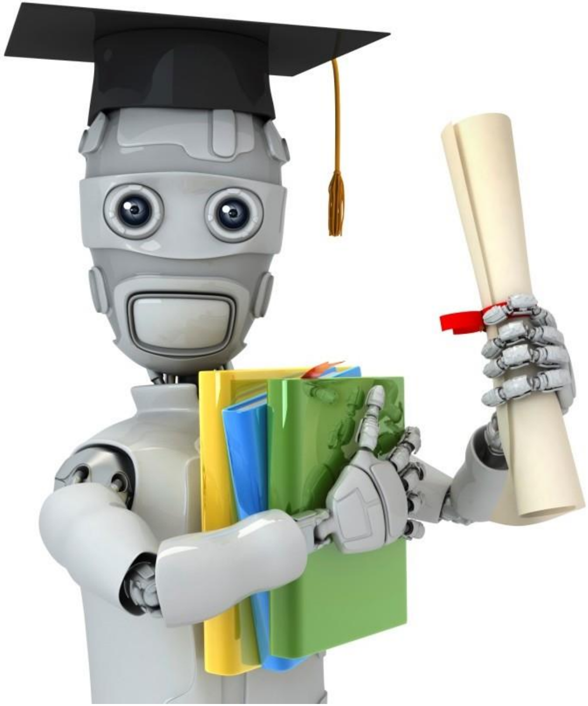


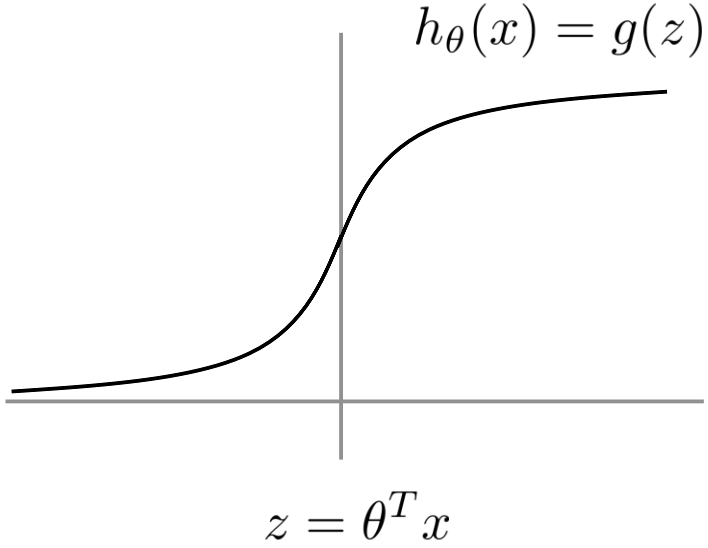
Machine Learning

Support Vector Machines

Optimization objective

Alternative view of logistic regression

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



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

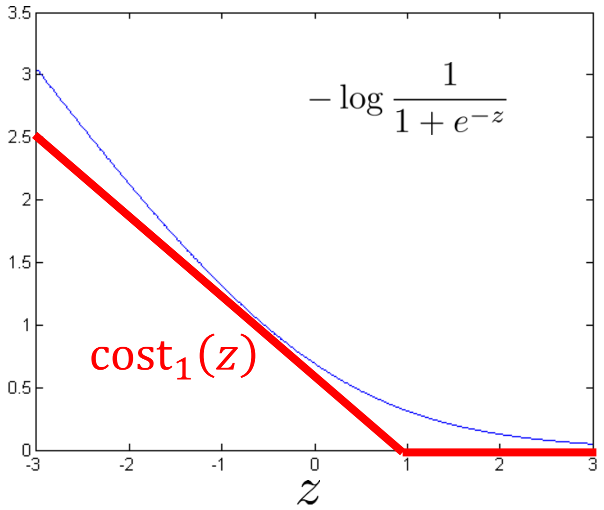
If $y = 0$, we want $h_\theta(x) \approx 0$, $\theta^T x \ll 0$

Alternative view of logistic regression

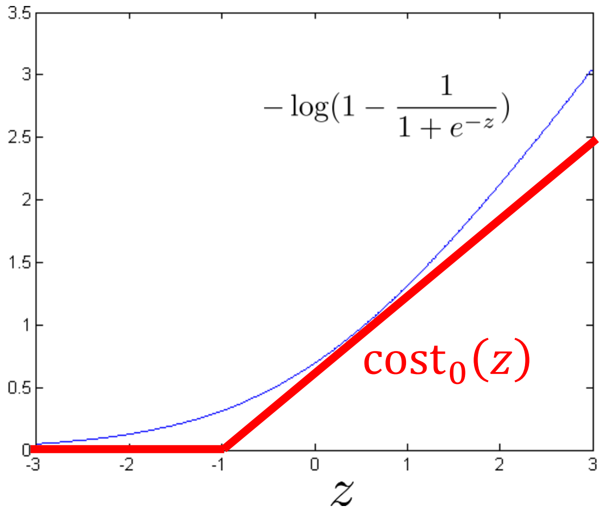
Cost of example: $-(y\log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

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

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



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



Support vector machine

Logistic regression:

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

Support vector machine:

