①✓ supervised learning 的算法中，各算法比较相似，但 Data Amount 等因素往往起了决定作用。
Feture 选择
Regularization 的参数

② 先从LR开始，再讲怎么修改，引出SVM。

Support Vector Machines

Optimization objective

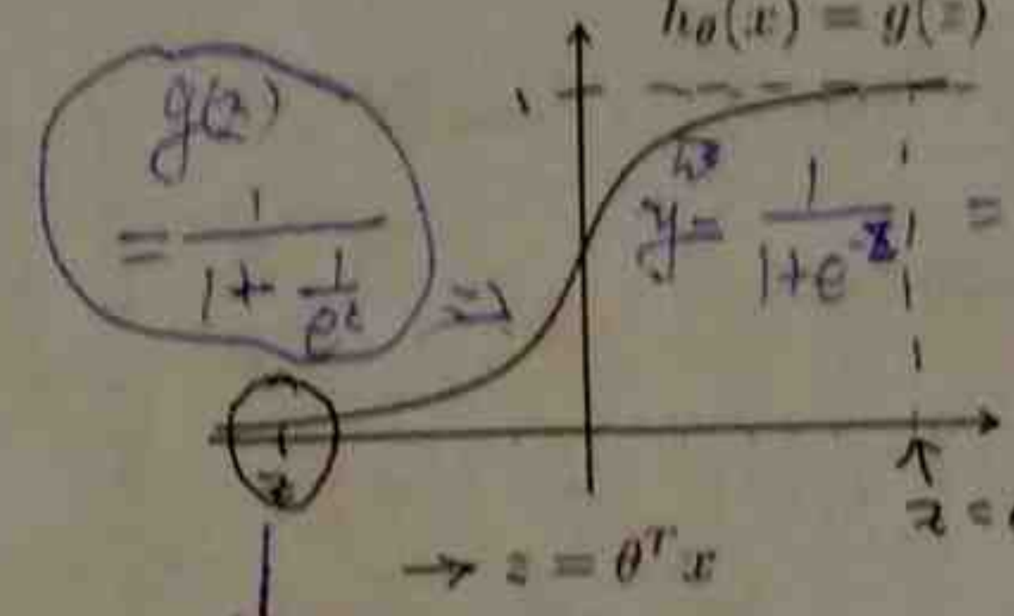
Machine Learning

① 但SVM非常powerful 在both 工业界和学术界
③ SVM 与神经网络相比 clearner，与Logistic Regression相比 powerful

Alternative view of logistic regression

先从讲LR开始。
hypothes function

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



$z \rightarrow +\infty$ $y \rightarrow 1$
$z = 0$ $y = \frac{1}{2}$
$z \rightarrow -\infty$ $y \rightarrow 0$

If $y = 1$, we want $h_\theta(x) \approx 1$, $\theta^T x \gg 0$ $\rightarrow +\infty$
If $y = 0$, we want $h_\theta(x) \approx 0$, $\theta^T x \ll 0$ $\rightarrow -\infty$
$z = \theta^T x$

