exp\left(-\frac{\sum_{j=1}^n (x_j - l_j^{(1)})^2}{2\sigma^2}\right)$$

If $x \approx l^{(1)}$:

$$f_1 \approx \exp\left(-\frac{0^2}{2\sigma^2}\right) \approx 1$$

$$\begin{array}{lcl} l^{(1)} & \rightarrow & f_1 \\ l^{(2)} & \rightarrow & f_2 \\ l^{(3)} & \rightarrow & f_3 \end{array}$$

If x is far from $l^{(1)}$:

$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \approx 0.$$

图示：

Example:

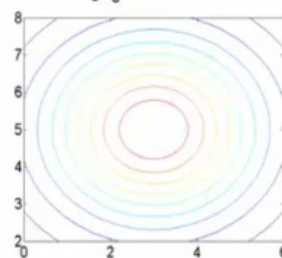
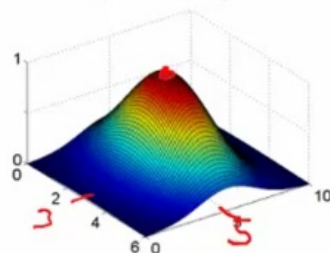
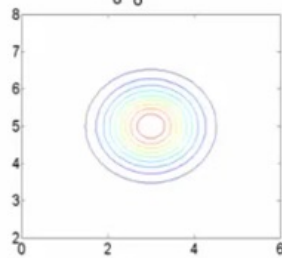
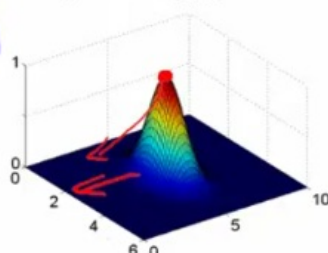
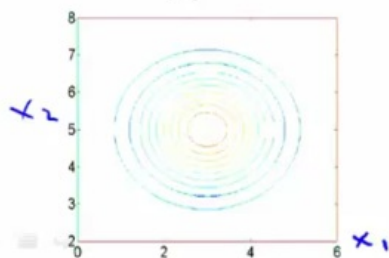
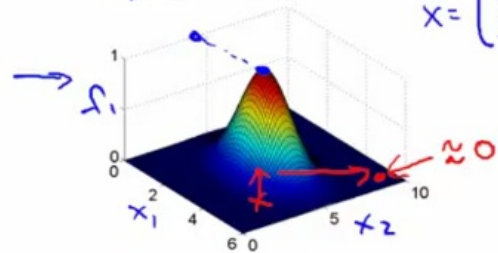
$$\rightarrow l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$$\rightarrow \sigma^2 = 1$$

$$x = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

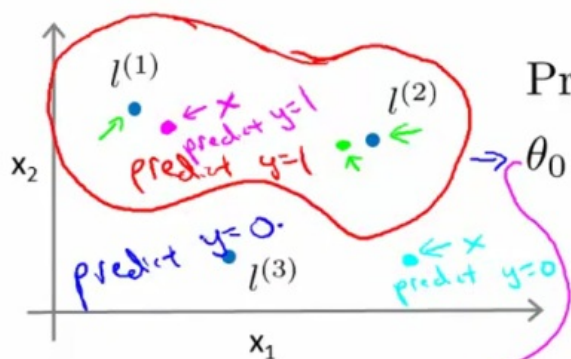
$$\sigma^2 = 0.5$$

$$\sigma^2 = 3$$



Andrew N

靠近标记点预测为1，远离标记点预测为0：



Predict "1" when

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

$$\uparrow \quad \uparrow \quad \uparrow$$

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

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

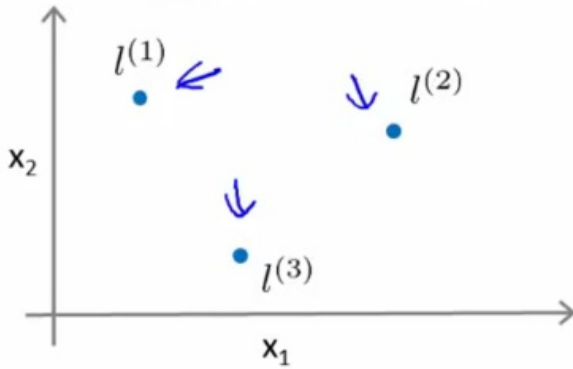
$$\rightarrow \theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0 = -0.5 + 1 = 0.5 \geq 0$$

