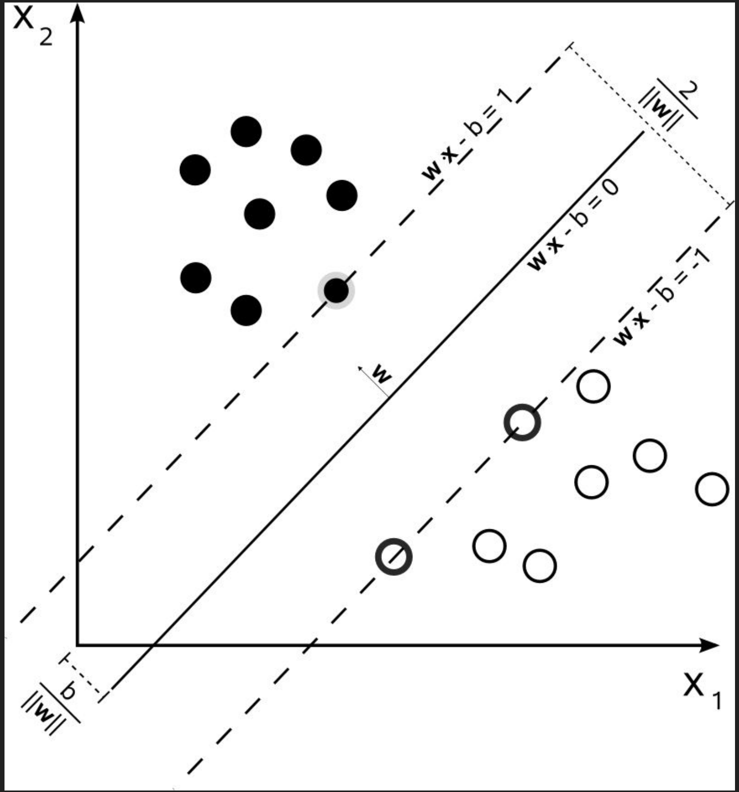
SVM
Support Vector Machines

Intuitions and Practical Uses

Support Vector Machines (SVMs) - Basic Ideas

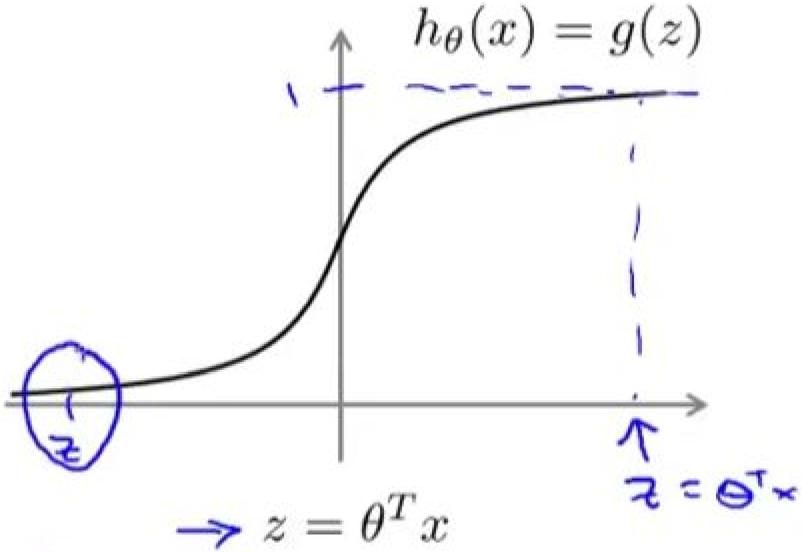
- Supervised learning model
- Used for classification and regression analysis
- We will focus on Classification tonight
- Constructs a hyperplane separating the classes
  - Hyperplane is a subspace of one less dimension than its ambient space
- Non-probabilistic
  - Uses Distances of sample data from class boundary - it is geometric
  - probabilistic classifier is a classifier that is able to predict, given a sample input, a probability distribution over a set of classes

Optimization Objectives Video on Coursera



Logistic Regression Function [1:30-2:56]