Andrew Ng

②

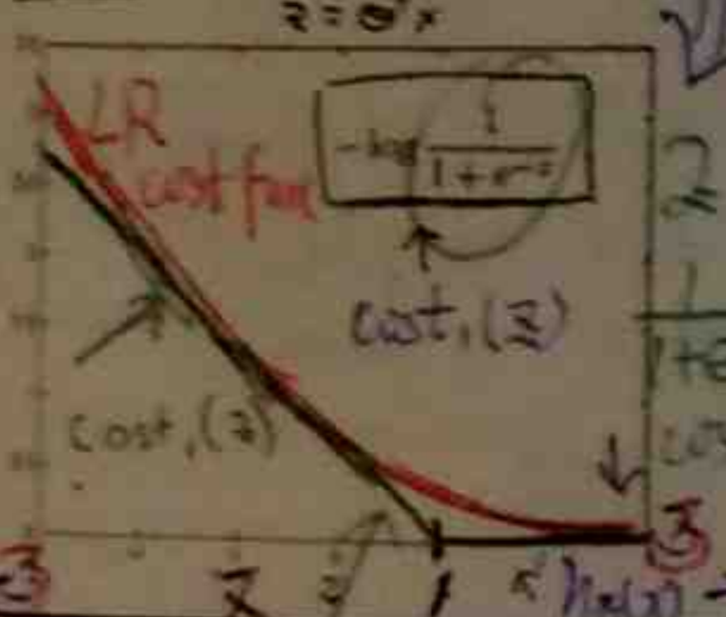
Alternative view of logistic regression

(x,y) 每个example contribute a term of cost function

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

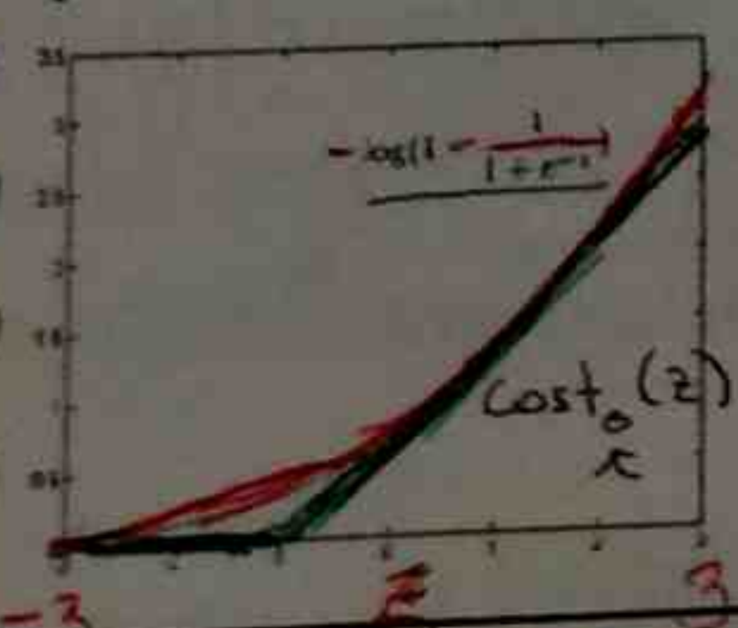
$$= \boxed{-y \log \frac{1}{1+e^{-\theta^T x}}} - \boxed{(1-y)\log(1 - \frac{1}{1+e^{-\theta^T x}})} \leftarrow \text{代入 } h_\theta(x)$$

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



y=1 y=0
$z \in (0, +\infty)$
$\frac{1}{1+e^{-z}} \in (0,1)$
cost $\in (+, 0)$

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



Andrew Ng

$h_\theta(x) \rightarrow +\infty$
$z = h_\theta(x) \rightarrow -\infty$
修改cost function
$cost(z) \rightarrow 0$
$z = \theta^T x \rightarrow +\infty$ $cost(z) \rightarrow 0$
$\frac{1}{1+e^{-z}} \rightarrow 1$
$y = -\log h$
$1 - z_0 = \frac{1}{1+e^{-z}} \rightarrow 0$
$z_0 = 1 - \frac{1}{1+e^{-z}} \rightarrow 1$
$\log(1 - \frac{1}{1+e^{z}}) \rightarrow 0$
$cost(z_0) \rightarrow 0$

Support vector machine

Logistic regression: 最小化cost func

把括号外的"-"移到括号内。

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

有 $\frac{\lambda}{m}$

Support vector machine:

不修改训练出的 θ。
$\min_u ((u-5)^2+1) \rightarrow u = 5$
$\min_u 10((u-5)^2+10) \rightarrow u = 5$

没有 $\frac{1}{m}$ 不改变最优 θ 值

用 $A + \lambda B$ 来控制 θ。
$C = \frac{1}{\lambda}$ C 的作用类似 $\frac{1}{\lambda}$，让
用 $CA + B$ 来控制 θ。
没有 $\frac{\lambda}{m}$ 也以
$C = \frac{1}{\lambda}$
则
$CA + B = A + \lambda B$
?

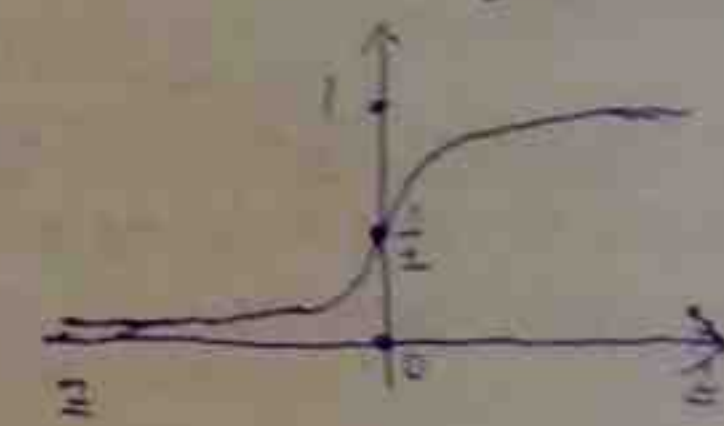
$$\min_\theta C \sum_{i=1}^m \left[y^{(i)} \underline{cost_1(\theta^T x^{(i)})} + (1-y^{(i)}) \underline{cost_0(\theta^T x^{(i)})}\right] + \frac{1}{2}\sum_{i=1}^n \theta_j^2$$

Andrew Ng

只是换了一个 Regularization Term ?
从 λ 换为 C。

hypothesis function of LR.

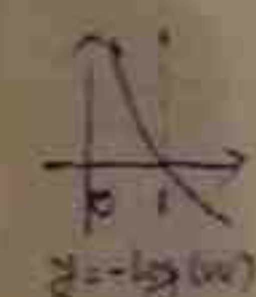
$\boxed{↗}$ $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \Leftrightarrow g(z) = \frac{1}{1+\frac{1}{e^z}}$



$\boxed{↙}$ $y=1$时，$z \to +\infty$，$g(z) \to 1$，$-\log(g(z)) \to 0$

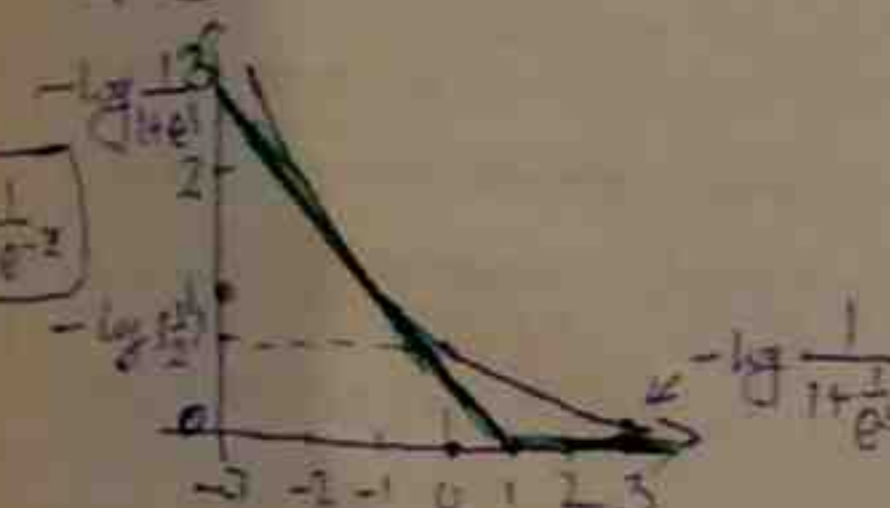
虽然 $g(z) > \frac{1}{2}$，$z>0$，但假定 $z \to -\infty$ 时，

