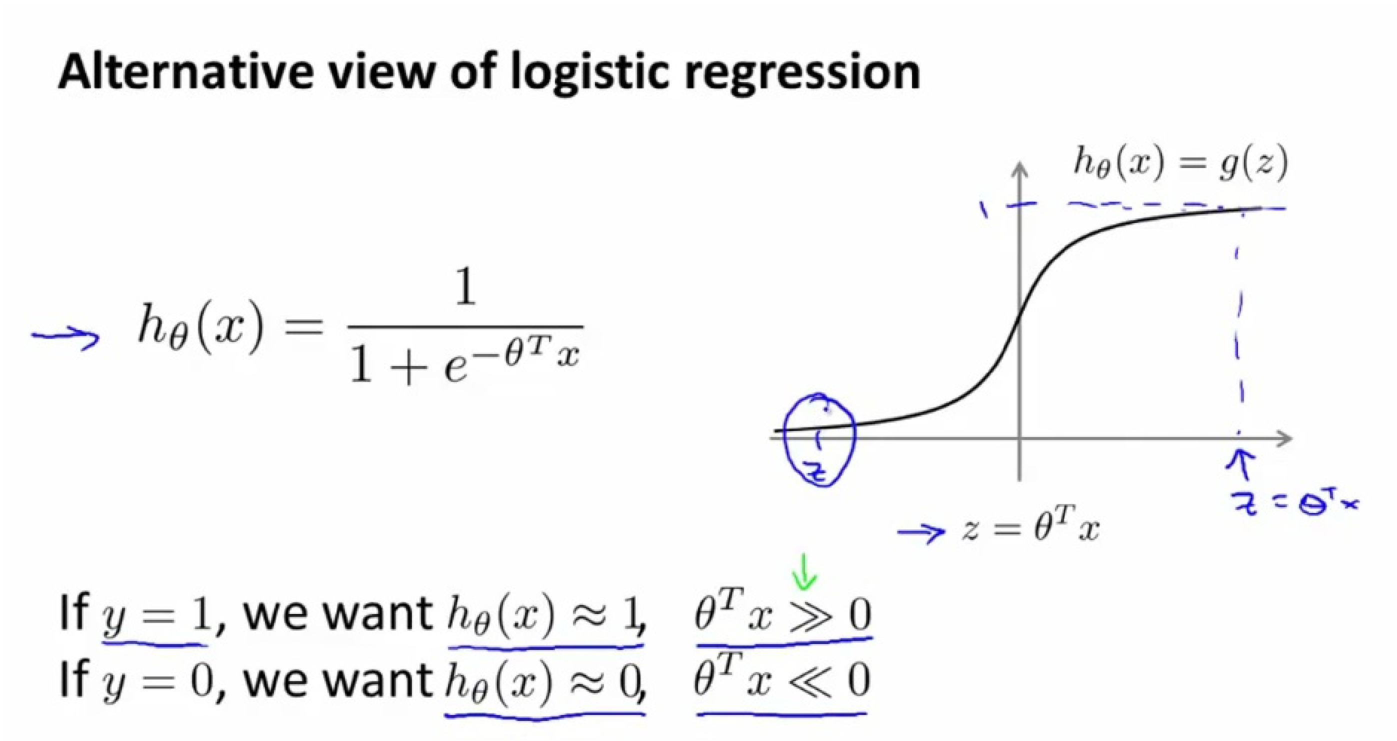
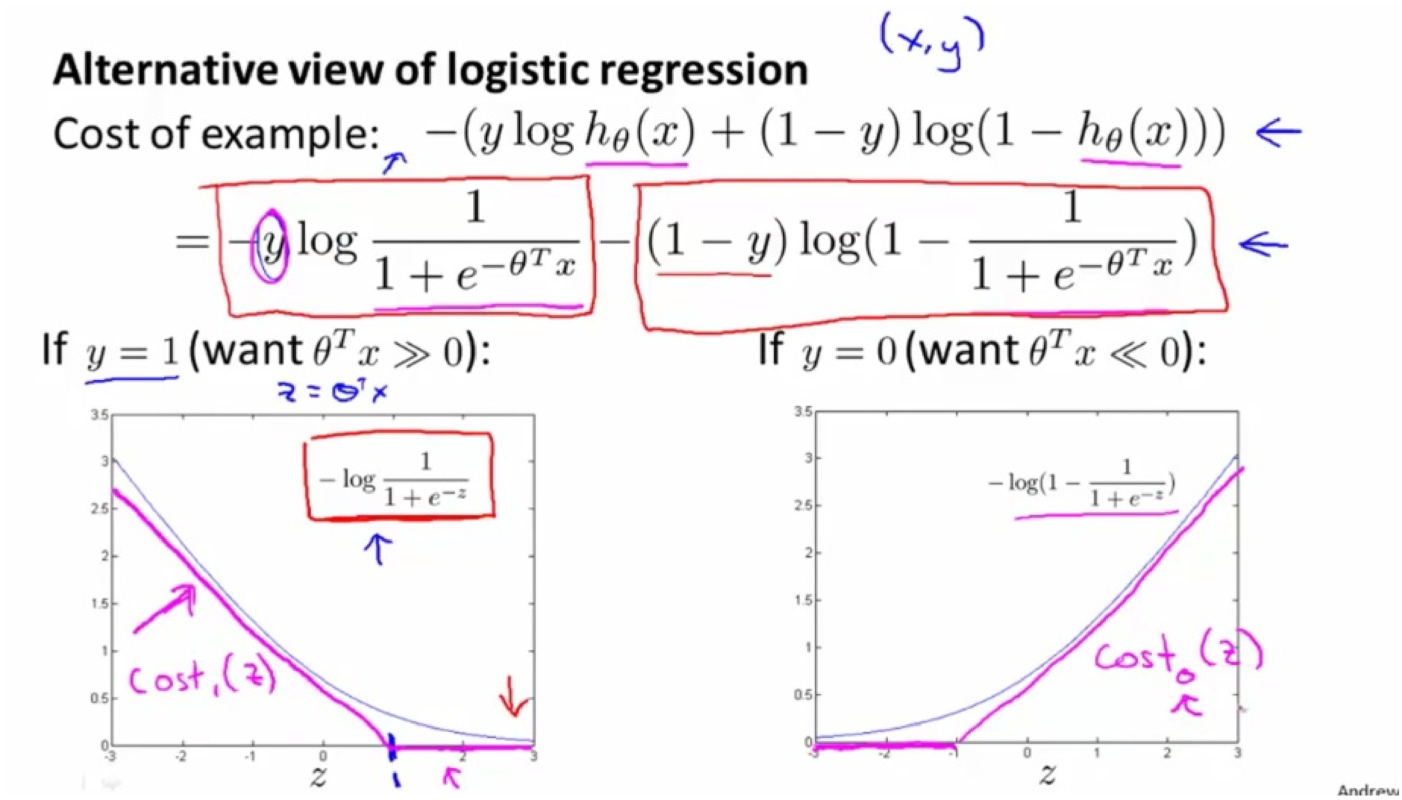
支持向量机（Support Vector Machine）

