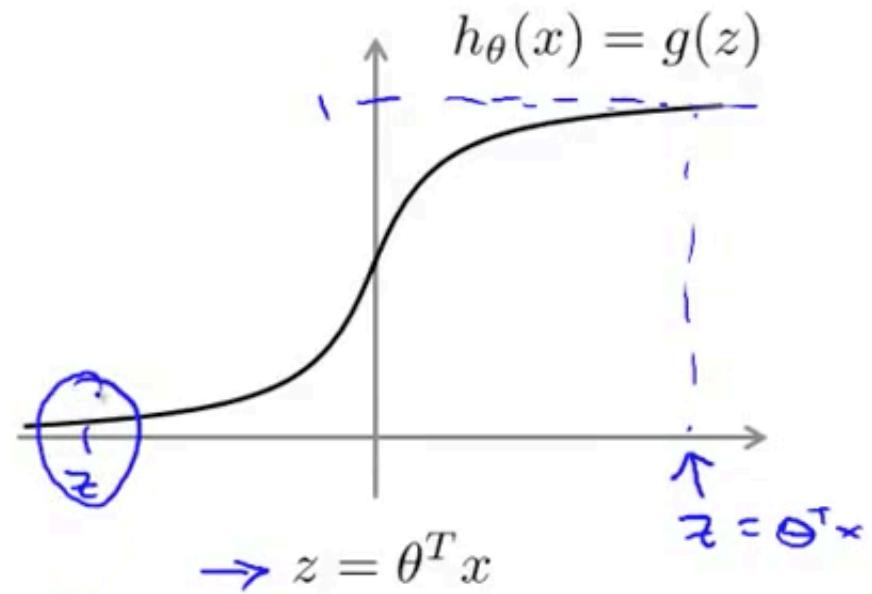


Alternative view of logistic regression

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



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

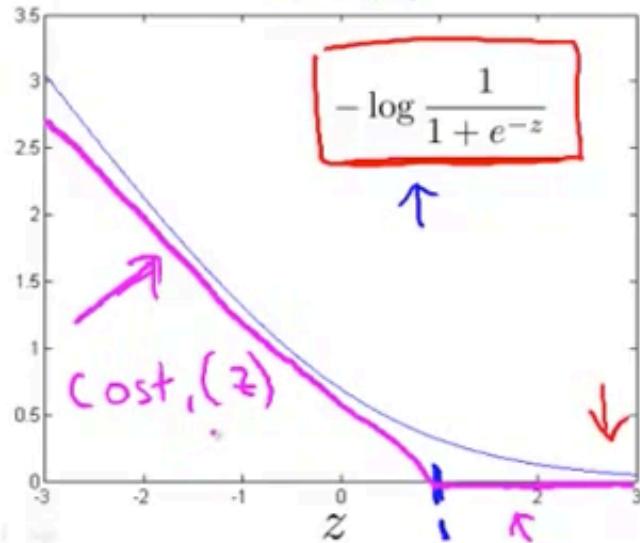
If $y = 0$, we want $h_{\theta}(x) \approx 0$, $\theta^T x \ll 0$

Alternative view of logistic regression

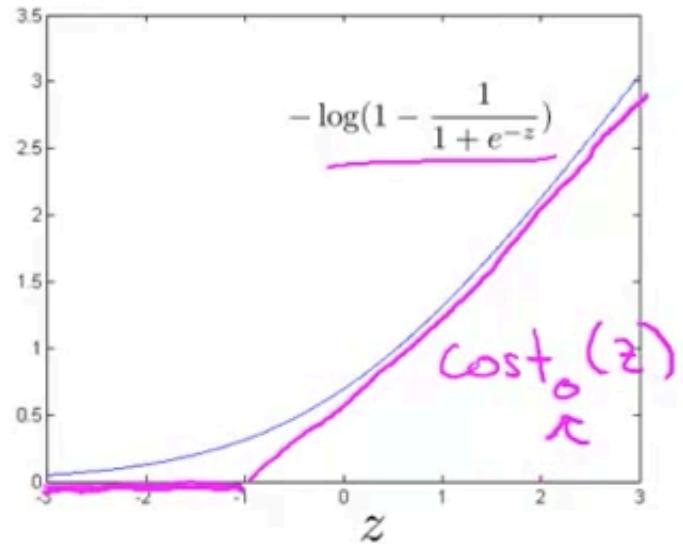
$$\begin{aligned}
 \text{Cost of example: } & -(y \log h_{\theta}(x) + (1 - y) \log(1 - h_{\theta}(x))) \leftarrow \\
 & = \boxed{-y \log \frac{1}{1 + e^{-\theta^T x}}} - \boxed{(1 - y) \log(1 - \frac{1}{1 + e^{-\theta^T x}})} \leftarrow
 \end{aligned}$$

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

$$z = \Theta^T x$$



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



Support vector machine

Logistic regression:

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

Support vector machine:

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

$$\min_u \frac{(u - \xi)^2 + 1}{2} \rightarrow u = \xi$$

$$\min_u \log(u - \xi)^2 + 1 \rightarrow u = \xi$$

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

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

SVM hypothesis

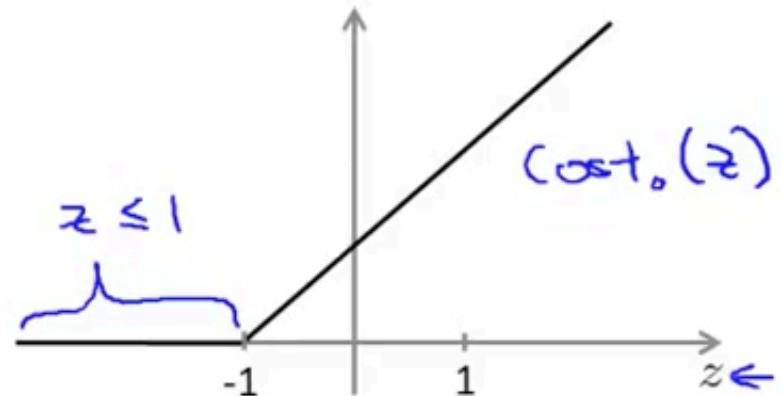
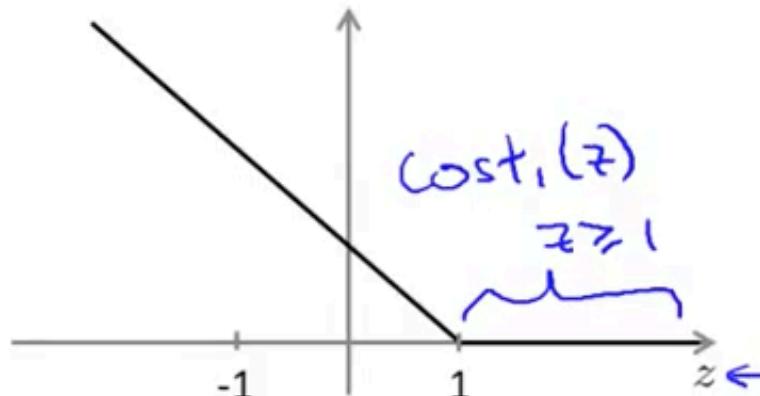
$$\Rightarrow \min_{\theta} C \sum_{i=1}^m \left[y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

Hypothesis:

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

Support Vector Machine

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



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

$$\theta^T x \geq \cancel{0} 1$$

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

$$\theta^T x \leq \cancel{0} -1$$

SVM Decision Boundary

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

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

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

$$\min_{\theta} \cancel{C \sum_{i=1}^m} + \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

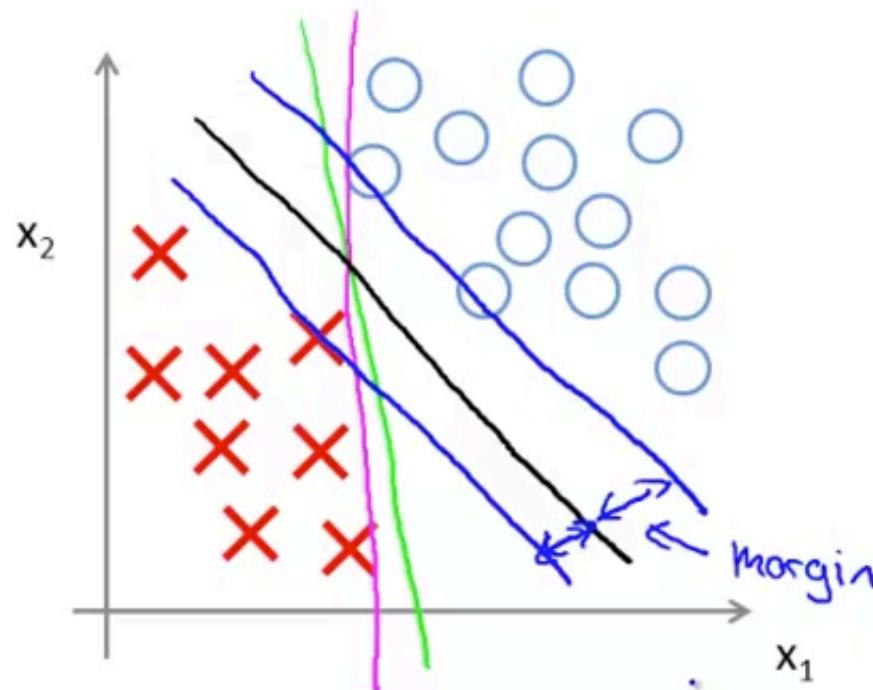
$$\text{s.t. } \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$