$z \to -\infty$，$g(z) \to 0$，$-\log(g(z)) \to +\infty$



因此

$\boxed{\Gamma(z) = -\log \frac{1}{1+e^{-z}}}$



$\boxed{z = -\theta^T x}$

同理可推 $y=0$ 时.

对正例，预测得越准，cost越小 (→0)
对正例，预测得越不准，cost越大 (→+∞)

SVM: cost function 作些修改　同理可推 $y=0$ 时

分为两段直线

$z>1$ 时，—

$z<1$ 时 ＼

SVM hypothesis

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

Hypothesis:

$z = \theta^T x$

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

不输出概率，直接判断

LR 不同 $z \geq 0$

$h_\theta(x) \begin{cases} \geq 0.5 & \text{if } \theta^T x \geq 0 \quad 期望 \theta^T x \to +\infty \\ < 0.5 & \text{otherwise} \quad 期望 \theta^T x \to -\infty \end{cases}$

$z < 0$

Andrew Ng

Support Vector Machines

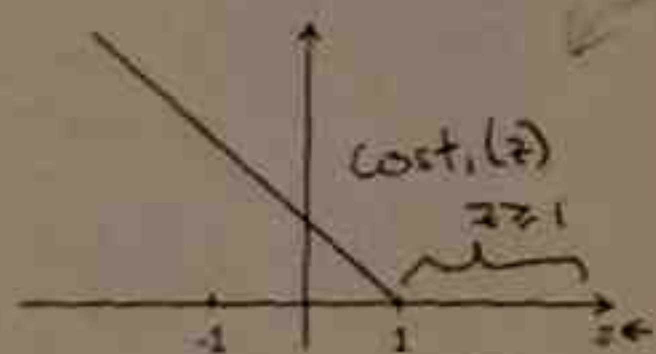
Large Margin Intuition

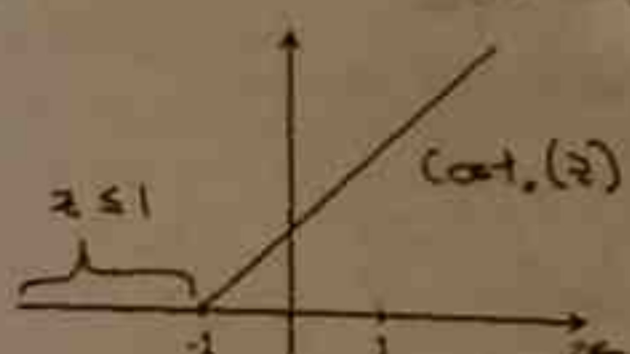
Machine Learning

SVM · Large Margin Classifier

Support Vector Machine

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





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

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

$C = 100{,}000$

当C非常大时

不像LR中，是希望用≥0，<来作为判断依据

但SVM不同，是以≥1，≤-1来作为判断依据

(Large Margin)

safety margin factor

Andrew Ng

SVM Decision Boundary 当C非常大时，这部分被希望趋于0

$$\min_\theta C \underbrace{\boxed{\sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right]}}_{=0} + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

Whenever $y^{(i)} = 1$:

$\theta^T x^{(i)} \geq 1$

Whenever $y^{(i)} = 0$:

$\theta^T x^{(i)} \leq -1$

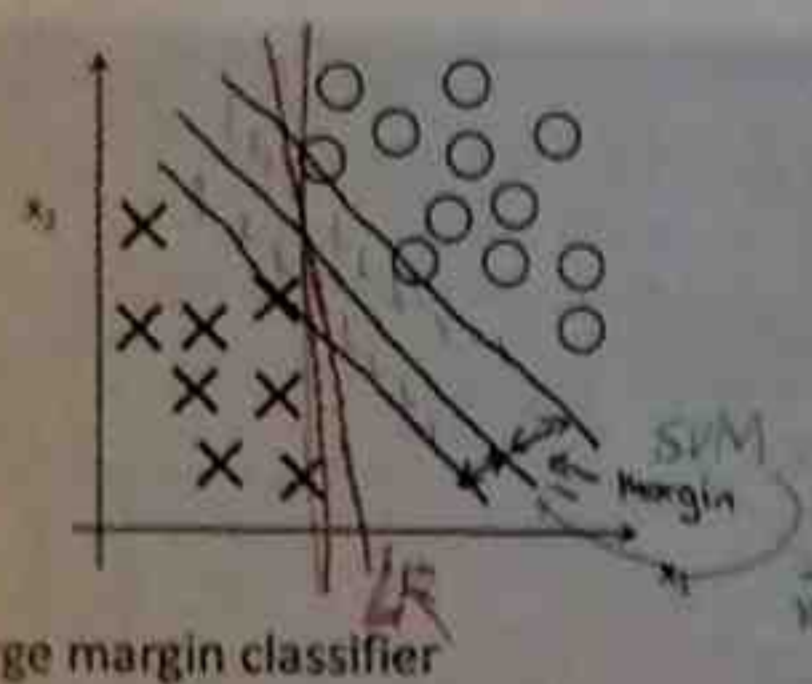
$\min_\theta \; \cancel{C \times 0} + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$