逻辑回归回顾：



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：



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:

$$\min_\theta \underbrace{\frac{1}{m}\left[\sum_{i=1}^m y^{(i)}\underbrace{\left(-\log h_\theta(x^{(i)})\right)}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\left((-\log(1-h_\theta(x^{(i)}))\right)}_{cost_0(\theta^T x^{(i)})}\right]}_{A} + \underbrace{\frac{\lambda}{2m}\sum_{j=1}^n \theta_j^2}_{B}$$

Support vector machine:

$$\min_\theta \cancel{\frac{1}{m}} \sum_{i=1}^m y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)}) + \frac{\lambda}{2\cancel{m}}\sum_{j=0}^n \theta_j^2$$

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

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

$A + \lambda B$

$C A + B$

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

SVM hypothesis

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

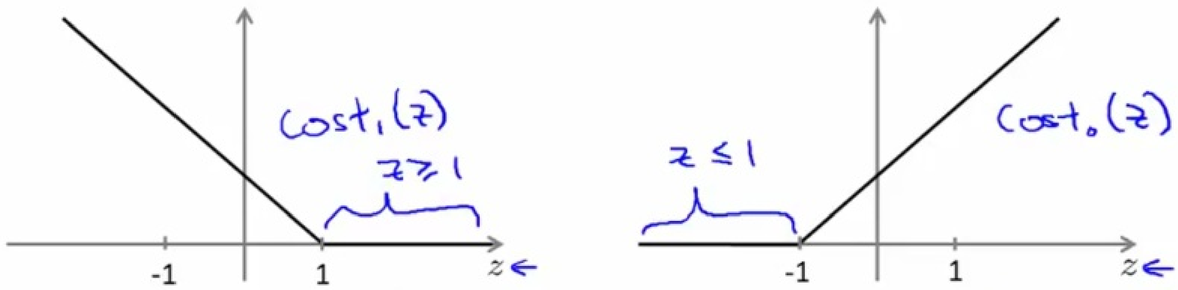
Hypothesis:

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

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

Support Vector Machine

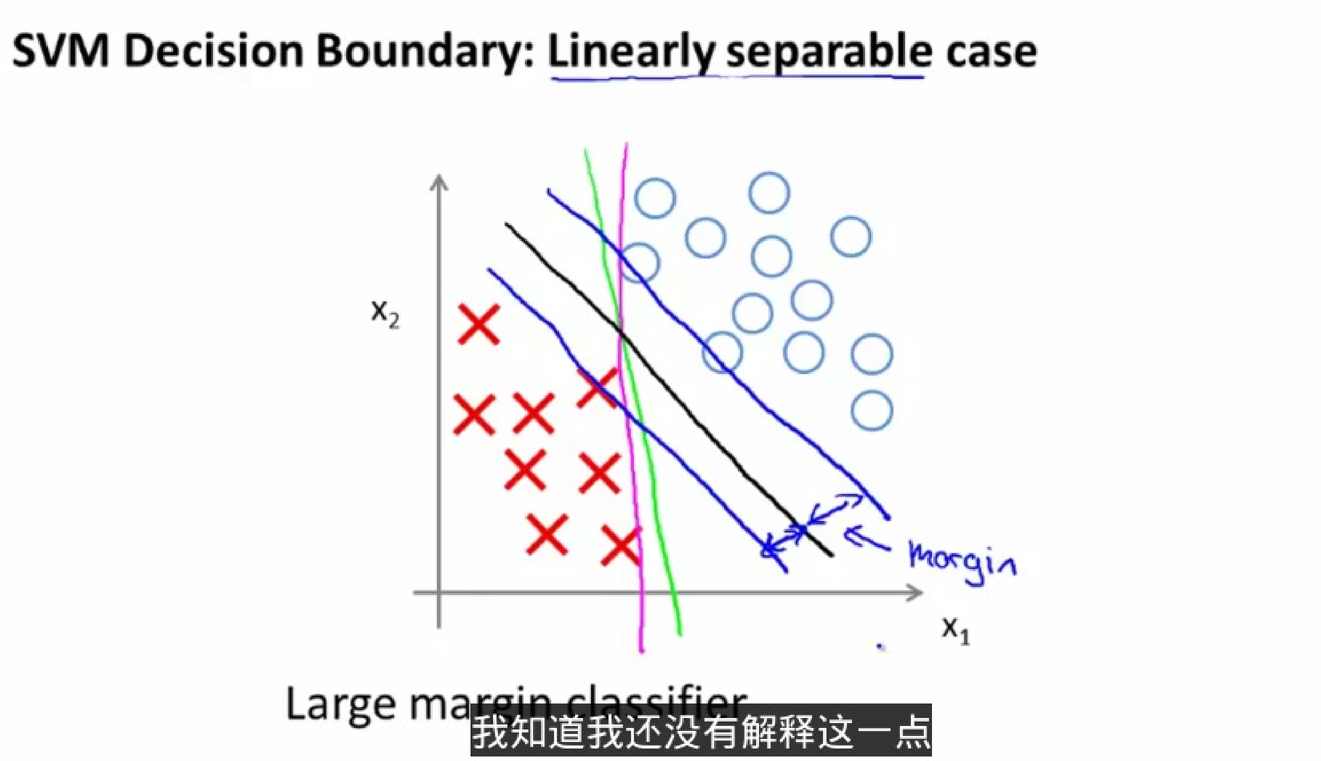
$$\rightarrow \min_\theta C\sum_{i=1}^m \left[y^{(i)} \underline{cost_1(\theta^T x^{(i)})} + (1-y^{(i)}) \underline{cost_0(\theta^T x^{(i)})}\right] + \frac{1}{2}\sum_{i=1}^n \theta_j^2$$



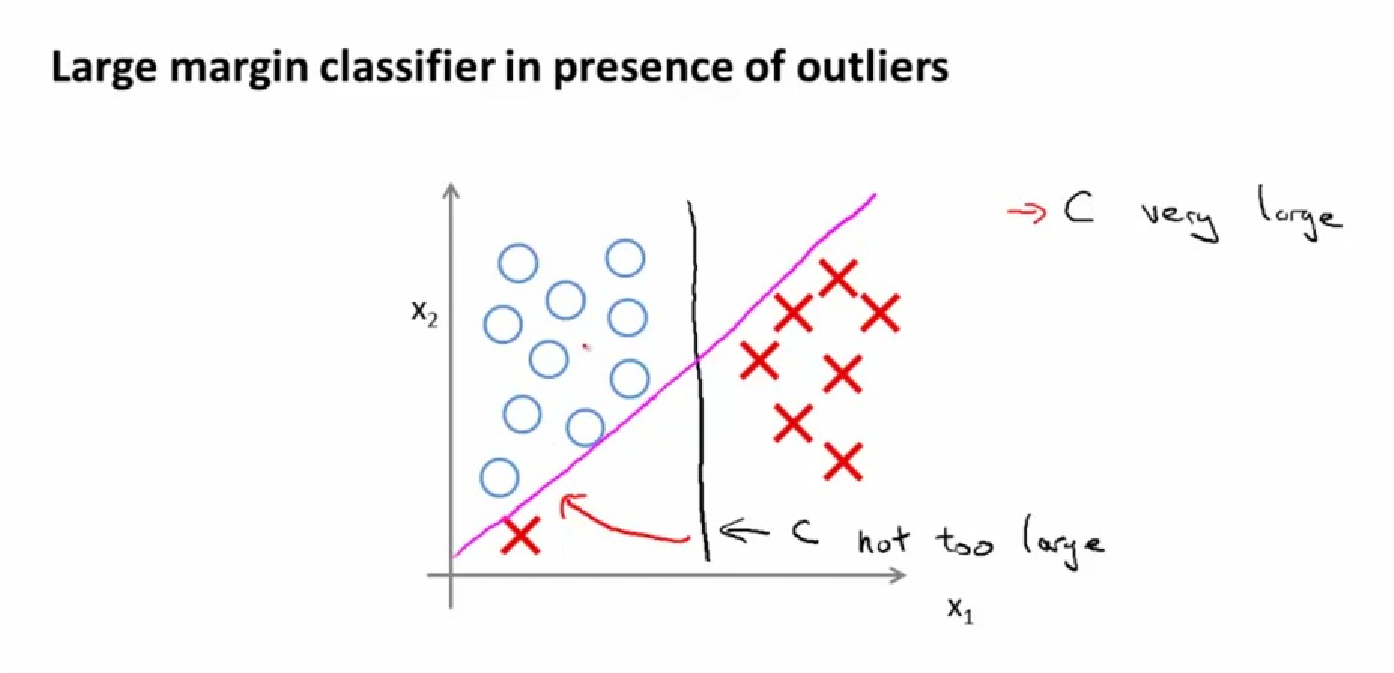
→ If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0) $\theta^T x \geq \cancel{0}\ 1$