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

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

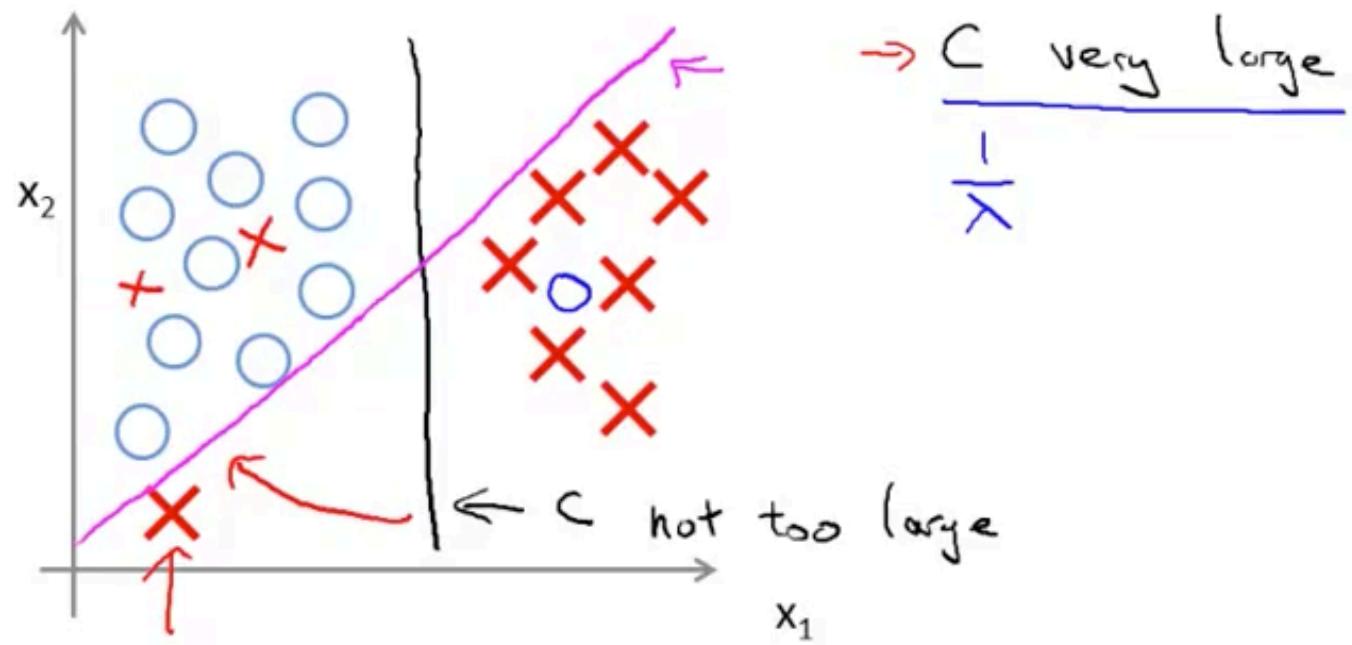
$$\theta^T x^{(i)} \leq -1$$

SVM Decision Boundary: Linearly separable case

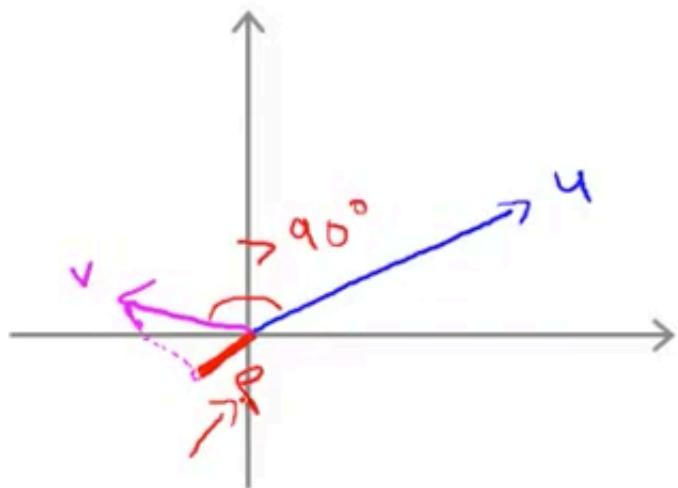
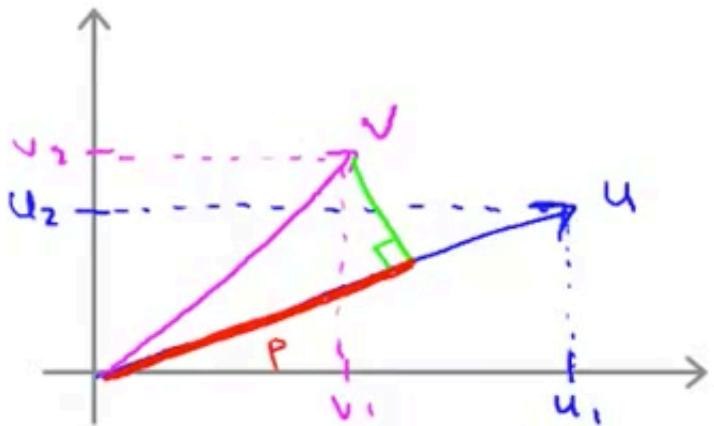