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

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

Andrew Ng

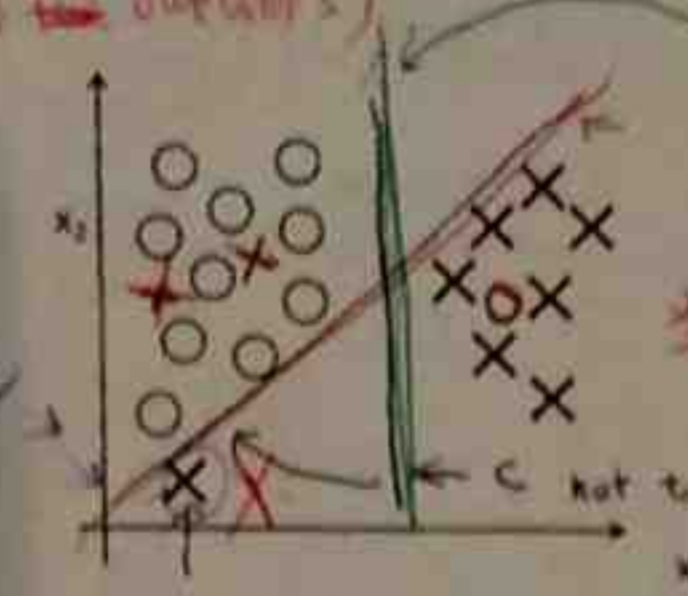
SVM Decision Boundary: Linearly separable case



Large margin classifier

Large margin classifier in presence of outliers

(sensitive to ~~the~~ outliers)

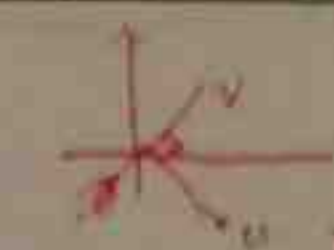


$\rightarrow$ C very large

$\frac{1}{\lambda}$

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

C大，等价于 λ小，无 regulization



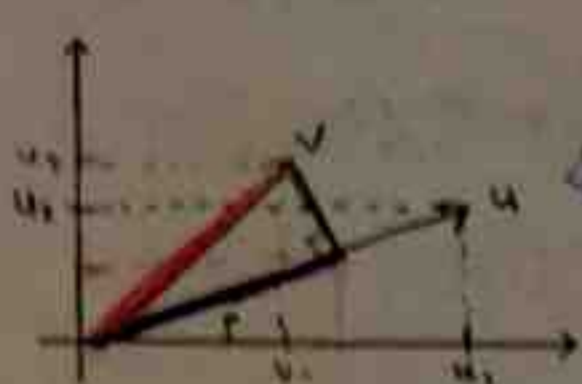
$v_1 u_1 + v_2 u_2 = 0$

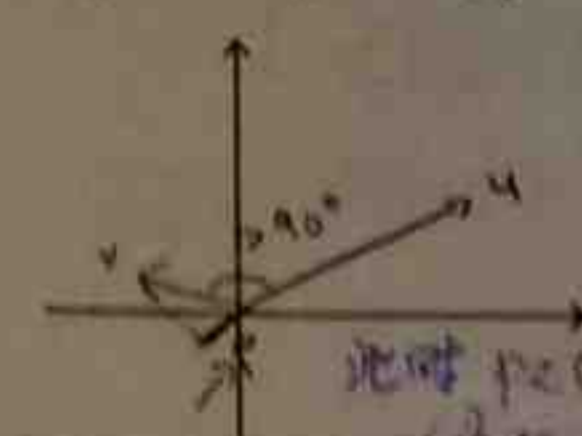
Support Vector Machines

The mathematics behind large margin classification (optional)

Machine Learning

Vector Inner Product





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

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

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

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

p = length of projection of v onto u.

signed

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

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

内积定义

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

$p < 0$

此时 $p<0$

(夹角大于90°)

3种算内积的方法

$u^T v$ $u_1 v_1 + u_2 v_2$

$v^T u$

$p \cdot \|u\|$

$\theta_0 = 0$，$n=2$ 简化到只有2个 feature

SVM Decision Boundary

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

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

$= \|\theta\|$

cost func

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

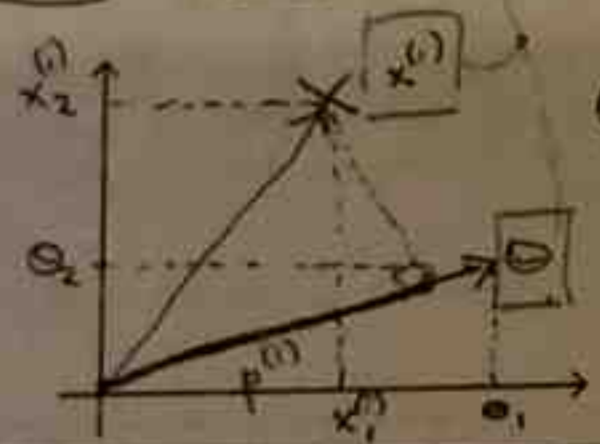
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$