$$f_1, f_2, f_3 \approx 0$$

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

但是如何选择标记点？数据集样本点位置：

Choosing the landmarks

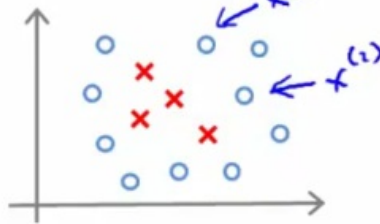


Given x :

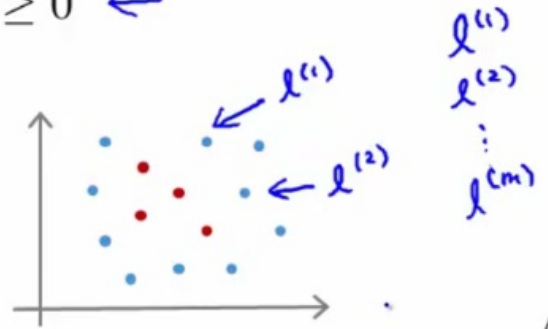
$$\begin{aligned} \rightarrow f_i &= \text{similarity}(x, l^{(i)}) \\ &= \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right) \end{aligned}$$

Predict $y = 1$ if $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

Where to get $l^{(1)}, l^{(2)}, l^{(3)}, \dots$?



的位置都与



An

选择训练集样本点作为标记点，通过核函数将训练样本映射为特征 f 。

对于一个训练样本 (x_i, y_i) ，通过核函数计算他的特征向量 f ，其中：

$f_1(i) = \text{sim}(x(i), l(1))$, $f_2(i) = \text{sim}(x(i), l(2))$ 每个训练样本都对应一个 f 向量。

SVM with Kernels

- \rightarrow Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$,
- \rightarrow choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \dots, l^{(m)} = x^{(m)}$.

Given example x :

$$\begin{aligned} \rightarrow f_1 &= \text{similarity}(x, l^{(1)}) \\ \rightarrow f_2 &= \text{similarity}(x, l^{(2)}) \\ &\vdots \end{aligned}$$

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

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

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

和相似度函数后

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

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ f_2^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \quad f_0^{(i)} = 1$$

Andrew N

使用特征向量进行预测：

SVM with Kernels

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

→ Predict "y=1" if $\theta^T \underline{f} \geq 0$

$$\theta_0 f_0 + \theta_1 f_1 + \dots + \theta_m f_m$$

Training:

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

$$- \sum_j \theta_j^2 = \theta^T \theta \quad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix} \quad (\text{ignore } \theta_0)$$

$M = 10,000$

使用成熟的软件包进行代价函数的计算，但是要注意参数的选择：

SVM parameters:

$C (= \frac{1}{\lambda})$. → Large C: Lower bias, high variance.

(small λ)

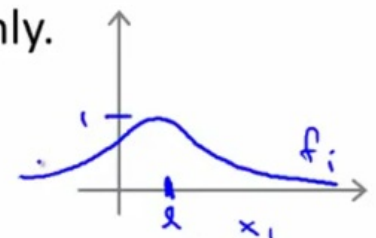
→ Small C: Higher bias, low variance.

(large λ)

σ^2 Large σ^2 : Features f_i vary more smoothly.

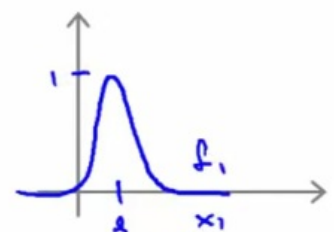
→ Higher bias, lower variance.

$$\exp\left(-\frac{\|x - \mu^{(i)}\|^2}{2\sigma^2}\right)$$



Small σ^2 : Features f_i vary less smoothly.