Large margin classifier

Large margin classifier in presence of outliers



Vector Inner Product



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

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

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

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

\nearrow Signed

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

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

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

$$p < 0$$

SVM Decision Boundary

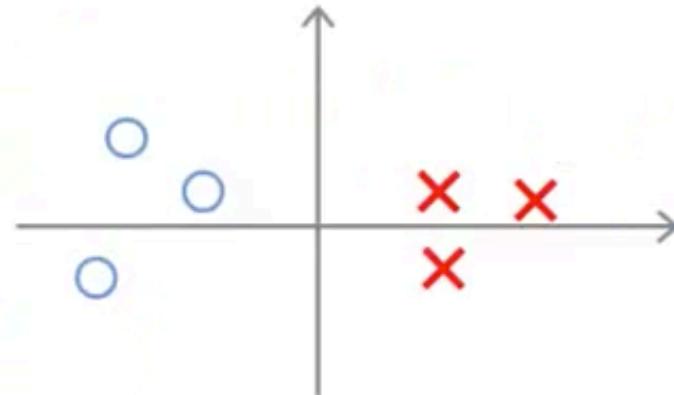
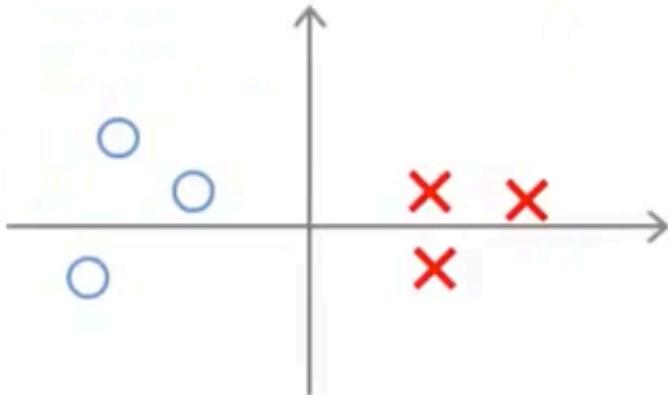
$$\min_{\theta} \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

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

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

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

Simplification: $\theta_0 = 0$



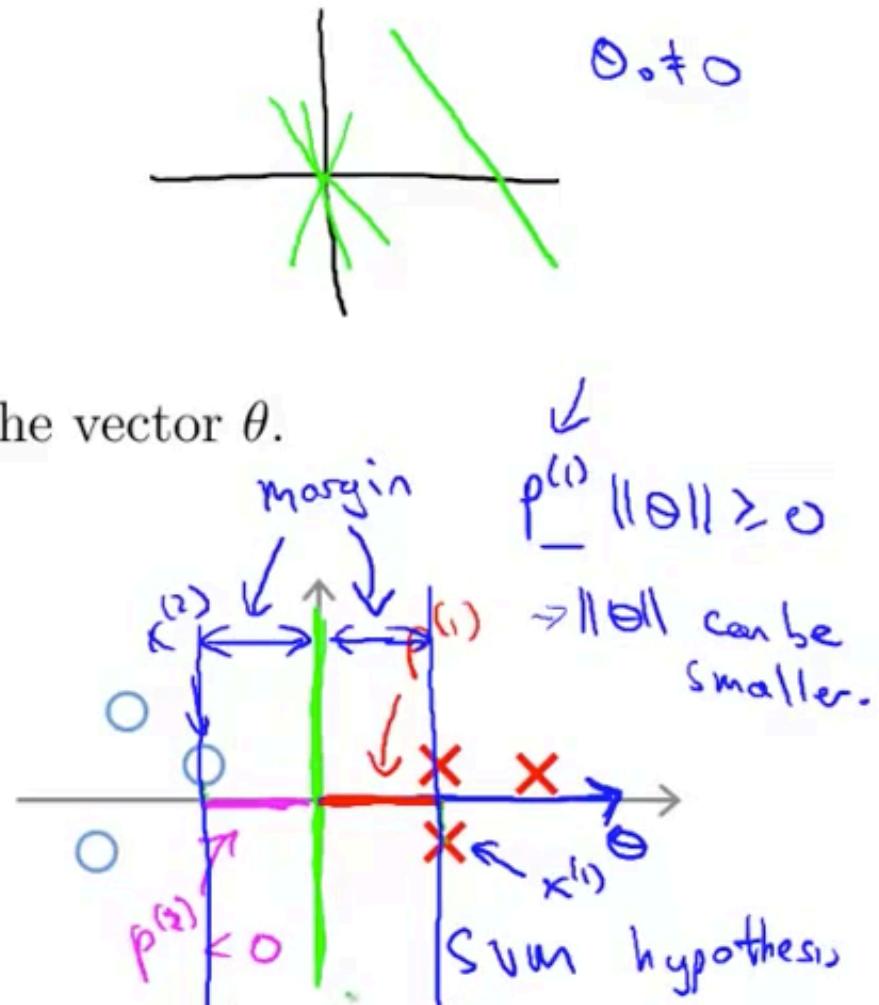
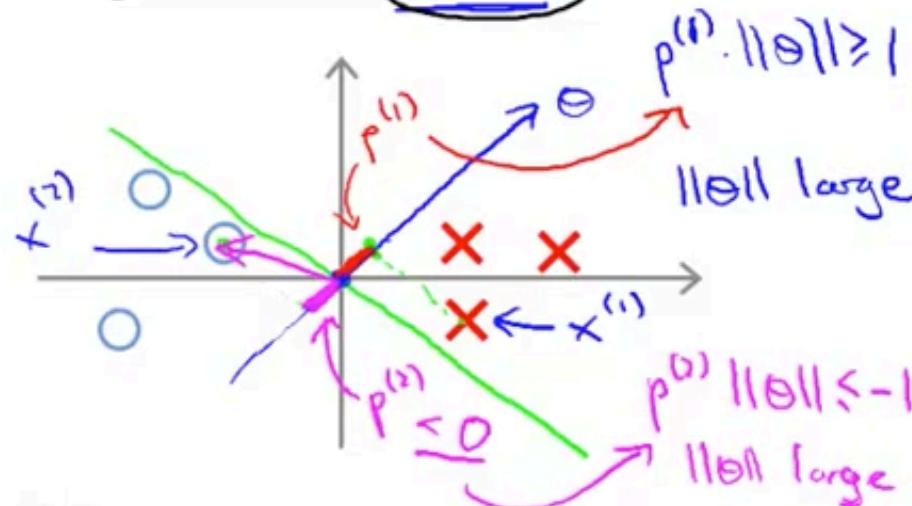
SVM Decision Boundary