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



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

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

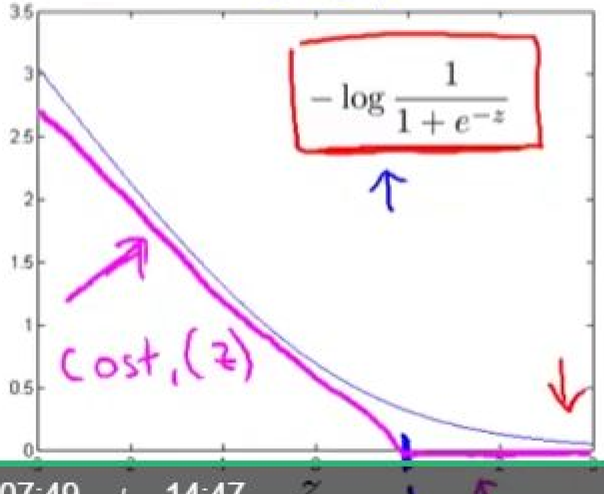
Representation Combines States [3:00-7:50]

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

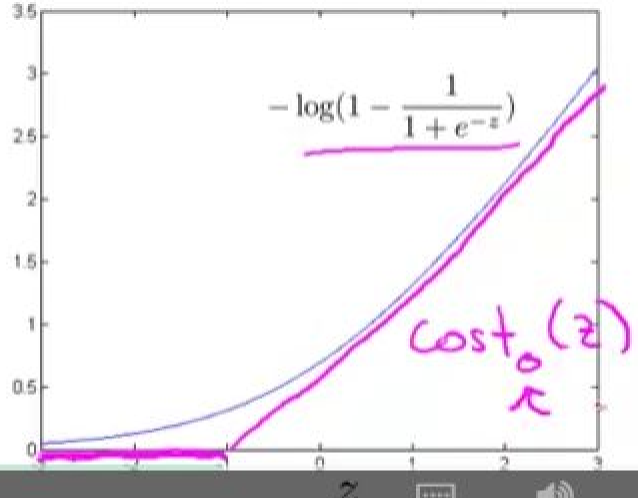
$$= \boxed{-y \log \frac{1}{1+e^{-\theta^T x}}} - \boxed{(1-y)\log\left(1-\frac{1}{1+e^{-\theta^T x}}\right)}$$

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

$z = \theta^T x$



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

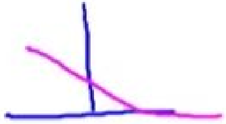


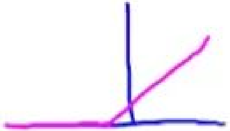
07:49 / 14:47

SVM vs. Logistic Regression Equations [7:50-13:50]

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:

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

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

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

$$A + \lambda B$$

$$C A + B$$

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

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

SVM Hypothesis [13:50-]

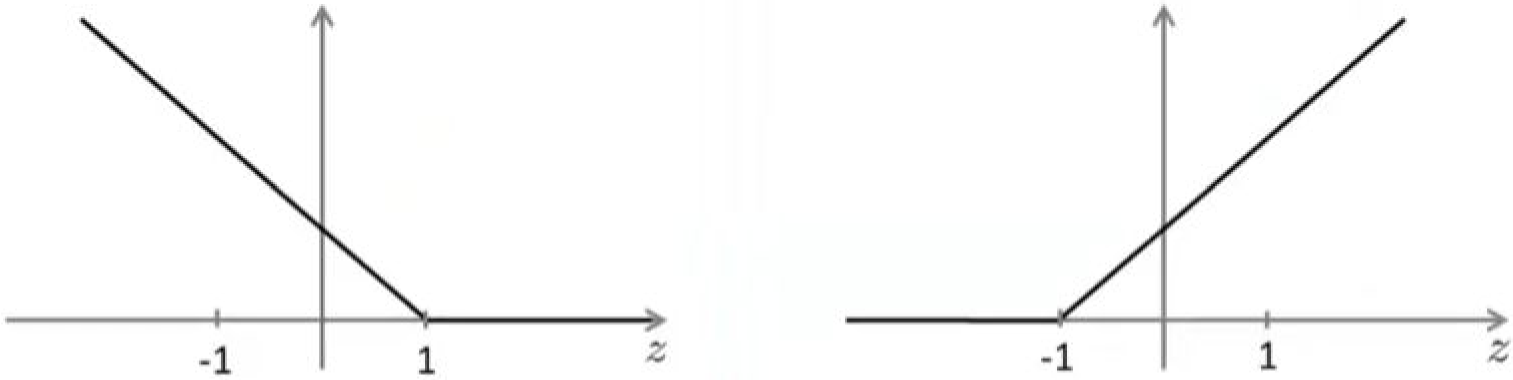
$$\min_{\theta} C \sum_{i=1}^{m} \left[y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