Simplification: $\theta_0 = 0$ $n=2$

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

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

$u^T v$



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

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

因此 $\theta^T x^{(i)} \geq 1$

$\theta^T x^{(i)} \leq -1$

可以替换为

$p^{(i)} \cdot \|\theta\| \geq 1$

$p^{(i)} \cdot \|\theta\| \leq -1$

SVM Decision Boundary

让 $\theta_0 = 0$ 来简化 {选择 boundary，使得 θ 最小}

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

s.t. $\boxed{p^{(i)} \cdot \|\theta\| \geq 1}$ if $y^{(i)} = 1$

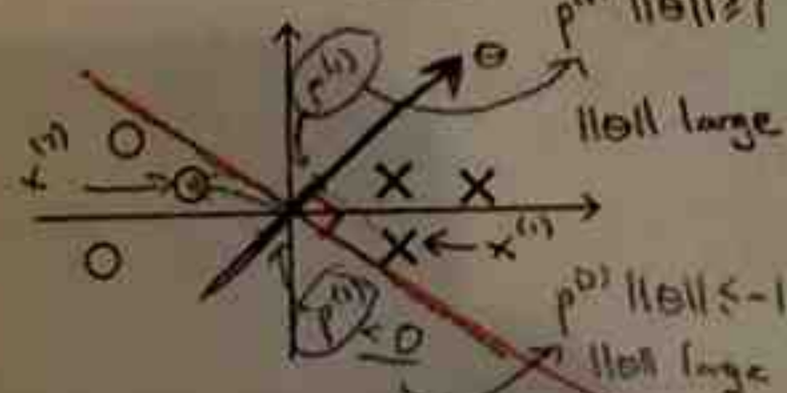
$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 1$

C very large

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

Simplification: $\theta_0 = 0$

$\theta_0 \neq 0$

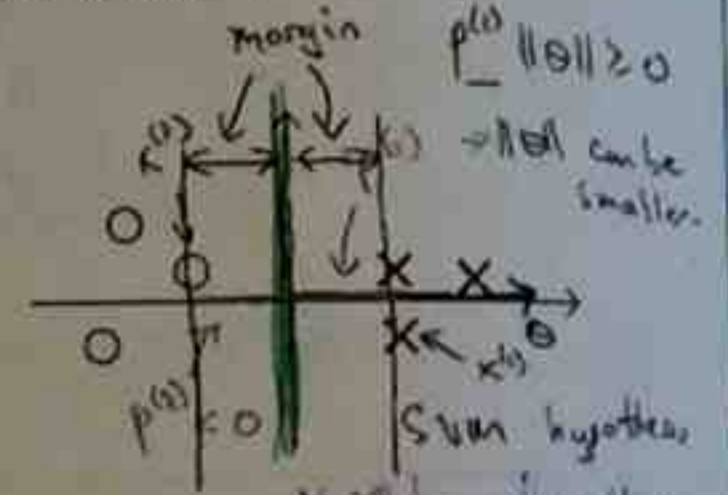


$p^{(i)}\|\theta\| \geq 1$

$\|\theta\|$ large

$p^{(i)}\|\theta\| \leq -1$

$\|\theta\|$ large



margin

$p^{(i)}\|\theta\| \geq 0$

$\to \|\theta\|$ can be smaller.

SVM hypothesis

如果 boundary 比较大，$p^{(1)}$，$p^{(2)}$ 也较大，所以容易让 $\|\theta\|$ 变小

对于 x，C 在 θ 上投影 $p^{(1)} > 0$，$p^{(2)} < 0$

都非常小，使得 $\|\theta\|$ 非常大，

但我们的 objective 是让 $\|\theta\|$ 小

不好，margin 太小

why SVM 不会选这条 boundary

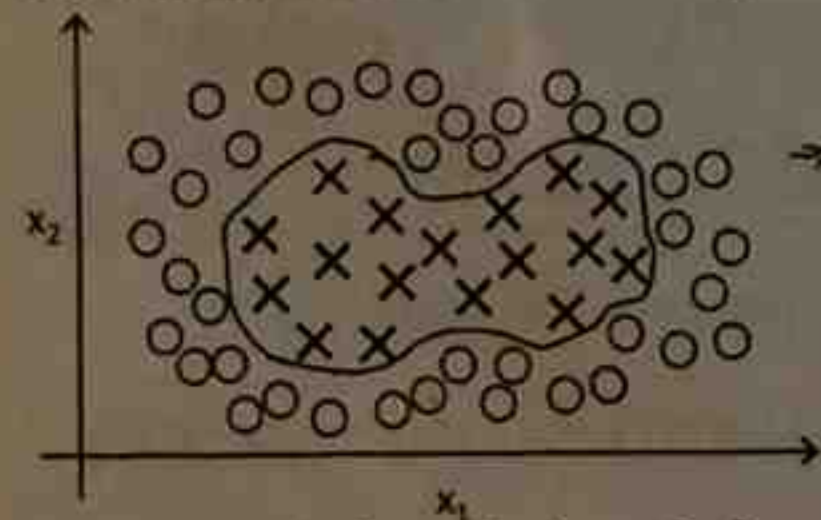
Support Vector Machines

Kernels I

Machine Learning

处理复杂的非线性分类

Non-linear Decision Boundary



Predict $y = 1$ if

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \theta_5 x_2^2 + \cdots > 0$$

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

$\to \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$

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

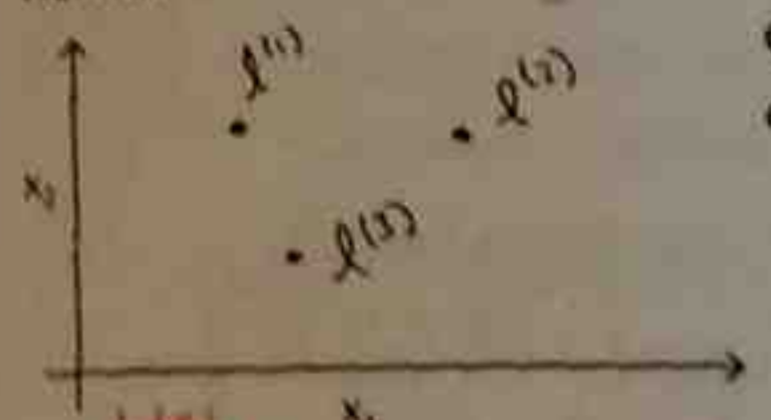
Is there a different / better choice of the features $f_1, f_2, f_3, \ldots$?