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

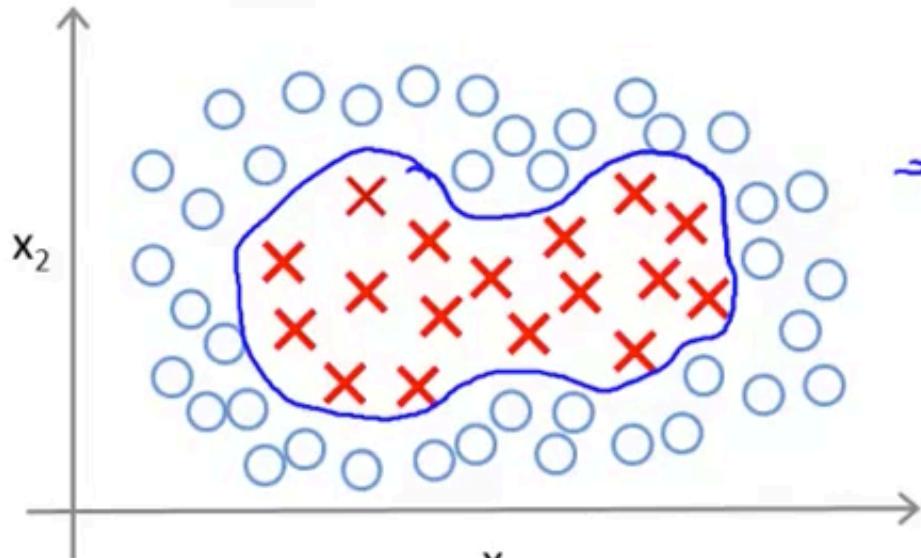
s.t. $p^{(i)} \cdot \|\theta\| \geq 1 \quad \text{if } y^{(i)} = 1$
 $p^{(i)} \cdot \|\theta\| \leq -1 \quad \text{if } y^{(i)} = -1$

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

Simplification: $\theta_0 = 0 \leftarrow$



Non-linear Decision Boundary



Predict $y = 1$ if

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

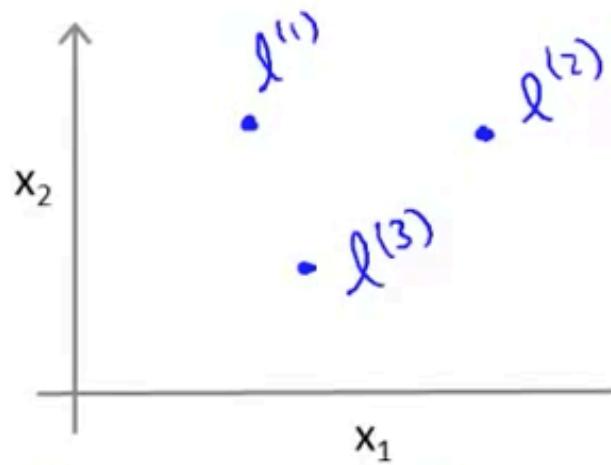
$$h_{\theta}(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \dots \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

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

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

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 :

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

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

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

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

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

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

$$\begin{aligned} l^{(1)} &\rightarrow f_1 \\ l^{(2)} &\rightarrow f_2 \\ l^{(3)} &\rightarrow f_3 \\ &\uparrow \\ &\uparrow \\ x & \end{aligned}$$