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

SVM决策边界（黑线），与样本有一定的间距，所以鲁棒性比较强：

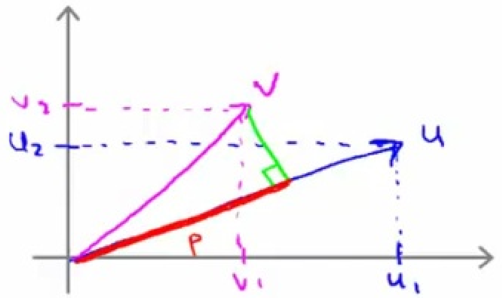


C是数据集的权重，如果将C设置的特别大，那么模型会非常依赖于数据集，那么受单个异常点的影响就特别大，比如，如果C设置的特别大，决策边界会是下图粉色线一样：



数学原理：

Vector Inner Product



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

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

$\|u\|$ = length of vector u

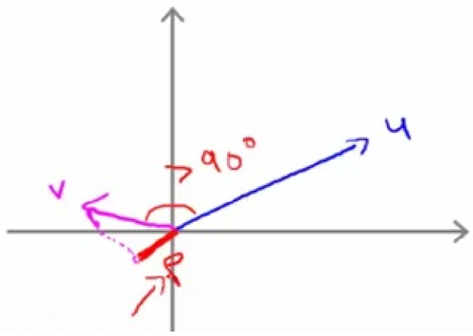
$= \sqrt{u_1^2 + u_2^2} \in \mathbb{R}$

p = length of projection of v onto u.

signed

$u^T v = \underline{p} \cdot \underline{\|u\|} \leftarrow \qquad = v^T u$

$= u_1 v_1 + u_2 v_2 \leftarrow \qquad p \in \mathbb{R}$



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

$p < 0$

SVM Decision Boundary

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

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

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

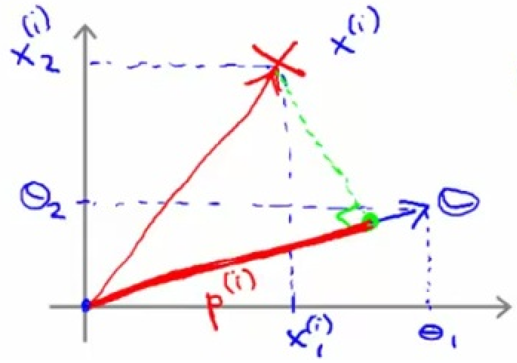
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$