Hypothesis:

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

SVM Cost Function [0:17-2:31]

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

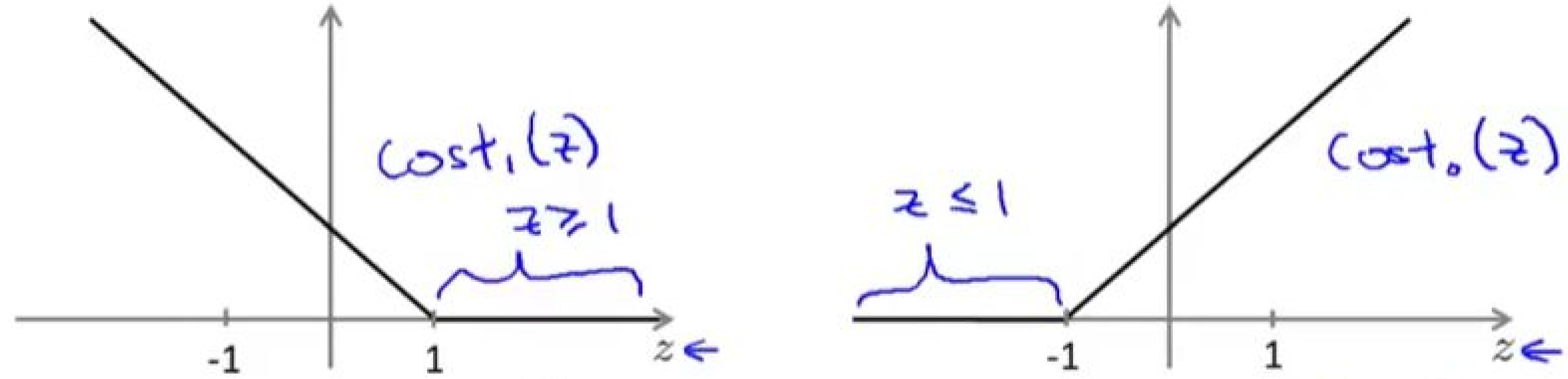


If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

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

SVM Cost Function cont

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



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

$C = 100{,}000$

SVM Decision Boundary [2:31-4:39]