我们不知道这些 f_i 是否有用

并且 expensive

Kernel

如何定义新features



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

$\|w\|$ $w = x - l^{(1)}$ 向量长度

Given 一个样本 x:

新 new features

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

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

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

相似度函数 Kernel (Gaussian Kernels) $k(x, l^{(1)})$

这个叫 高斯 kernels）

还有其它 kernel

Kernels and Similarity

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

忽略 x_0 （总为1）

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

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

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

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

$l^{(1)} \to f_1$

$l^{(2)} \to f_2$

$l^{(3)} \to f_3$

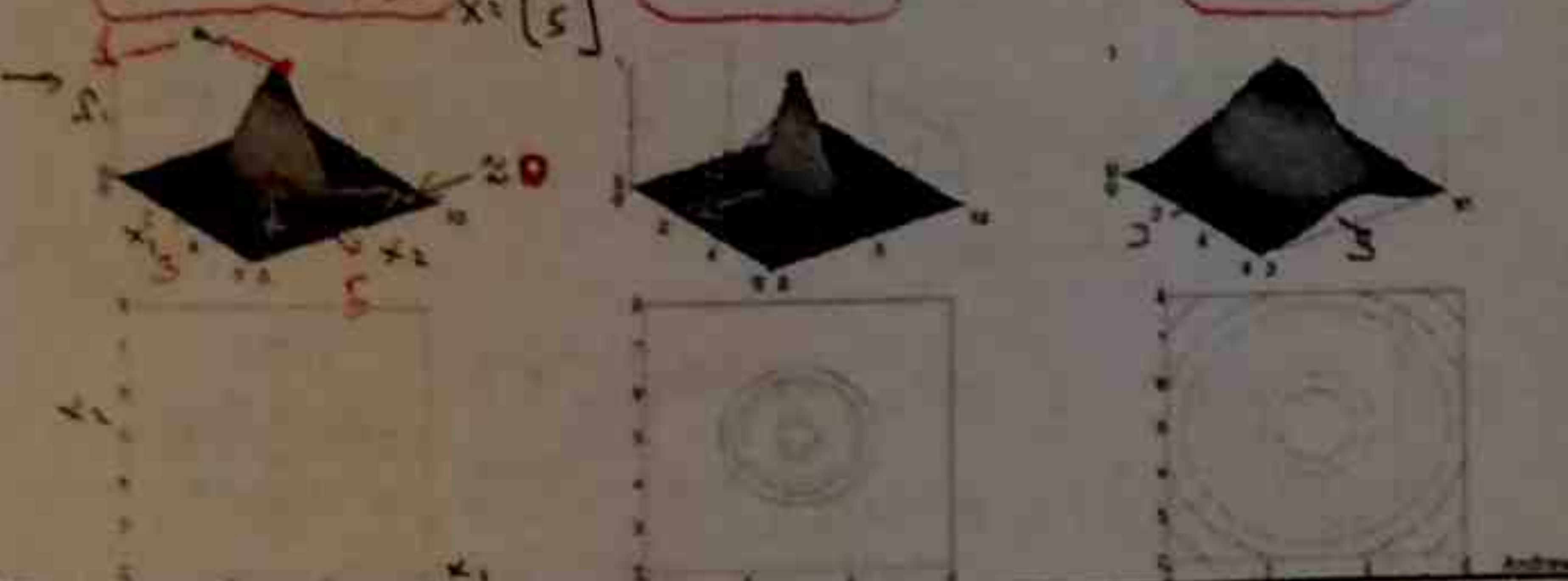
x

Example:

plot 一个新增的 feature 看起来像什么

$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$, $f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$ 2 feature x_1, x_2

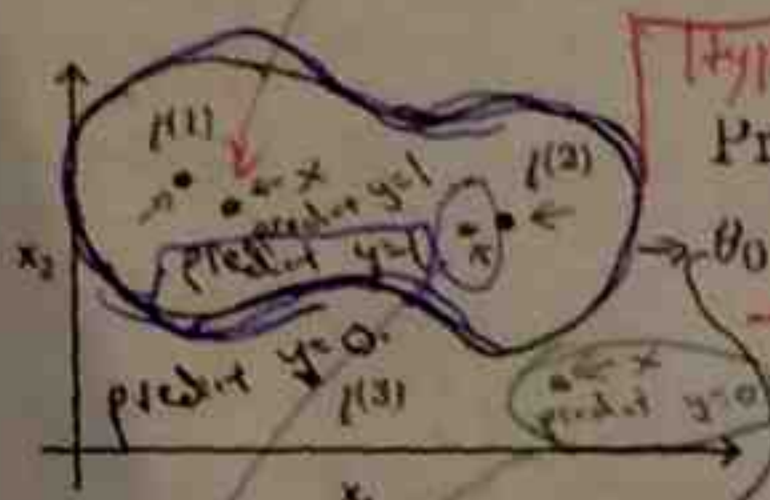
$\sigma^2 = 1$ $x = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $\sigma^2 = 0.5$ $\sigma^2 = 3$



$\sigma^2 = 0.5$ kernel 下降到0更快

$\sigma^2 = 3$ kernel 下降到0更慢

如果有一个 example 位于这里，会怎样



Hypothesis function

Predict "1" when

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

例如 训练出了

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

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

$\to \theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$ （代入 x）

$= -0.5 + 1 = 0.5 \ge 0$ （代入 θ） 得 $0.5 \ge 0$ predict 1.