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

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

$\theta^T x^{(i)} = ?$

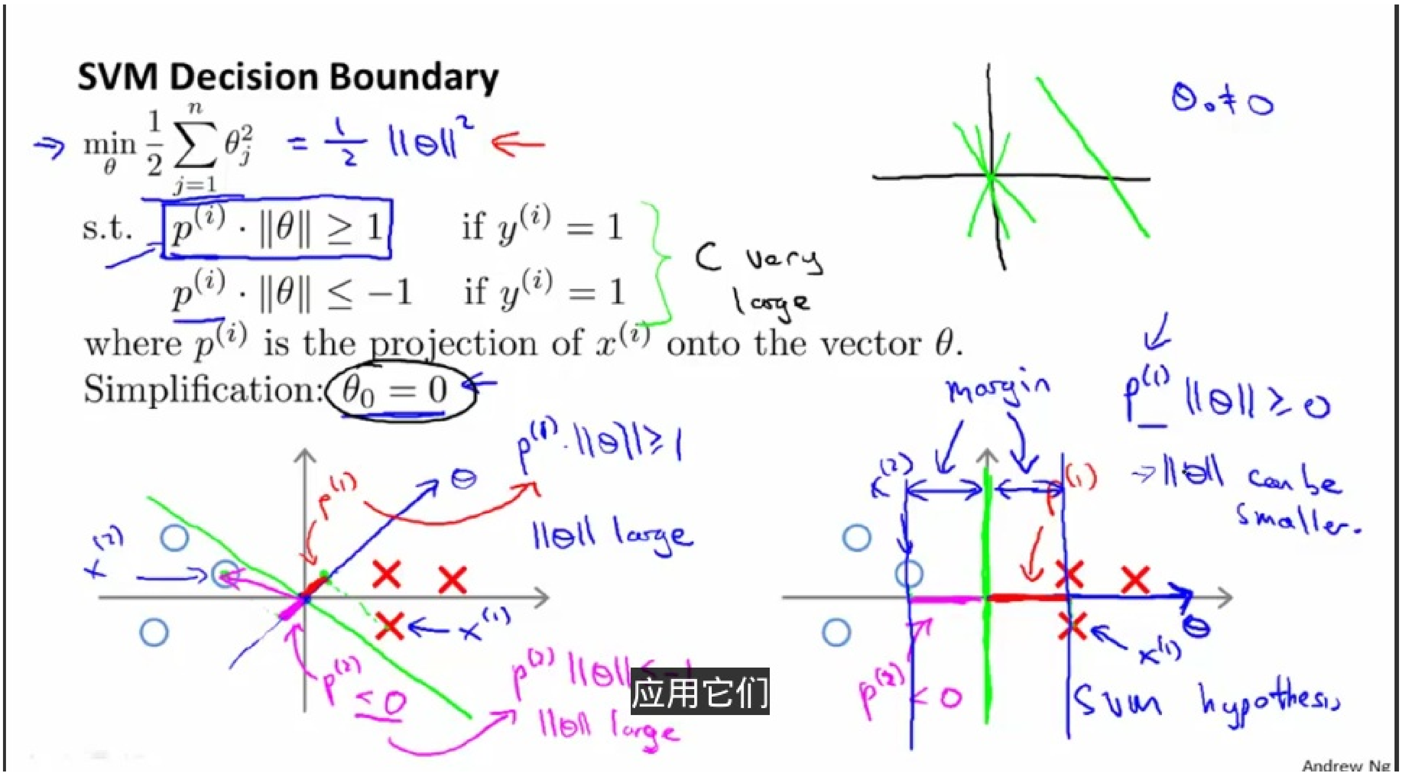
$u^T v$



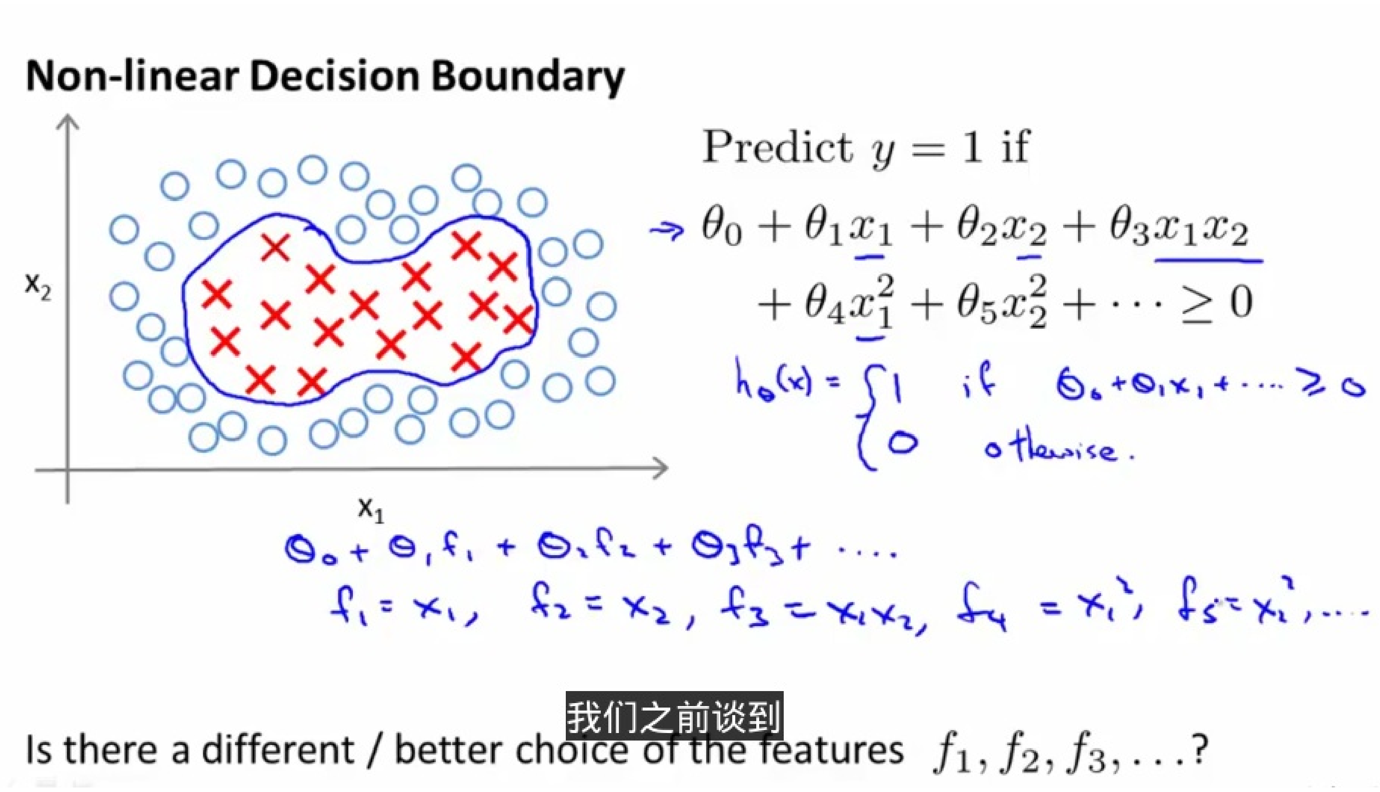
$\theta^T x^{(i)} = p^{(i)} \cdot \|\theta\| \leftarrow$

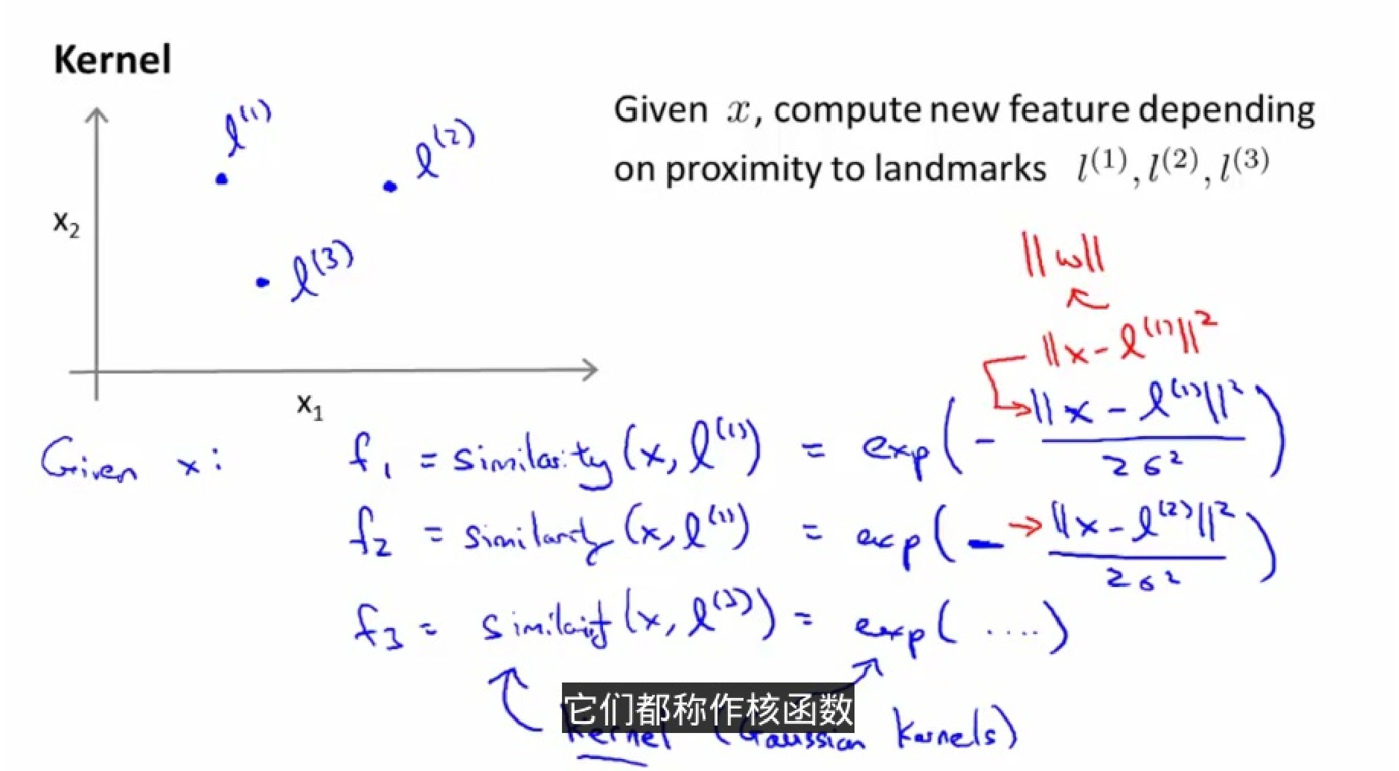
$= \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} \leftarrow$

