Week 07

- Large Margin Classification
- Kernels
- SVMs in Practice

I. Large margin classification

1. Optimization objectives

Alternative view of logistic regression

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} = g(z)$; __/‾‾

if $y = 1; \Rightarrow h_\theta(x) \approx 1$ when $\theta^T x >> 0$

if $y = 0; \Rightarrow h_\theta(x) \approx 0$ when $\theta^T x << 0$

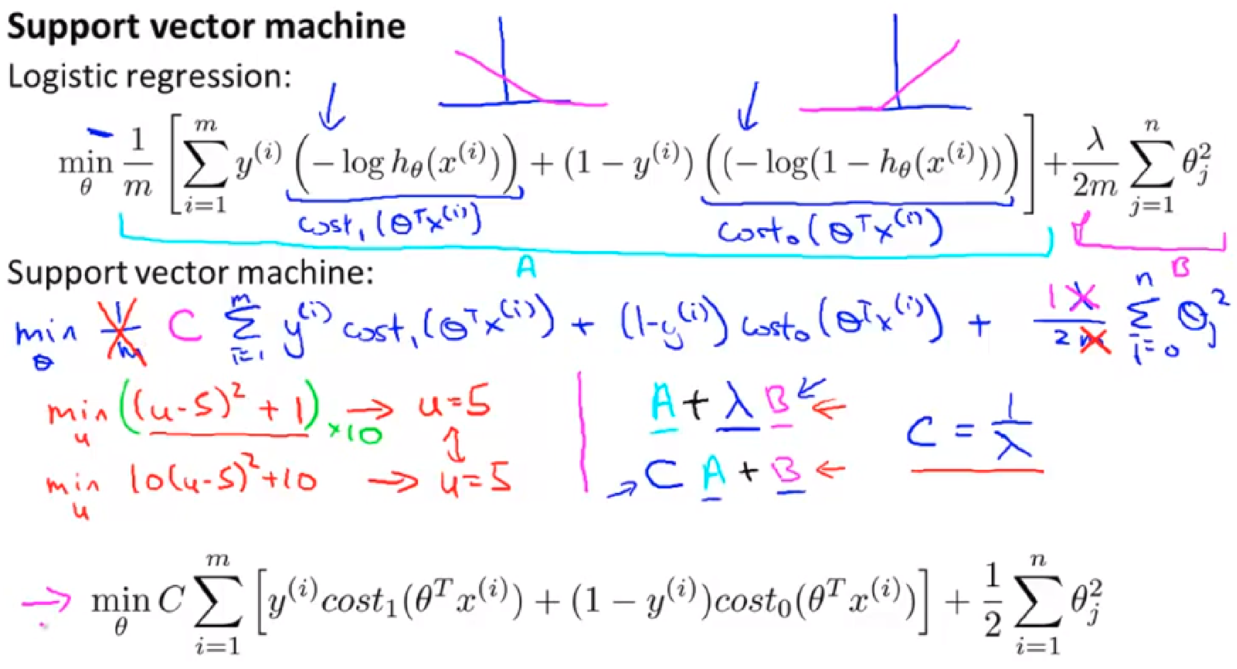
Cost of $-y\ log(h_\theta(x)) + (1-y)\ log(1 - h_\theta(x))$

=

Cost of $-y\ log(\frac{1}{1+e^{-\theta^T x}}) + (1-y)\ log(1 - \frac{1}{1+e^{-\theta^T x}})$

$y = 1$ then \__

$y = 0$ then __/



Support vector machine

Logistic regression:

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

Support vector machine:

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

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

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

$$A + \lambda B \leftarrow$$

$$C A + B \leftarrow$$

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

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

SVM hypothesis

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

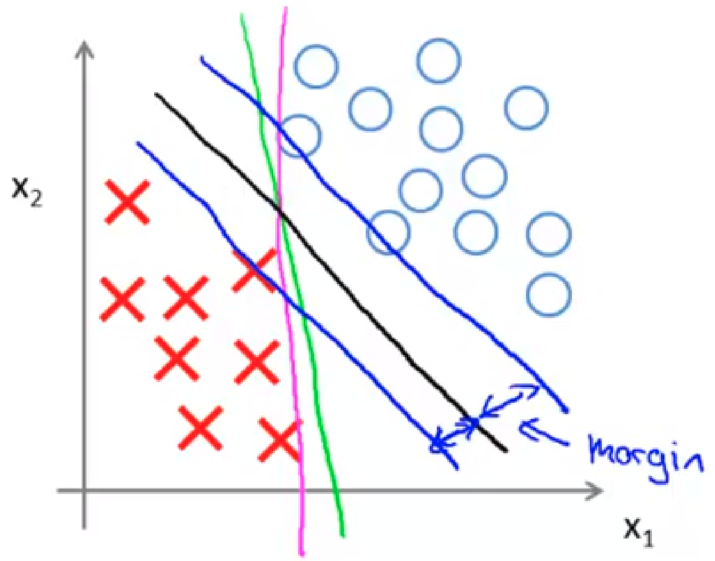
Hypothesis:

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

2. Large Margin intuition

It is about C value;

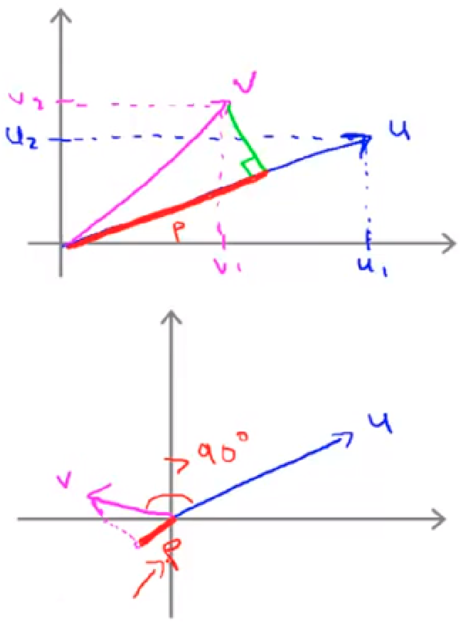
SVM Decision Boundary: Linearly separable case



Large margin classifier

3. Math behind Large Margin intuition

Vector Inner Product



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

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

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

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

p = length of projection of v onto u.

Signed

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

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

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

$p < 0$

Andrew Ng

SVM Decision Boundary

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

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

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

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

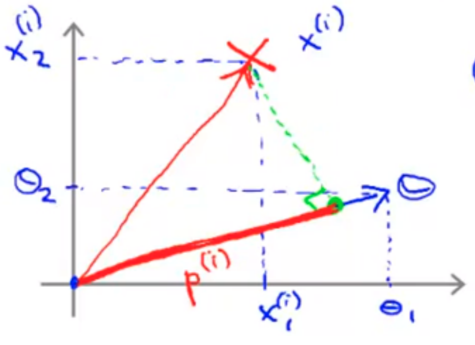
Simplication: $\underline{\theta_0 = 0}$. $\quad \underline{n=2}$