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

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

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

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

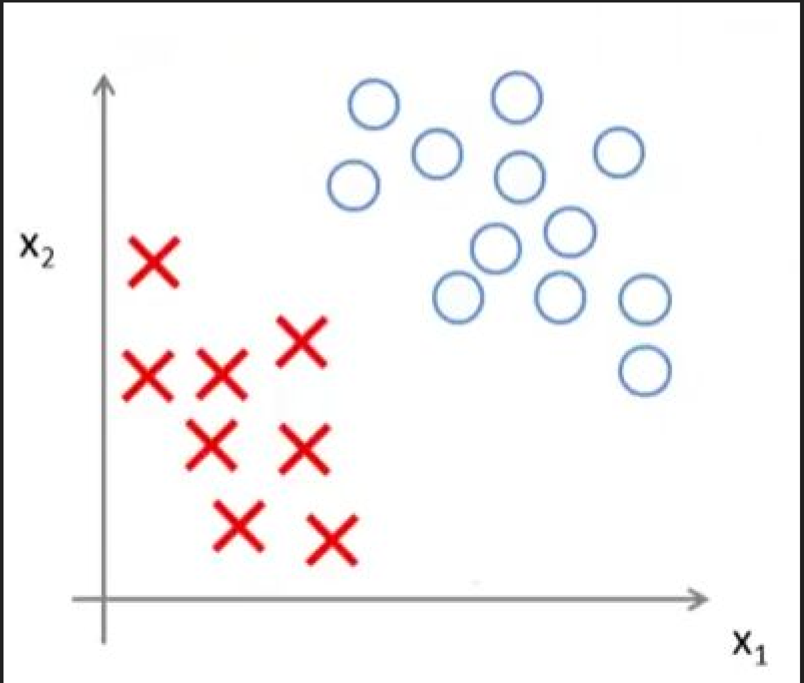
$\theta^T x^{(i)} \leq -1$

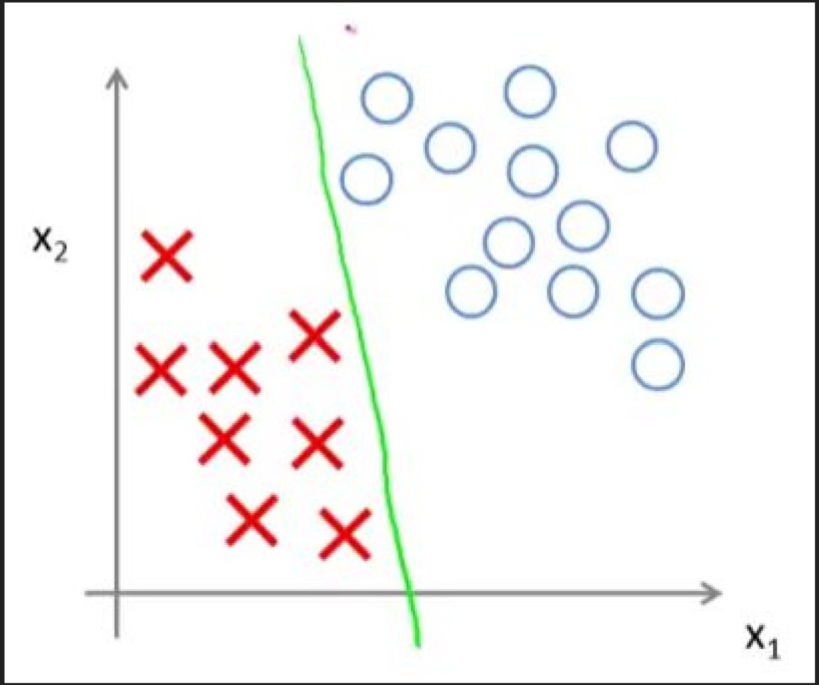
$$\min_{\theta} \cancel{C \times 0} + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

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

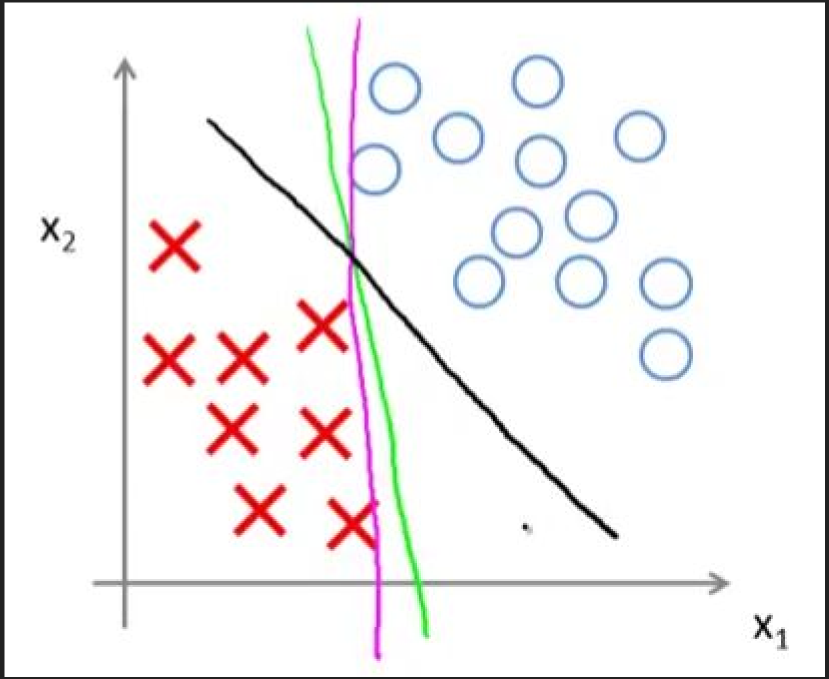
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$.