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

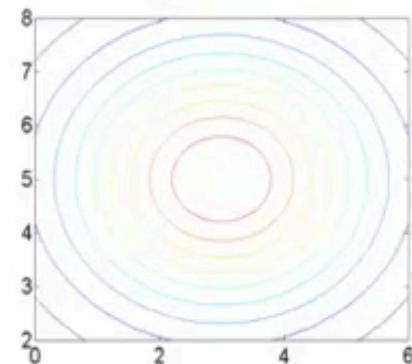
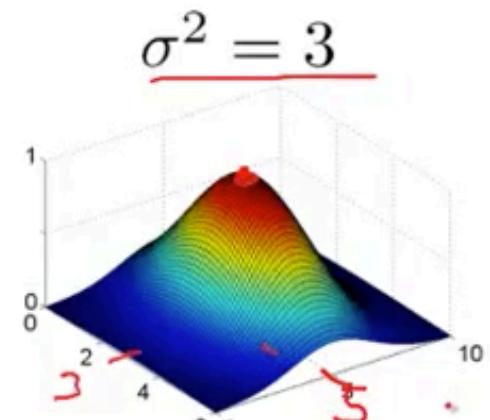
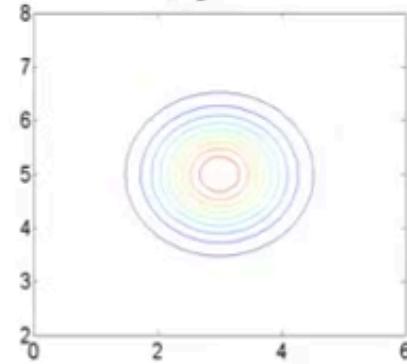
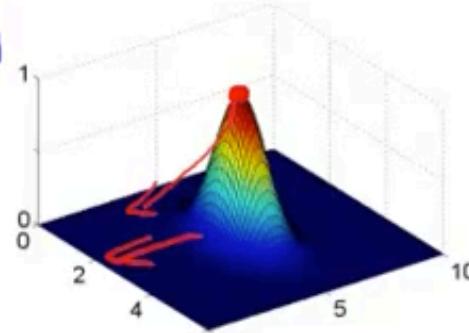
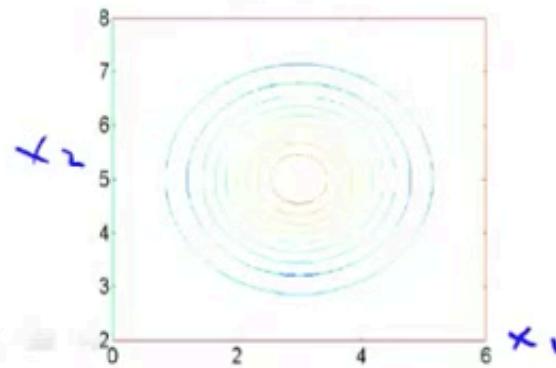
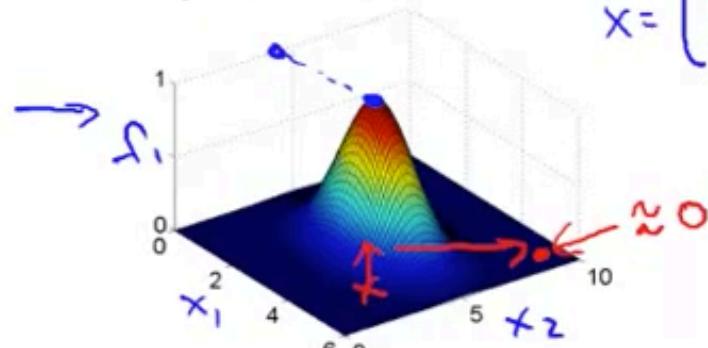
$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \underset{\uparrow}{\approx} 0.$$

Example:

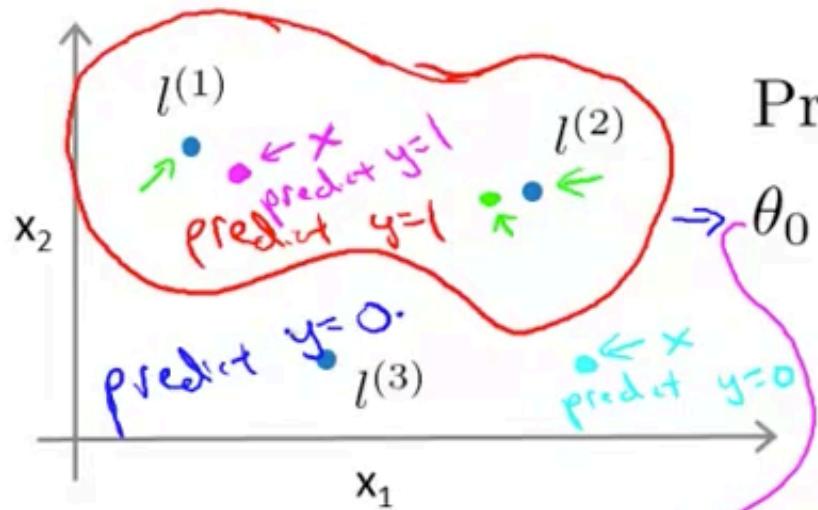
$$\rightarrow l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x-l^{(1)}\|^2}{2\sigma^2}\right)$$