另一 example predict $y=0$

$f_1, f_2, f_3 \approx 0$

$\to \theta_0 + \theta_1 \cdot 0 + \dots = -0.5 < 0$

第3个 example，可以预测 $y=1$，因为发现 接近 $l^{(1)}, l^{(2)}$ 的都是 $y=1$；远离的都是 $y=0$

推导到更复杂的非线性分类 $\Leftarrow$ 通过 landmark 找到 $y=1$ 的 boundary.

① 找 landmark.
② 用 GK 找 …
来 build $h_\theta(x)$

⇒ How to find landmarks
放在 m个 training examples 上
$l^{(1)}$ $l^{(2)}$ … $l^{(m)}$
m个 examples ⇒ m个 landmark

对于任意一个 … 计算 … (下面的)

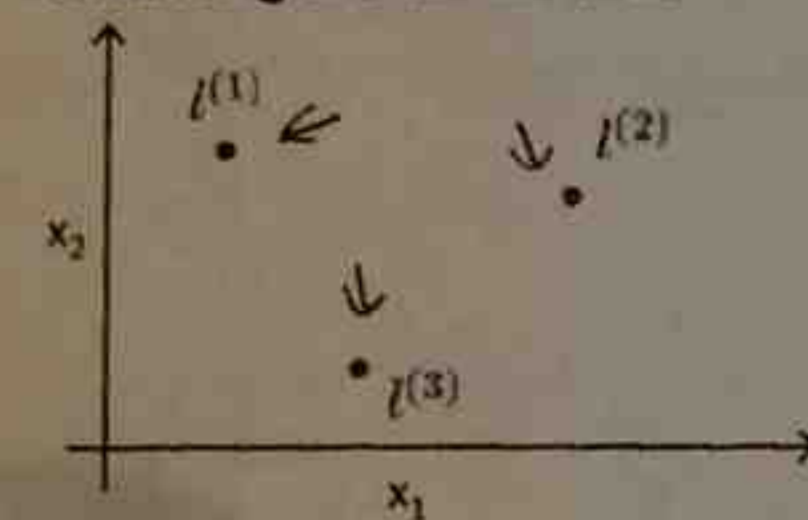
Support Vector Machines

Kernels II

Machine Learning

① detail
② how to use
③ bias-variance problems

Choosing the landmarks



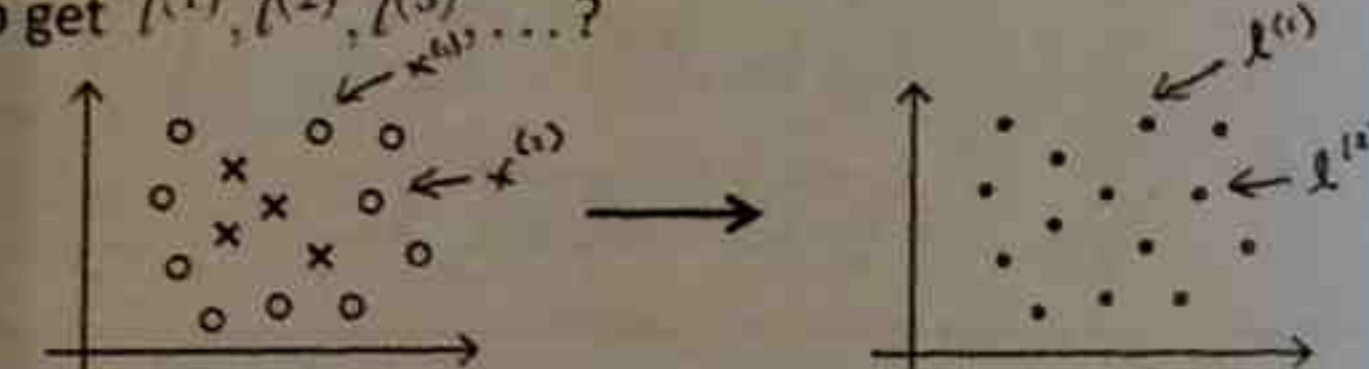
Given x:

$\rightarrow f_i = \text{similarity}(x, l^{(i)})$

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

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

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



$l^{(1)}$
$l^{(2)}$
$\vdots$
$l^{(m)}$

SVM with Kernels

(所有 examples) 变 m个 landmark

→ 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:

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

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

…

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

Train / Cross Val / Test set

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

$$x^{(i)} \rightarrow \begin{array}{l} f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)}) \\ \vdots \\ f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp(-\frac{0}{2\sigma^2}) = 1 \\ \vdots \\ f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)}) \end{array}$$

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

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

其中有一个 landmark 就是 example 自己

用 feature vector 来表示 training example

SVM with Kernels

用x与landmark相似度算一个 feature

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

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

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

n=m

Training:

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

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

原来的 cost function → θ_0

不用 $\theta^T x^{(i)}$ 改用 $\theta^T f^{(i)}$

$- \sum_j \theta_j^2 = \theta^T \theta \leftarrow \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_m \end{bmatrix}$ (ignore θ_0)

$\|\theta\|^2$

$m = 10{,}000$

$- \rightarrow \underbrace{\theta^T M \theta}_{\text{Matrix}}$

很多SVM会换一下 $\theta^T M \theta$
虽然目标都是 minimize $\|\theta\|$ 实际上 $m = n$.
会让 svm 更有效

Kernel 用在 LR上会很慢.