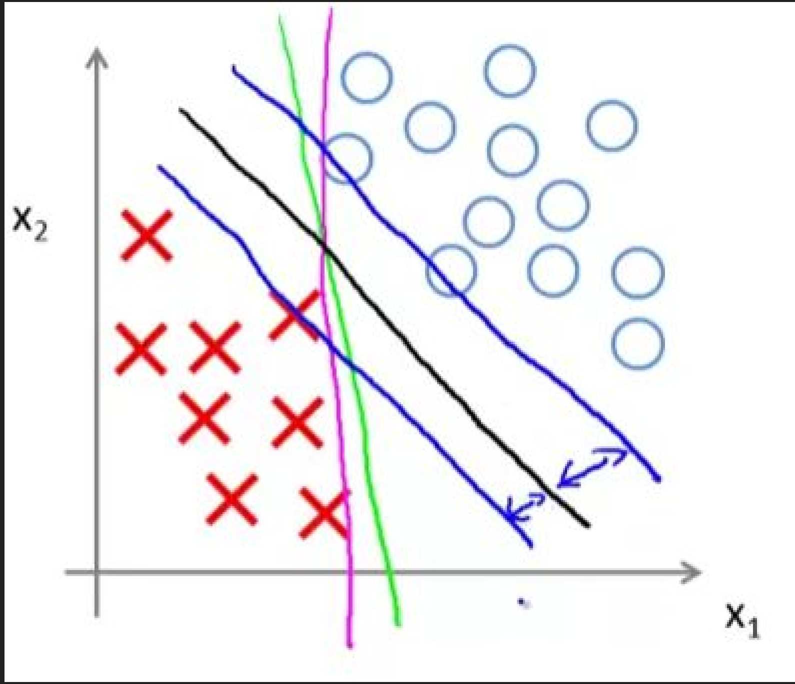
SVM Decision Boundary: Linearly Separable Case