Andrew N

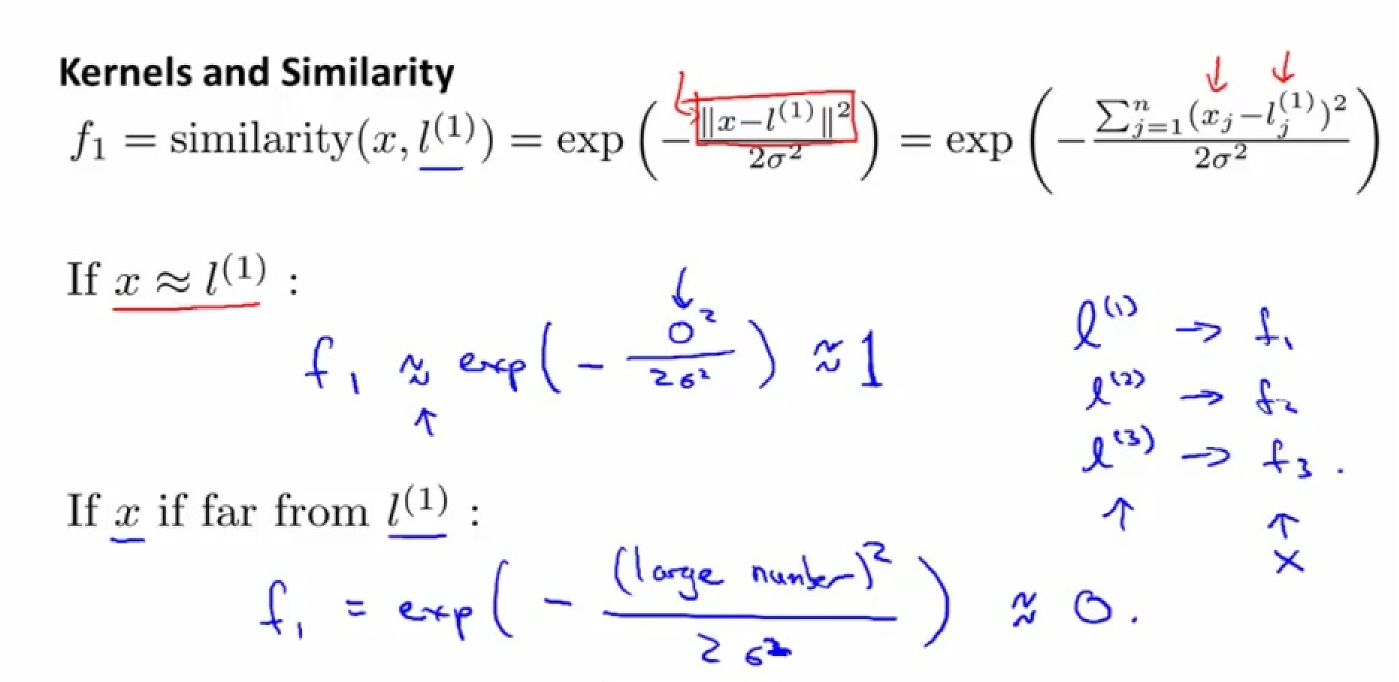


核函数（Kernals）：非线性问题中如果选择更好的特征：

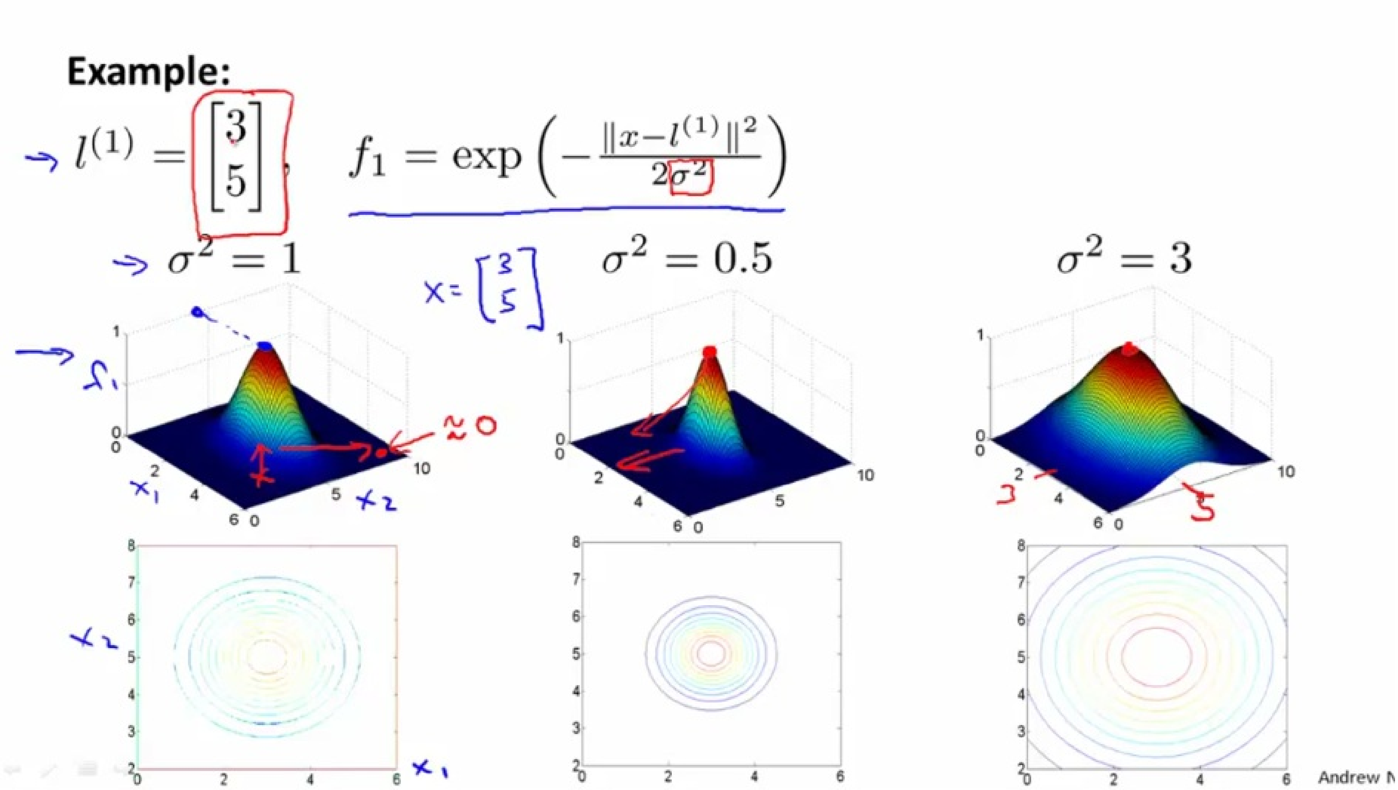




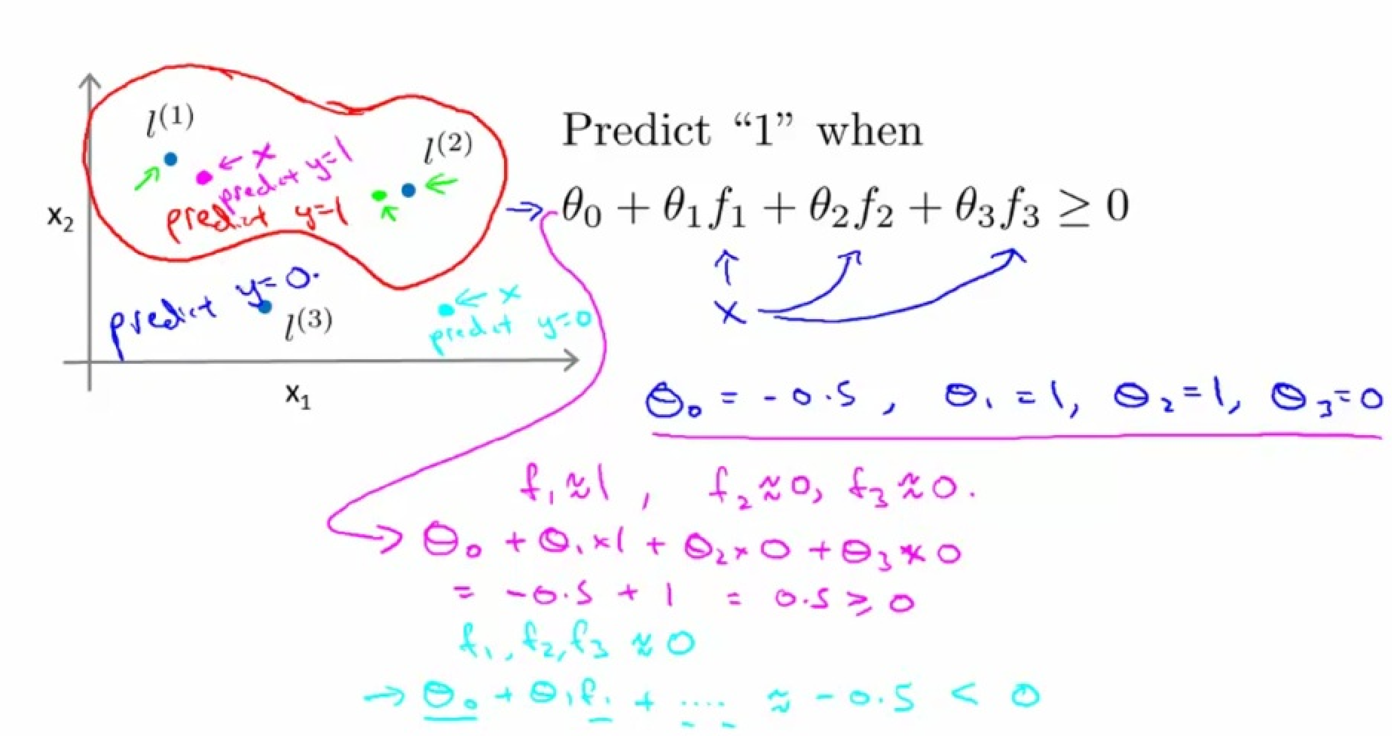
上图中相似性函数即核函数。上图所用的为高斯核函数。



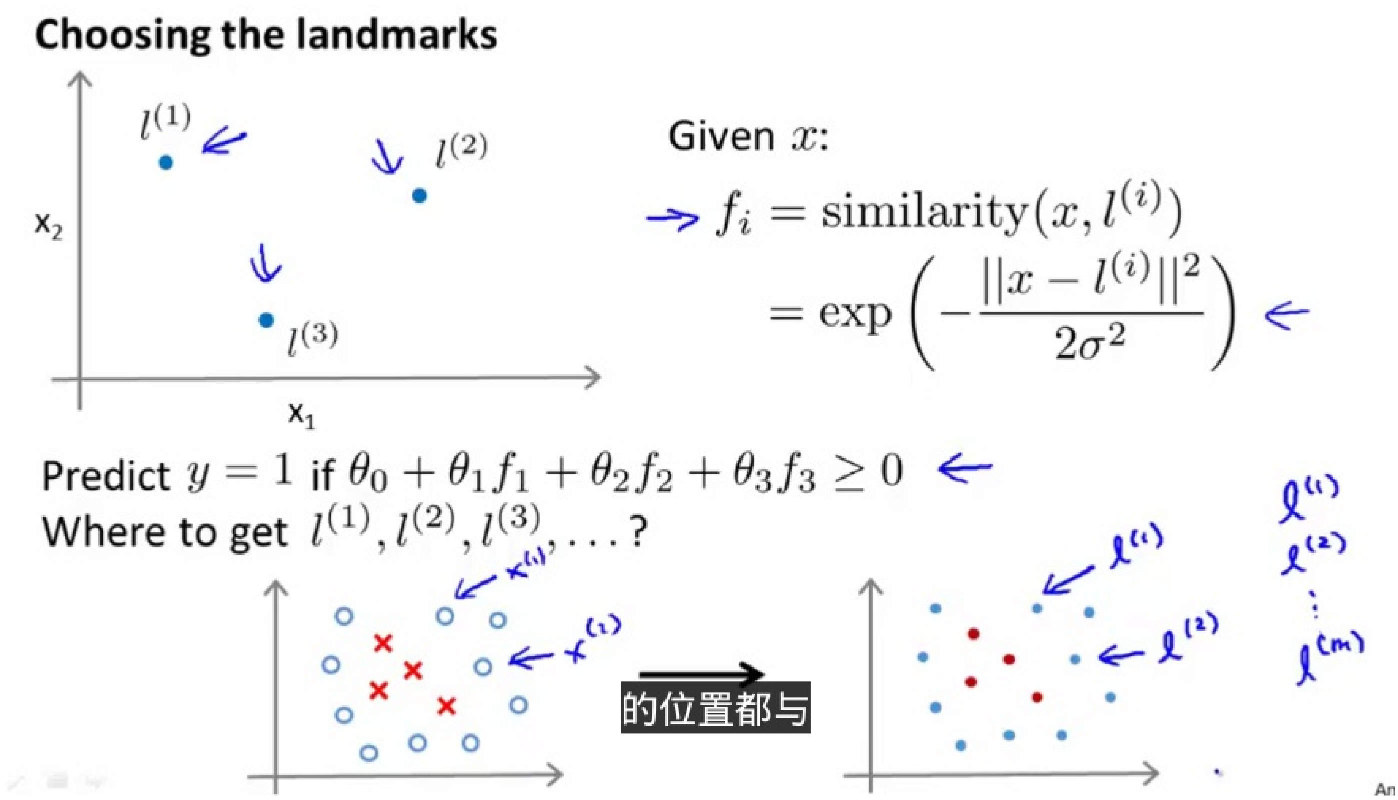
图示：



靠近标记点l预测为1，远离标记点l预测为0：



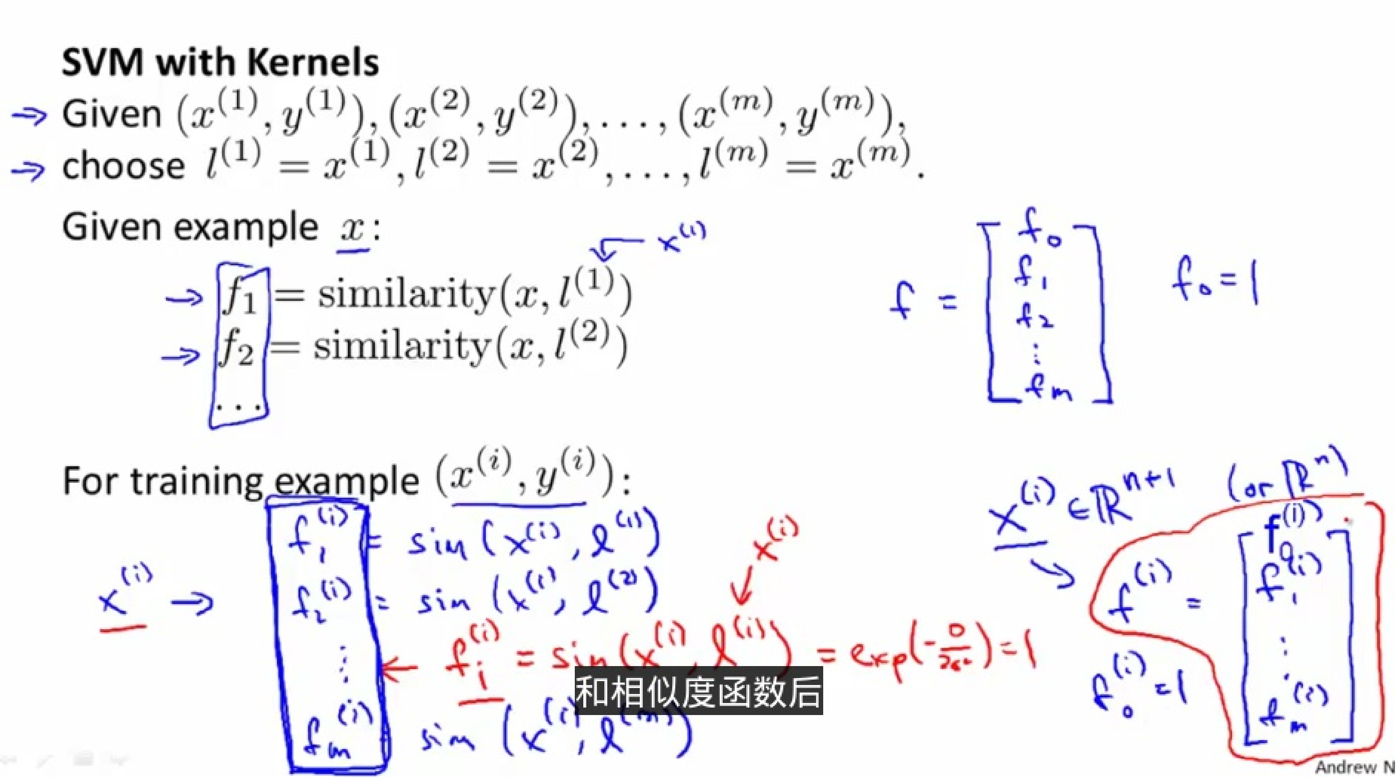