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

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

$\uparrow \;\; \uparrow$

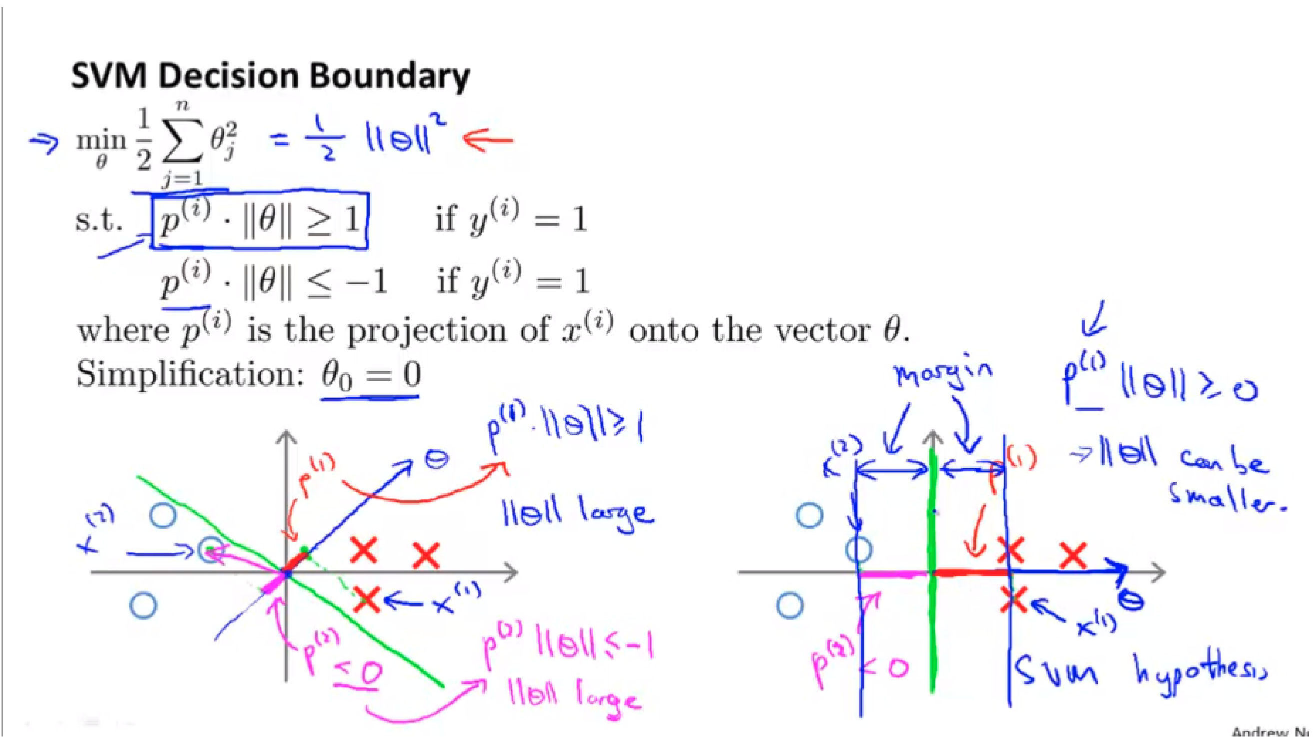
$u^T v$



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

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

Andrew Ng



SVM Decision Boundary

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

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

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

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

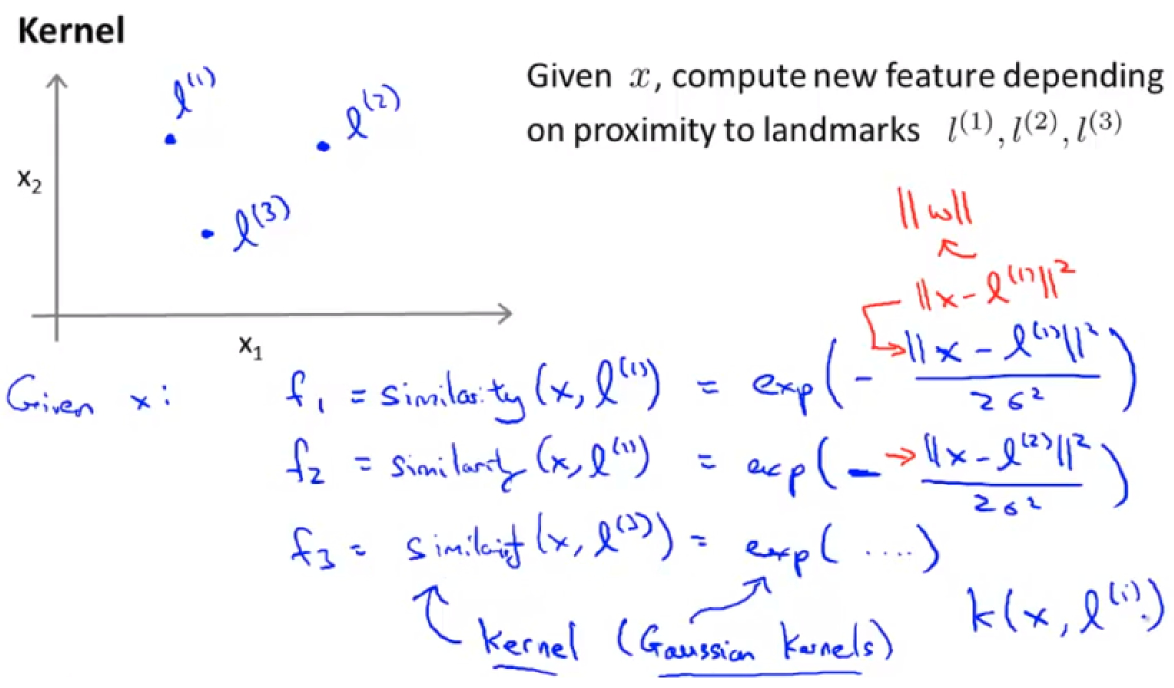
Simplification: $\theta_0 = 0$

II. Kernels

Non-linear decision boundary:

- $h_\theta(x) = 1$ if polynomial $\theta_0 + \theta_1 x_1 + .. \geq 0$
- which can be written as $\theta_0 + \theta_1 f_1 + ..$ where $f_1 = x_1, ...$