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

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



Andrew Ng



Predict "1" when

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



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

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

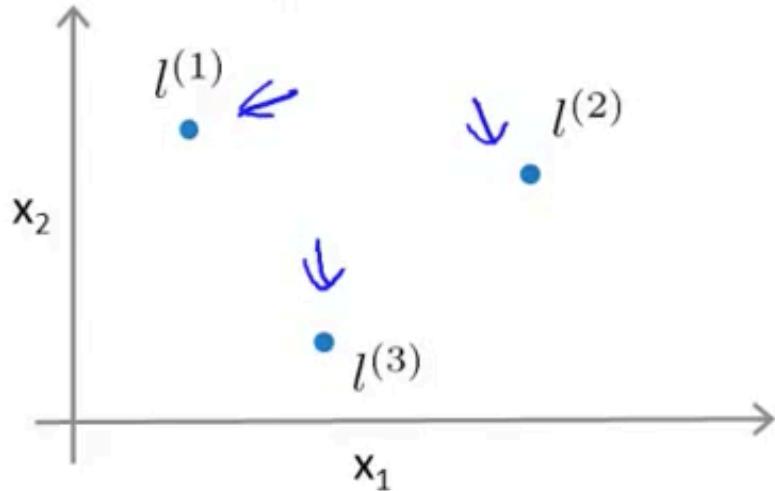
$$\theta_0 + \theta_1 \cdot 1 + \theta_2 \cdot 0 + \theta_3 \cdot 0$$

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

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

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

Choosing the landmarks

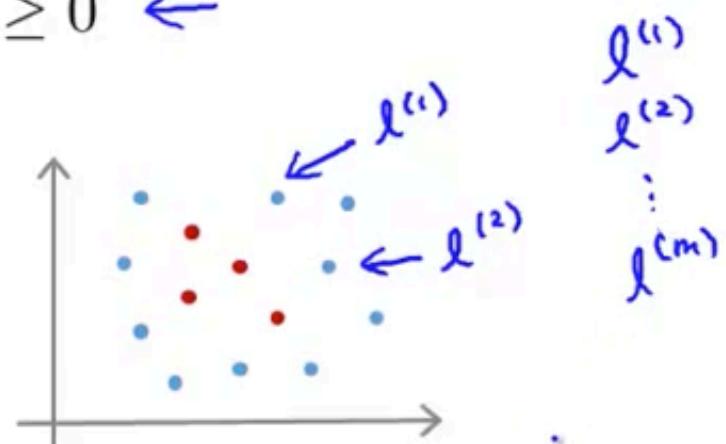
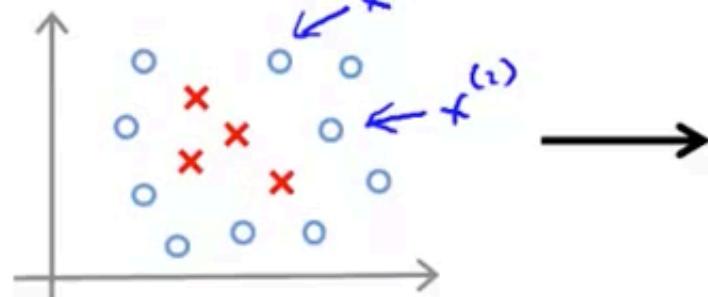


Given x :

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

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

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



SVM with Kernels

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

Given example \underline{x} :

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

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

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

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

$$\begin{aligned} \underline{x^{(i)}} \in \mathbb{R}^{n+1} \quad (\text{or } \mathbb{R}^n) \\ f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ f_2^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \\ f_0^{(i)} = 1 \end{aligned}$$

Andrew Ng

SVM with Kernels

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

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

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

$$\theta \in \mathbb{R}^{m+1}$$

Training:

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

$n = m + 1$

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

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

$\theta^T M \theta \leftarrow \|\theta\|^2$

$M = 10,000$

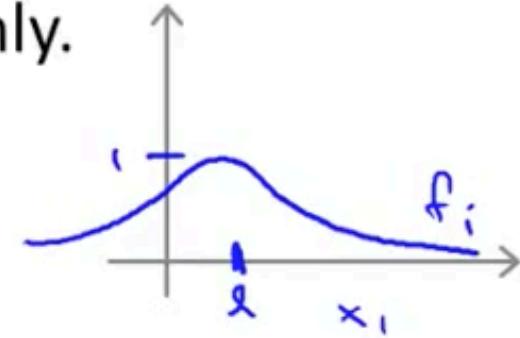
SVM parameters:

$C \left(= \frac{1}{\lambda} \right)$. \rightarrow Large C: Lower bias, high variance. (small λ)
 \rightarrow Small C: Higher bias, low variance. (large λ)

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

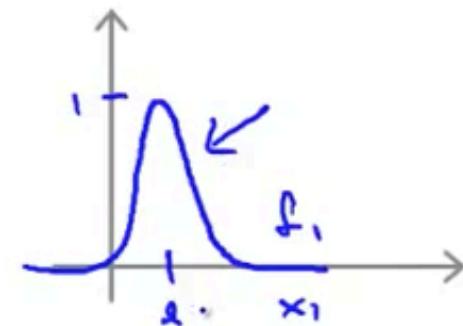
\rightarrow Higher bias, lower variance.

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



Small σ^2 : Features f_i vary less smoothly.

Lower bias, higher variance.