如何选参数来平衡 bias/variance

SVM parameters:

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

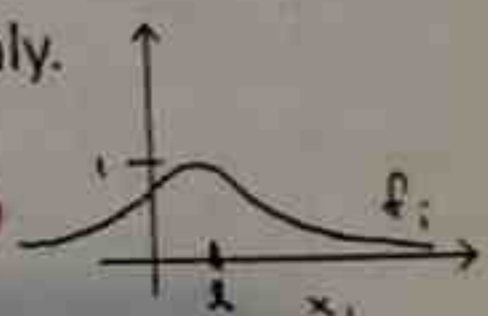
→ Small C: Higher bias, low variance. (large λ)

Large σ^2: Features f_i vary more smoothly.

→ Higher bias, lower variance.

$\exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)$ σ大时，⇒

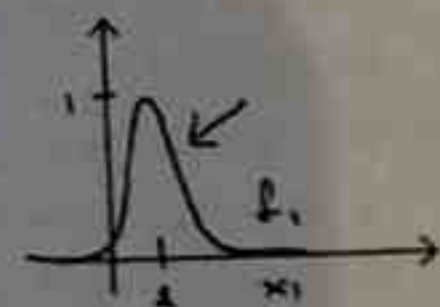
容易欠拟合 不容易过拟合



Small σ^2: Features f_i vary less smoothly.

Lower bias, higher variance.

容易过拟合 不容易欠拟合



Andrew Ng

Machine Learning

Support Vector Machines

Using an SVM

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters θ. 那怎么用呢，也都要

Need to specify:

→ Choice of parameter C.

Choice of kernel (similarity function):

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

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

$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \geq 0$ $x \in \mathbb{R}^{n+1}$

→ n large, m small

→选择1：不用kernel，拟合线性的边界 当feature很多(n大) example很少(m小)时候用

→ Gaussian kernel:

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

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

Need to choose σ^2.

feature少，样本多时，

用kernel来拟合一个复杂的非线性模型

Andrew Ng

有些要自己实现一个kernel

Kernel (similarity) functions: $x^{(i)}$ $l^{(j)} = x^{(j)}$

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

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

```
return
```

例如高斯 kernel

$x \to f_1, f_2, \ldots, f_m$

→ Note: Do perform feature scaling before using the Gaussian kernel. $x \in \mathbb{R}^n$

$\|x - l\|^2$ $v = x - l$

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

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

1000 feet² 1-5 bedrooms

Andrew Ng

feature scaling 是有必要的

要归一到同一个区间段

Other choices of kernel

: Linear kernel, meaning no kernel, 多为此 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).

访数学方法用的太多未学, 不会还未讲述

Many off-the-shelf kernels available:

- Polynomial kernel: $k(x, l) = (x^T l)^2$, $(x^T l)^3$, $(x^T l + 1)^3$, $(x^T l + 5)^4$, $(x^T l + \text{constant})^{\text{degree}}$

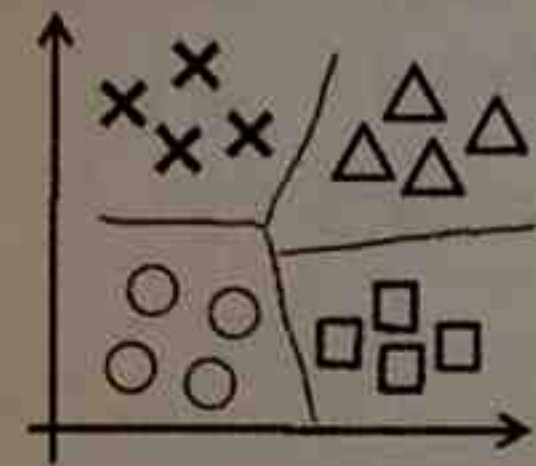
  通常比高斯 kernel 差, X 与 L 都 negative

- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, ... $\text{sim}(x, l)$

  其他 kernel; 估有会用这一个

Andrew Ng

Multi-class classification



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

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 $i = 1, 2, \ldots, K$), get $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}$
Pick class i with largest $(\theta^{(i)})^T x$

$y=1$ $y=2$ ... $\theta=K$

Andrew Ng

用 K 个 model 全都预测一下
分到得分最高的那个类别中

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. $n \geq m$, $n = 10{,}000$, $m = 10$ 至 1000)
→ Use logistic regression, or SVM without a kernel ("linear kernel")

} 特征够多了, 用 LR 就可以

→ If n is small, m is intermediate: ($n = 1$-1000, $m = 10$-$10{,}000$)
→ Use SVM with Gaussian kernel

} 特征少, 样本也量也不是太多

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

} 样本量太大了, 还是加特征采用 LR 释放的问题

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

神经网络都不错
只是训练比较慢

Andrew Ng

LR 与 SVM-NO-KERNEL 很相似

SVM : Convex optimization ✗ worry for local optimization problem
Natural Network : 一般也不用担心 local optimization problem
但还是比较慢