  $\Rightarrow$ Question: is a better or different choice of features $f_1, f_2, ..$



Kernel

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

Given x:

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

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

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

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

Kernels and Similarity

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

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

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

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

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

$l^{(3)} \rightarrow f_3$.

$\vdots$ $\quad\uparrow$

x

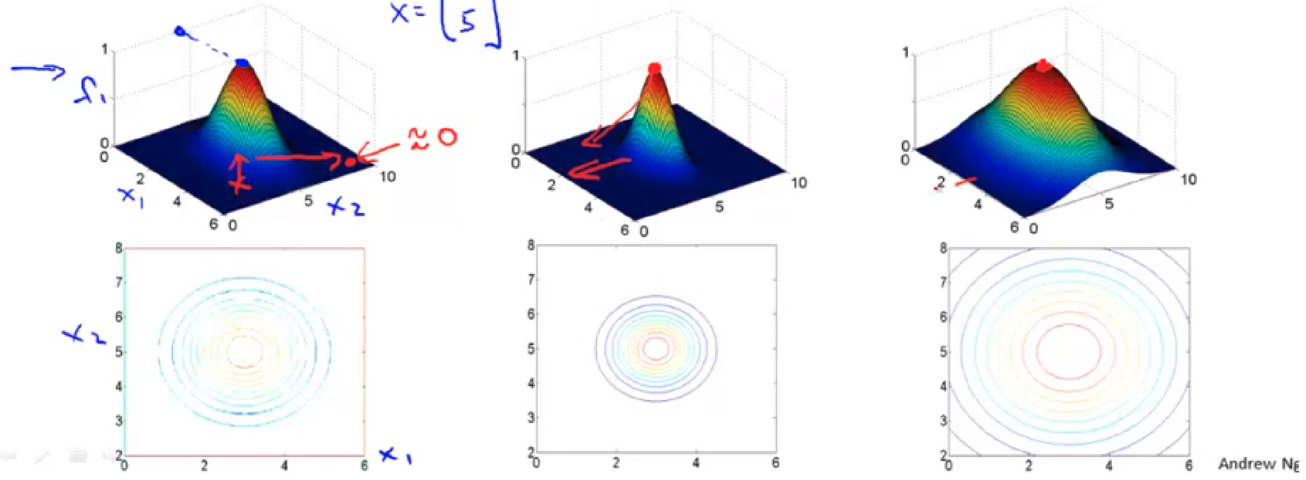
If $\underline{x}$ if far from $\underline{l^{(1)}}$:

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

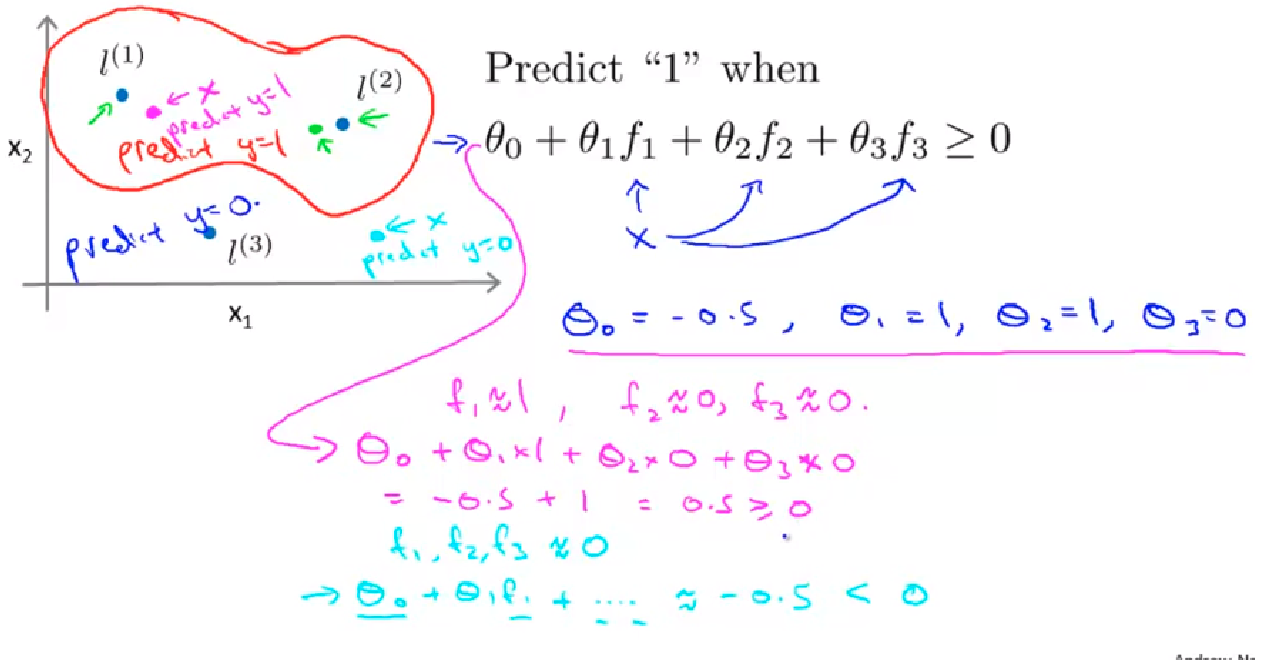
Example:

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

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



Andrew Ng



Predict "1" when

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

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

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

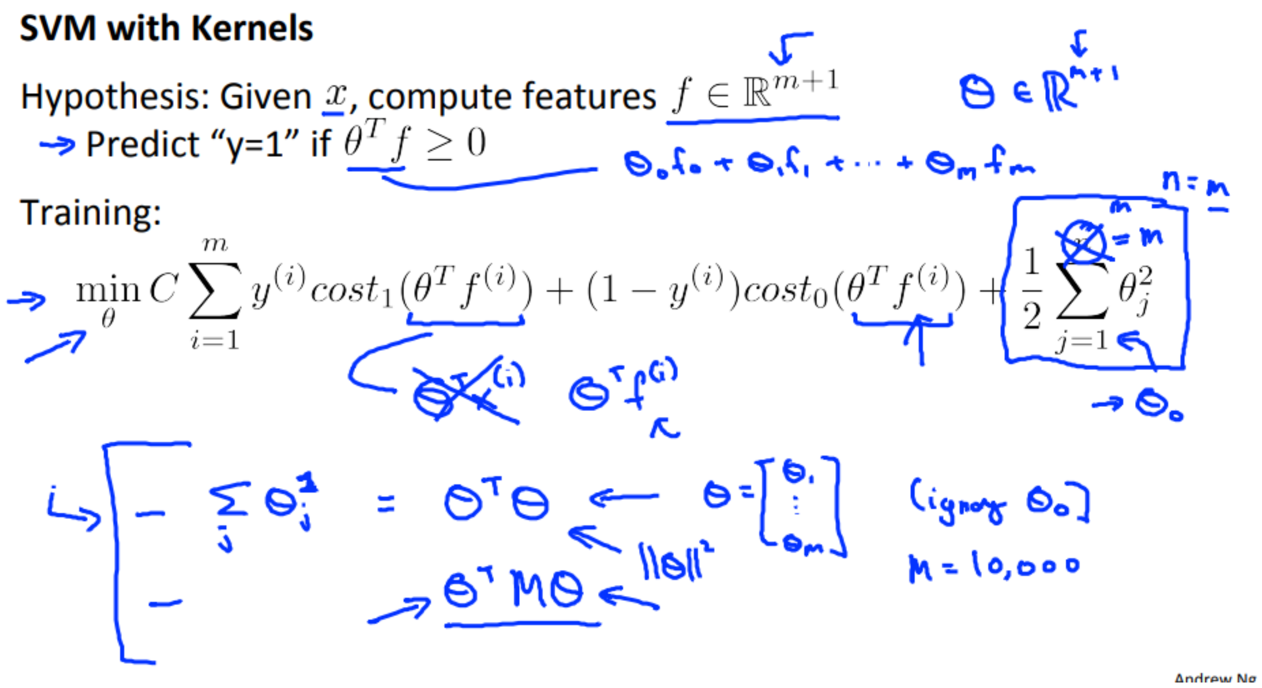
$\theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$

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

$f_1, f_2, f_3 \approx 0$

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

We can learn pretty complex non-linear decision boundary, like what is just drew up where we predict positive when we're close to either one of the two landmarks. And we predict negative when we're very far away from any of the landmarks. And so this is part of the idea of kernels of and how we use them with the support vector machine, which is that we define these extra features using landmarks and similarity functions to learn more complex nonlinear classifiers.



SVM with Kernels

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

Predict "y=1" if $\theta^T f \ge 0$ $\quad \theta_0 f_0 + \theta_1 f_1 + \dots + \theta_m f_m$