但是如何选择标记点？数据集样本点位置：



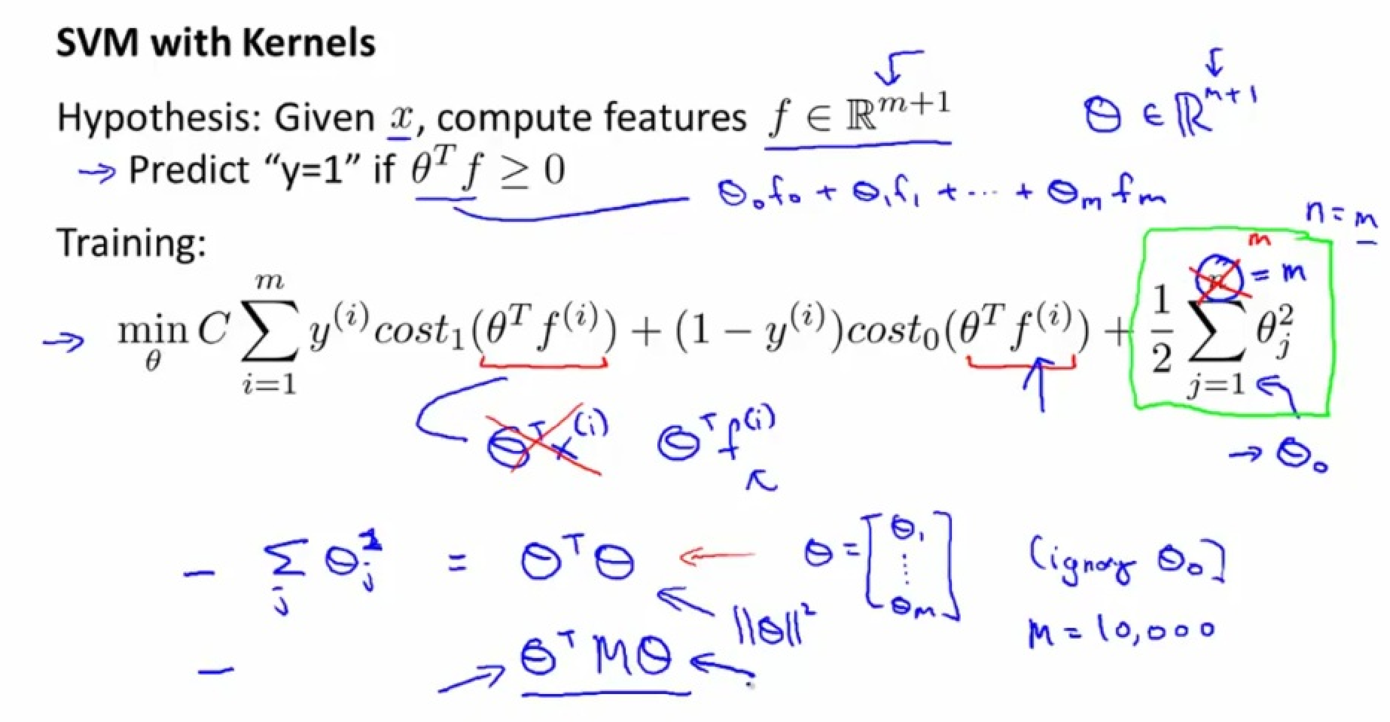
选择训练集样本点作为标记点，通过核函数将训练样本映射为特征f。

对于一个训练样本（xi,yi），通过核函数计算他的特征向量f，其中：

f1(i) = sim(x(i), l(1)), f2(i) = sim(x(i), l(2)) .... 每个训练样本都对应一个f向量。



使用特征向量进行预测：



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：

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 $\theta^T x \geq 0$

$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \geq 0$

→ n large, m small $x \in \mathbb{R}^{n+1}$

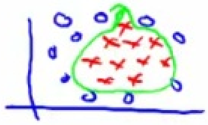
Gaussian kernel:

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

Need to choose σ^2.