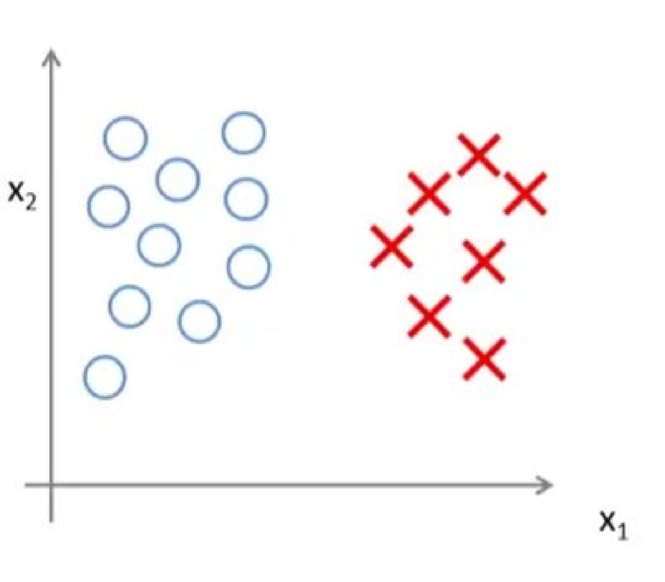


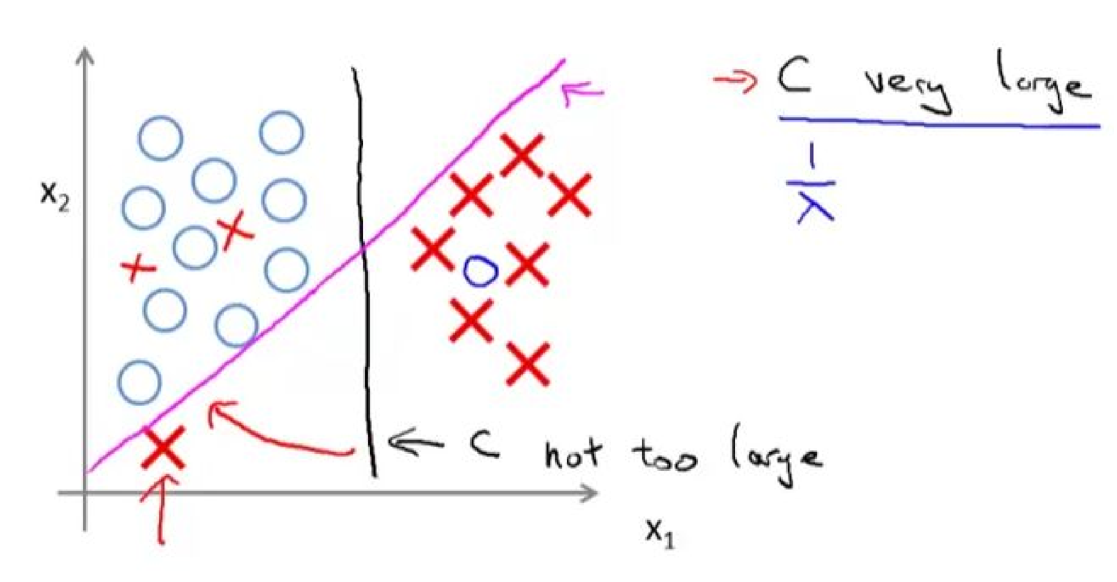
SVM



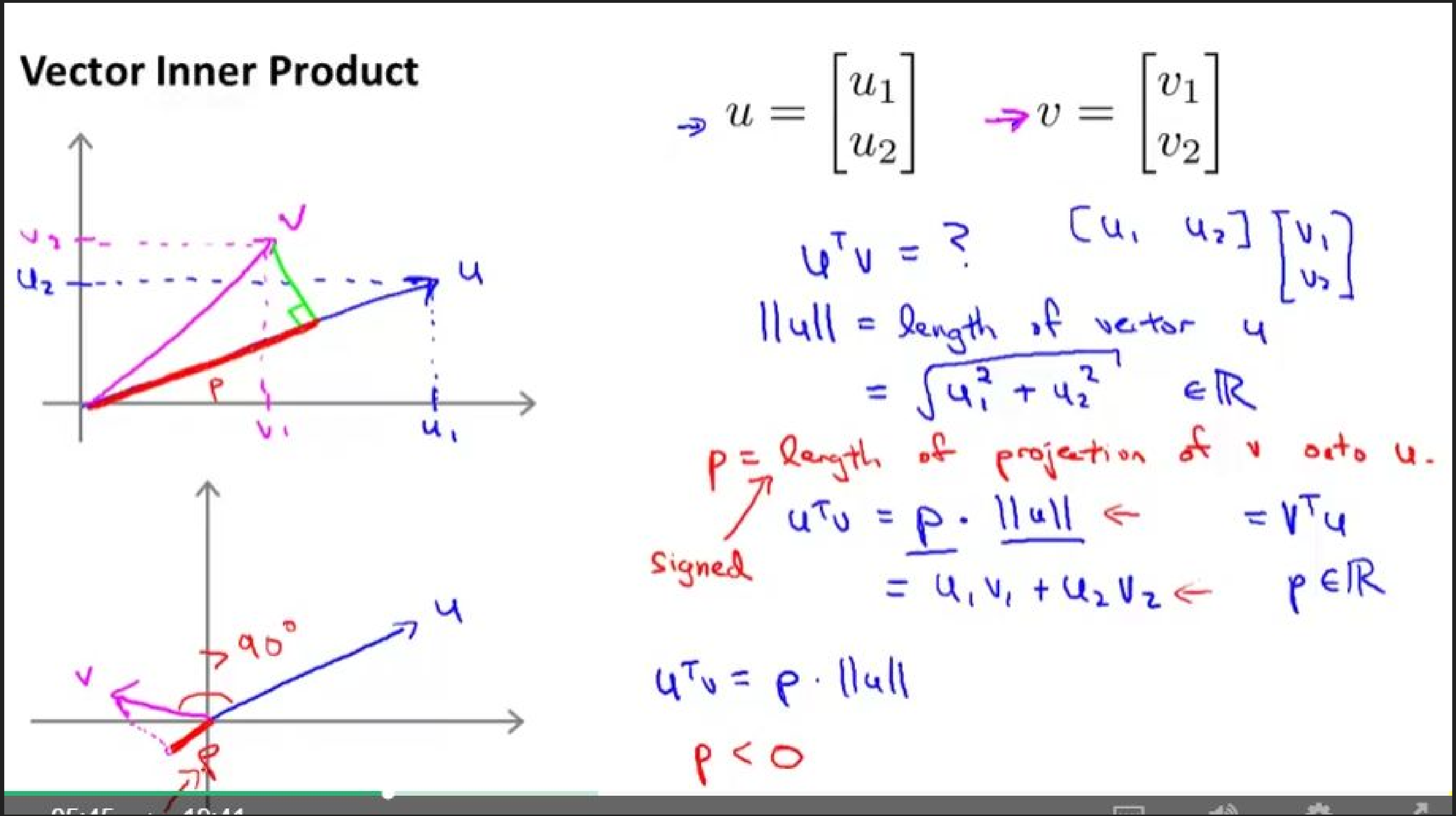


SVM Large Margin Classifier with Outliers [7:19-]





SVM Math - Vector Inner Product [0:00-5:45]



SVM Math - Decision Boundary [5:46-11:11]

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

SVM Decision Boundary

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

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

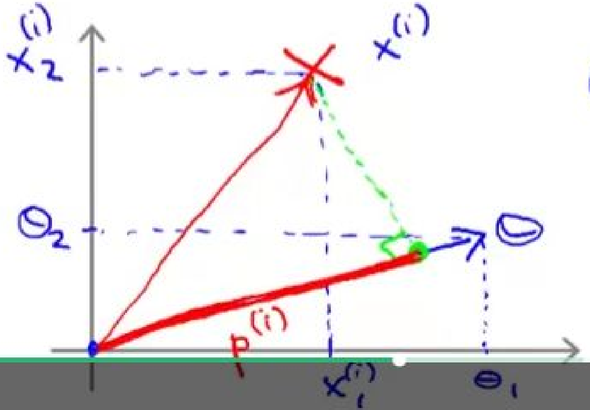
$\theta^T x^{(i)} \le -1$ if $y^{(i)} = 0$