Lower bias, higher variance.



特征的变化会变得不平滑

使用SVM：现有的软件库：

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters θ .

Need to specify:

→ Choice of parameter C.

Choice of kernel (similarity function):

E.g. No kernel ("linear kernel")

Predict "y = 1" if $\theta^T x \geq 0$


$$\theta_0 + \theta_1 x_1 + \dots + \theta_n x_n \geq 0 \quad \rightarrow \quad \underline{n} \text{ large}, \quad \underline{m} \text{ small} \quad \underline{x \in \mathbb{R}^{n+1}}$$

Gaussian kernel:

$$f_i = \exp\left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2}\right), \text{ where } l^{(i)} = x^{(i)}.$$

Need to choose $\underline{\sigma^2}$.

$x \in \mathbb{R}^n$, n small
and/or m large



Kernel (similarity) functions:

function $f = \text{kernel}(\underline{x1}, \underline{x2})$

$$f = \exp\left(-\frac{\|\underline{x1} - \underline{x2}\|^2}{2\sigma^2}\right)$$

return

$x \rightarrow \begin{matrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{matrix}$

→ Note: Do perform feature scaling before using the Gaussian kernel.

$\rightarrow \|x - l\|^2$

$$v = x - l$$

$$\|v\|^2 = v_1^2 + v_2^2 + \dots + v_n^2$$

$$= \underbrace{(x_1 - l_1)^2}_{1000 \text{ feet}^2} + \underbrace{(x_2 - l_2)^2}_{1-5 \text{ bedrooms}} + \dots + (x_n - l_n)^2$$

$x \in \mathbb{R}^n$

其他类型的核函数：

Other choices of kernel

Note: Not all similarity functions $\text{similarity}(x, l)$ make valid kernels.

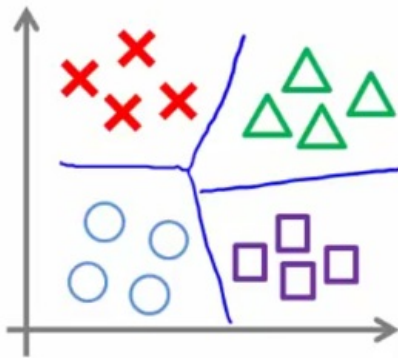
→ (Need to satisfy technical condition called “Mercer’s Theorem” to make sure SVM packages’ optimizations run correctly, and do not diverge).

Many off-the-shelf kernels available:

- Polynomial kernel: $k(x, l) = (x^T l)^2$, $(x^T l)^3$, $(x^T l + 1)^3$, $(x^T l + 5)^4$ (degree 4)
More esoteric: String kernel, chi-square kernel, histogram intersection kernel, 这些更加难懂的核函数

多类分类问题：内置

Multi-class classification



$$y \in \{1, 2, 3, \dots, K\}$$

Many SVM packages already have built-in multi-class classification functionality.

→ Otherwise, use one-vs.-all method. (Train K SVMs, one to distinguish $y = i$ from the rest, for $i = 1, 2, \dots, K$), get $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(K)}$
Pick class i with largest $\theta^{(i)}$ 对于更为常见的情况

逻辑回归和SVM选择：

Logistic regression vs. SVMs

n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

→ If n is large (relative to m): (e.g. $n \geq m$, $n = 10,000$, $m = 10 \dots 1000$)

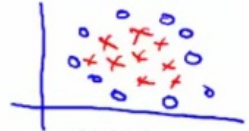
→ Use logistic regression, or SVM without a kernel ("linear kernel")

→ If n is small, m is intermediate: ($n = 1-1000$, $m = 10-10,000$) ←

→ Use SVM with Gaussian kernel

If n is small, m is large: ($n = 1-1000$, $m = 50,000+$)

→ Create/add more features, then use logistic regression or SVM without a kernel



→ [↑]Neural network likely to work well for most of these settings, but may be slower to train.