Training: $n = m$

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

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

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

$\|\theta\|^2$

$\theta^T M \theta$

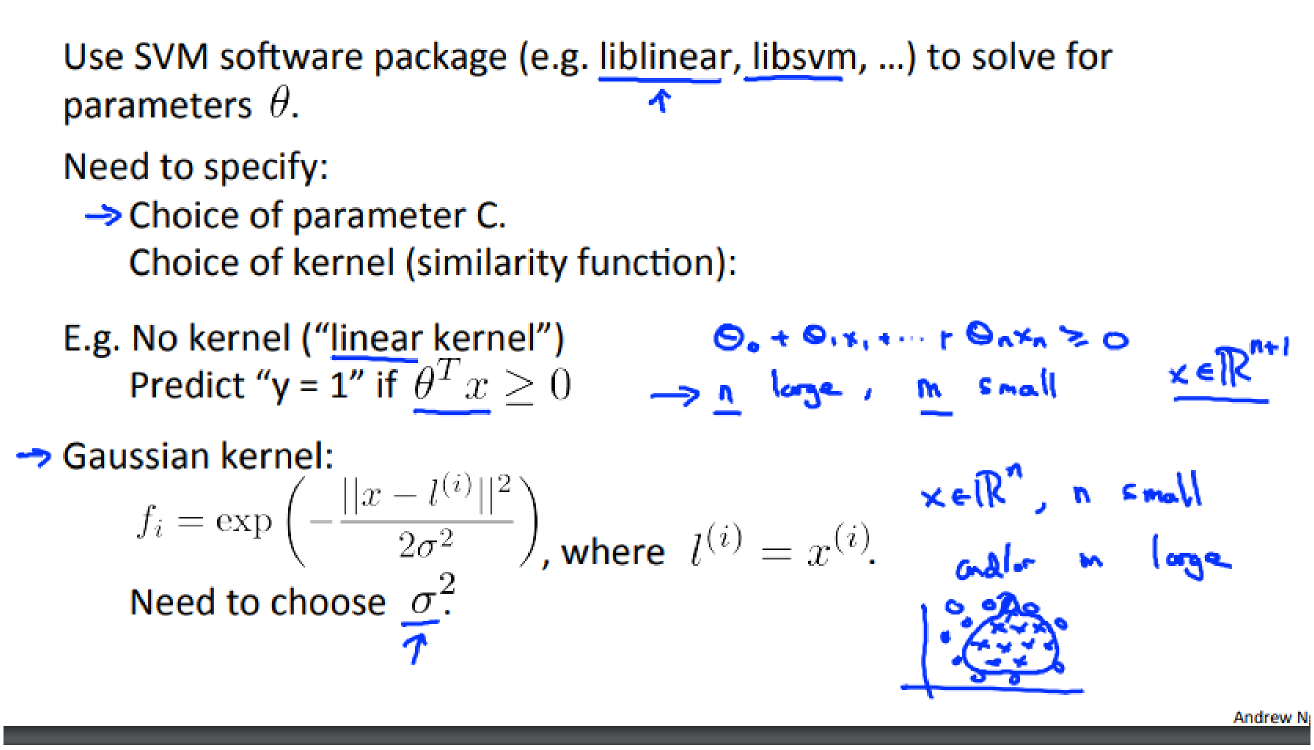
$M = 10{,}000$

Change x by f.

- SVM parameters:
    - C
        - Large: Lower bias, high variance. small λ

        - Small: High bias, low variance. large λ
- σ^2
    - Large: Features f_i vary more smoothly + Higher bias, lower variance. --^--
    - Small: Features f_i vary less smoothly + lower bias, high variance. _/_

III. SVMs in practice



Use SVM software package (e.g. $\underset{\uparrow}{\underline{\text{liblinear}}}$, $\underline{\text{libsvm}}$, ...) to solve for parameters θ.

Need to specify:

$\rightarrow$ Choice of parameter C.

Choice of kernel (similarity function):

E.g. No kernel ("$\underline{\text{linear}}$ kernel")

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

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

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

$\rightarrow$ Gaussian kernel:

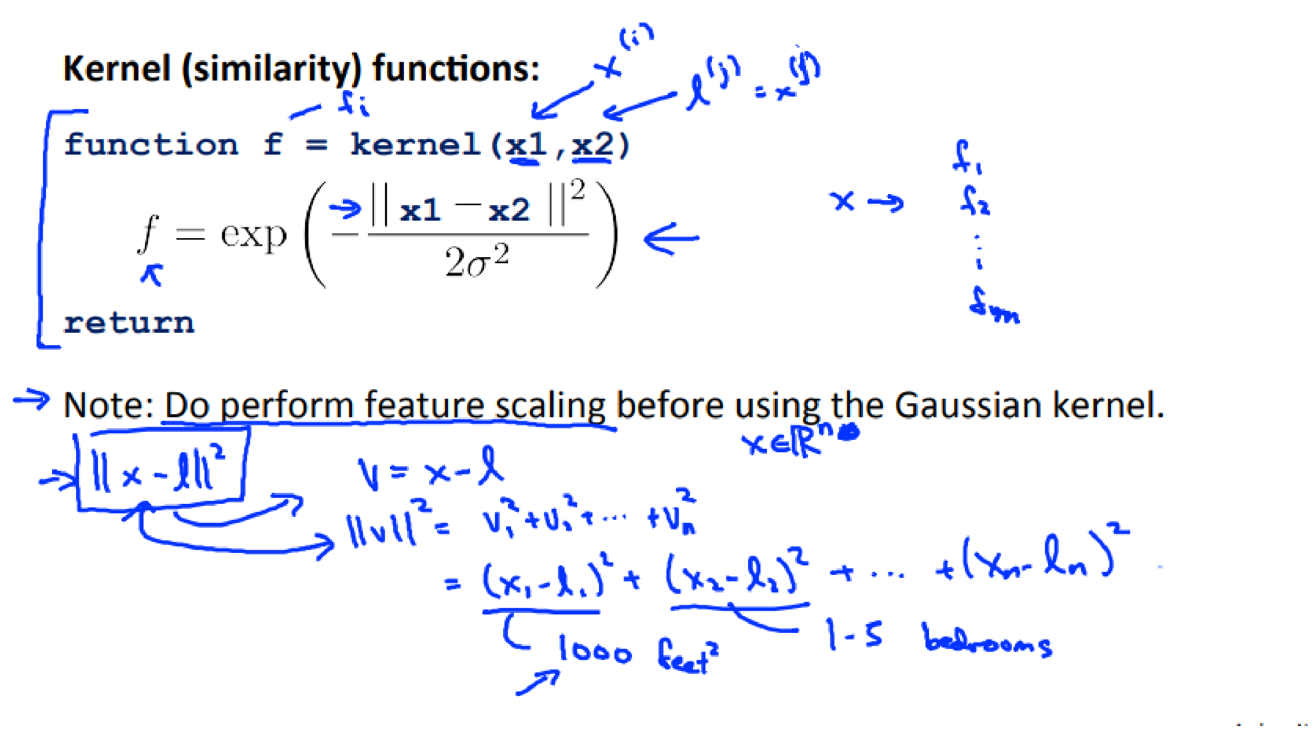
$$f_i = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right), \text{ where } l^{(i)} = x^{(i)}.$$

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

and/or m large

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

Andrew Ng



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

f_i

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

    f = exp( -||x1 - x2||^2 / (2σ^2) )  ←

return
```

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

$\rightarrow$ Note: $\underline{\text{Do perform feature scaling}}$ before using the Gaussian kernel.

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

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

$v = x - l$

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

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

1000 feet^2 $\qquad$ 1-5 bedrooms

Other choices of kernel

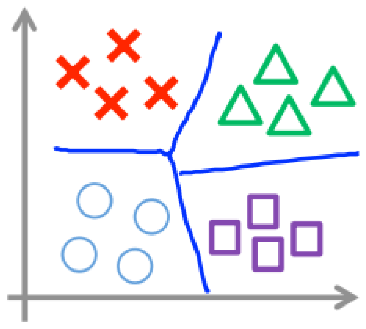
Note: Not all similarity functions $\mathrm{similarity}(x, l)$ make valid kernels.
→ (Need to satisfy technical condition called "Mercer's Theorem" to make sure SVM packages' optimizations run correctly, and do not diverge).

Many off-the-shelf kernels available:

- Polynomial kernel: $k(x, l) = (x^T l)^2$, $(x^T l)^3$, $(x^T l + 1)^3$, $(x^T l + 5)^4$ $(x^T l + \text{constant})^{\text{degree}}$
- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, … $\mathrm{sim}(x, l)$

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

$y=1$ $y=2$ …. $y=K$

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.

Andrew Ng