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

and/or m large



Kernel (similarity) functions:

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

```
function f = kernel(x1,x2)
```

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

```
return
```

f_i

$x \rightarrow$ f_1, f_2, $\vdots$, f_m

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

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

$\boxed{||x - l||^2}$

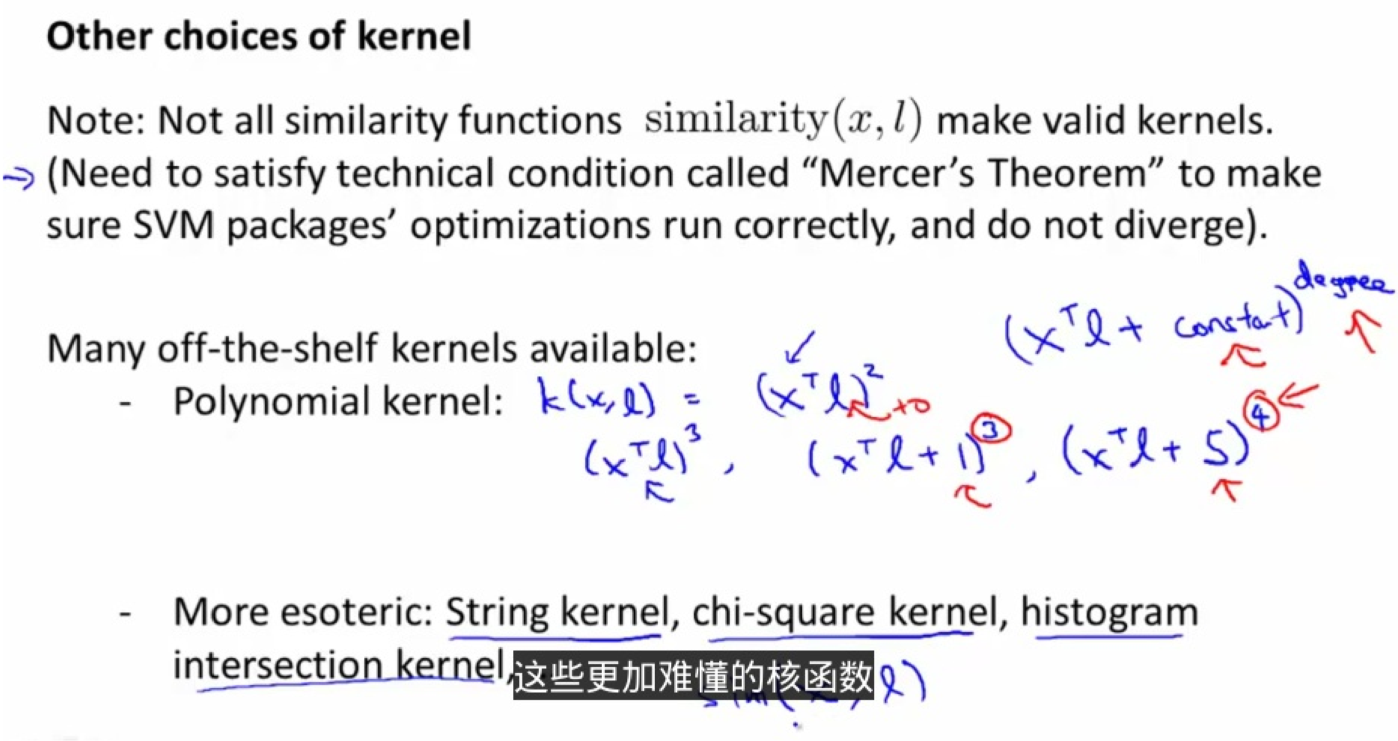
$v = x - l$

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

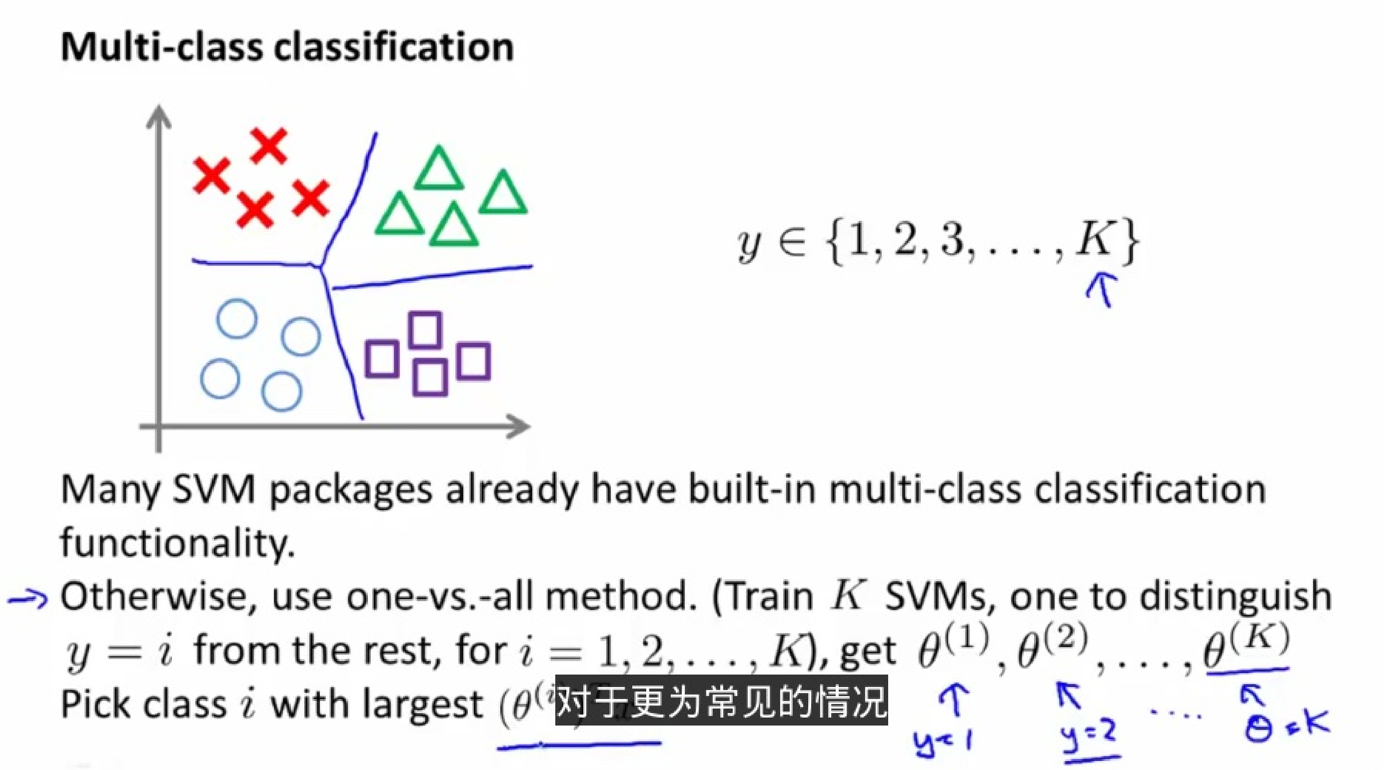
$= (x_1 - l_1)^2 + (x_2 - l_2)^2 + \cdots + (x_n - l_n)^2$.

1000 feet^2 1-5 bedrooms

其他类型的核函数：



多类分类问题：内置

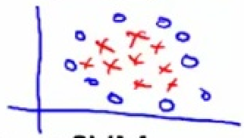


逻辑回归和SVM选择：

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. $n \geq m$, $n = 10{,}000$, $m = 10 \cdots 1000$)
- 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.