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 $\underline{\theta^T x} \geq 0$

$$\Theta_0 + \Theta_1 x_1 + \dots + \Theta_n x_n \geq 0$$

→ 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 $\underline{\sigma^2}$

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

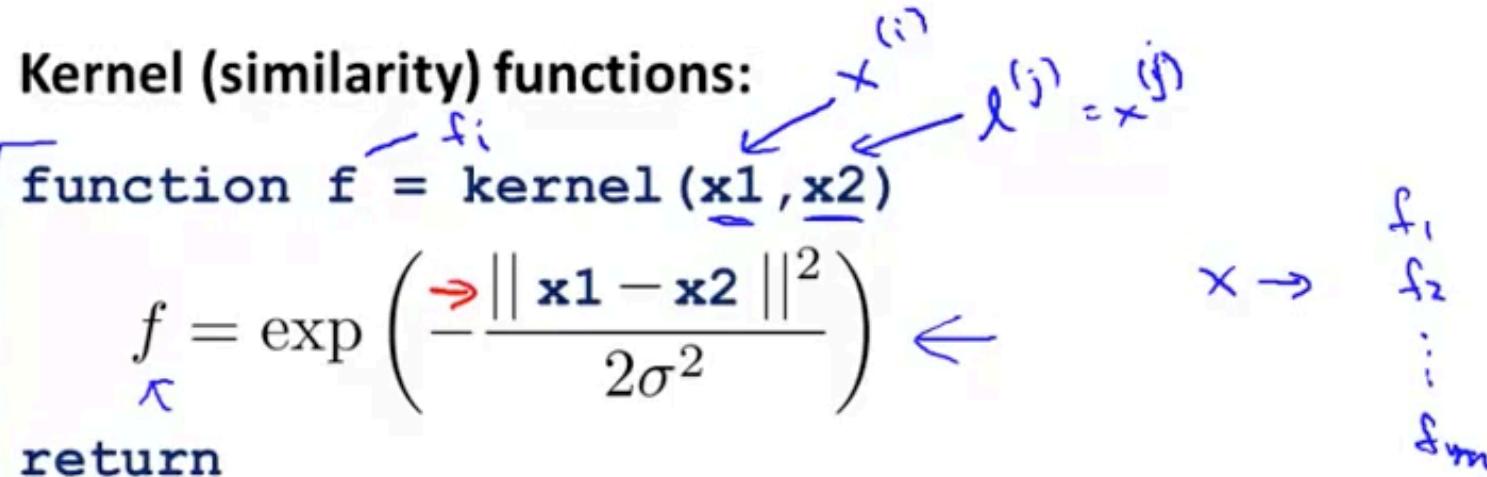


Kernel (similarity) functions:

function $f = \text{kernel}(\underline{x_1}, \underline{x_2})$

$$f = \exp\left(-\frac{\|\underline{x_1} - \underline{x_2}\|^2}{2\sigma^2}\right)$$

return



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

$$\begin{aligned} & \rightarrow \boxed{\|\underline{x} - \underline{l}\|^2} \quad \underline{x} \in \mathbb{R}^n \\ & \|\underline{v}\|^2 = v_1^2 + v_2^2 + \dots + v_n^2 \\ & = (x_1 - l_1)^2 + (x_2 - l_2)^2 + \dots + (x_n - l_n)^2 \\ & \quad \underbrace{1000 \text{ feet}^2}_{\text{1-5 bedrooms}} \end{aligned}$$

Other choices of kernel

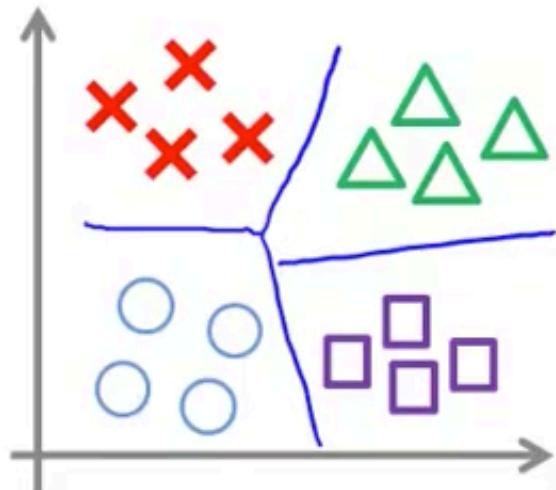
Note: Not all similarity functions $\text{similarity}(x, l)$ make valid kernels.

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

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 $\theta^{(1)}, \theta^{(2)}, \dots, \underline{\theta^{(K)}}$
Pick class i with largest $\underline{(\theta^{(i)})^T x}$

$$\begin{matrix} \uparrow \\ y=1 \end{matrix} \quad \begin{matrix} \nwarrow \\ y=2 \end{matrix} \quad \cdots \quad \begin{matrix} \nwarrow \\ \theta=K \end{matrix}$$

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. $n \geq m$, $n = \underline{10,000}$, $m = \underline{10 \dots 1000}$)
- Use logistic regression, or SVM without a kernel ("linear kernel")
- If n is small, m is intermediate: ($n = \underline{1-1000}$, $m = \underline{10 - 10,000}$)
 - Use SVM with Gaussian kernel
- If n is small, m is large: ($n = \underline{1-1000}$, $m = \underline{50,000+}$)
 - 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.