Simplication: $\theta_0 = 0$. $n=2$

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

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

$u^T v$

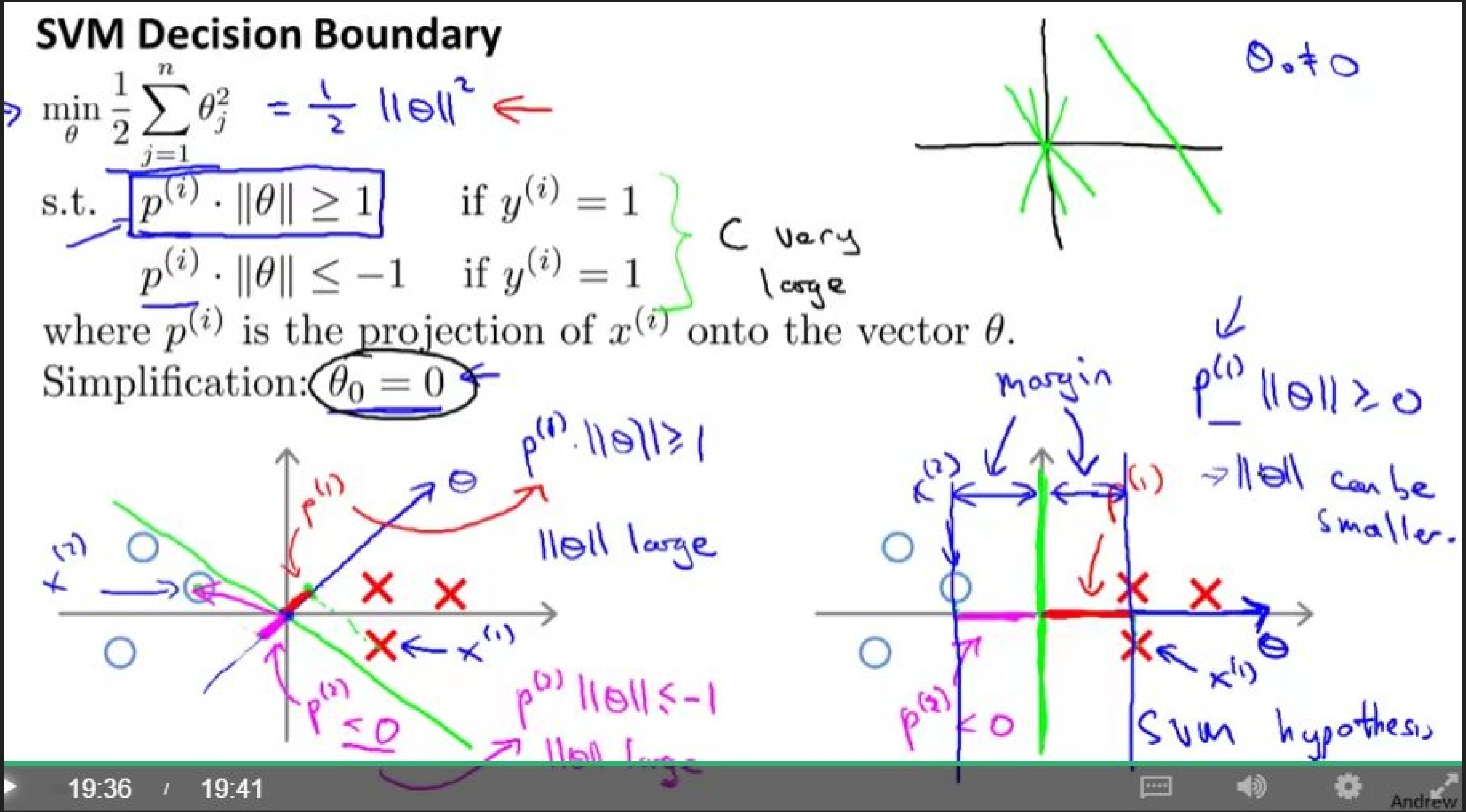


$\theta^T x^{(i)} = \boxed{p^{(i)} \cdot \|\theta\|}$

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

11:11 / 19:41

SVM Math - Decision Boundary



SVM - Using SVM

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

SVM Kernel (Similarity): Gaussian [4:34-8:25]

Kernel (similarity) functions:

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

f_i

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

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

```
return
```

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

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

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

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

$v = x - l$

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

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

SVM Kernels - Other choices

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).

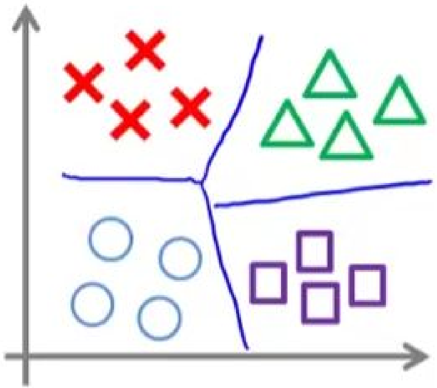
$(x^T l + \text{constant})^{\text{degree}}$

Many off-the-shelf kernels available:

- Polynomial kernel: $k(x, l) = (x^T l)^2$ ≥ 0
  $(x^T l)^3$, $(x^T l + 1)^3$, $(x^T l + 5)^4$
- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, …
  $\text{sim}(x, l)$

SVM - Multi-class Classification

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 $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)}$

y=1 y=2 .... θ=K

Pick class i with largest $(\theta^{(i)})^T x$

14:25 / 21:02

Logistic Regression vs. SVMs

Logistic regression vs. SVMs

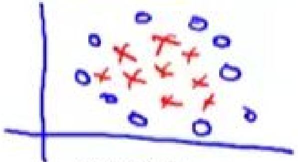
$n =$ number of features ($x \in \mathbb{R}^{n+1}$), $m =$ number of training examples

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.