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

Quiz

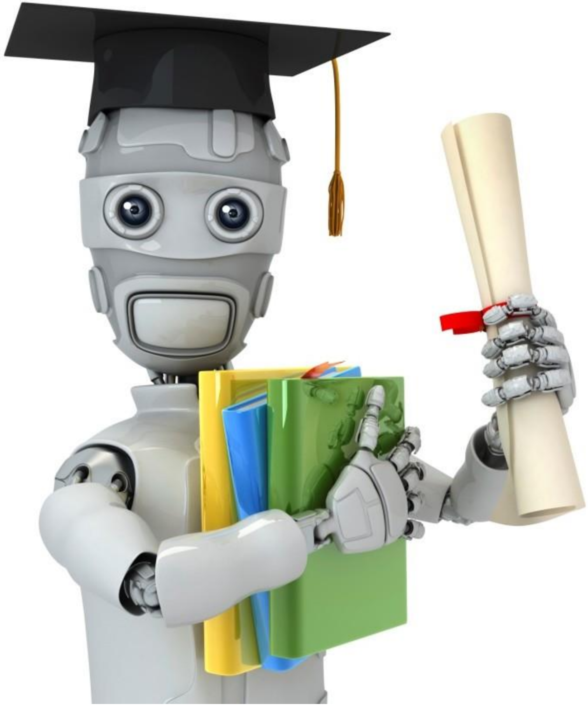
Consider the following minimization problems:

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

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

These two optimization problems will give the same value of θ (i.e., the same value of θ gives the optimal solution to both problems) if:

- $C = \lambda$
- $C = -\lambda$
- $\boxed{C = \frac{1}{\lambda}}$
- $C = \frac{2}{\lambda}$



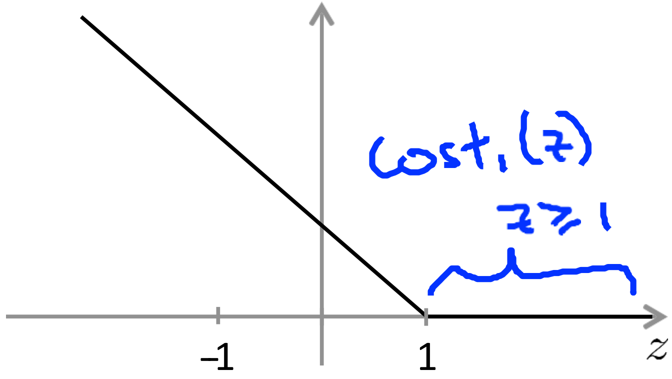
Machine Learning

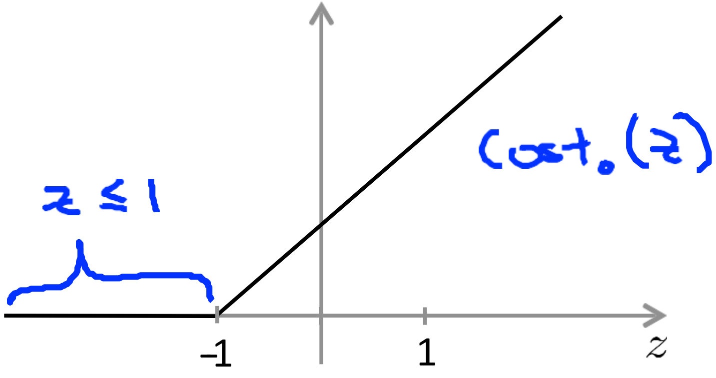
Support Vector Machines

Large Margin Intuition

Support Vector Machine

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





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

SVM Decision Boundary

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

$C = 100{,}000$

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

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

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

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

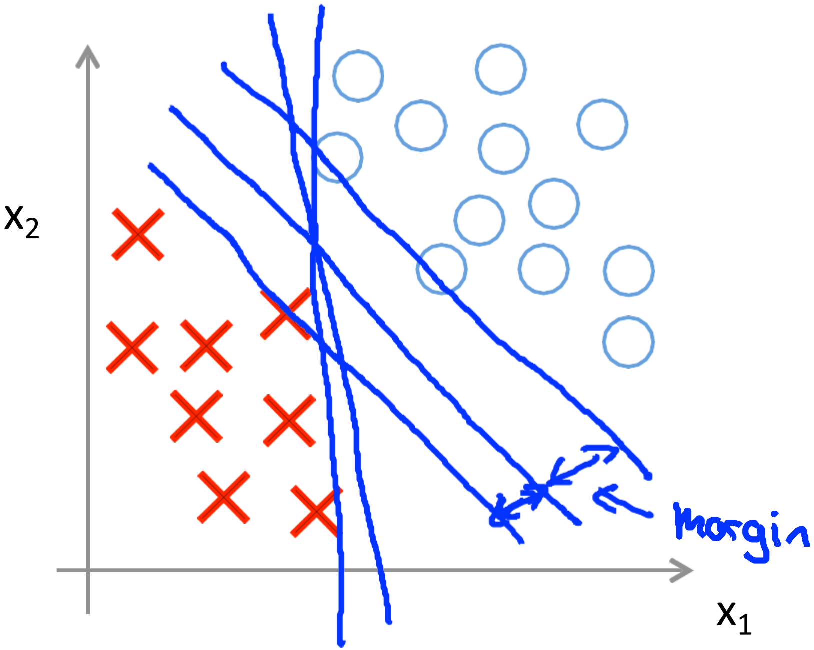
$\theta^T x^{(i)} \leq -1$ if $y^{(i)} = 0$.

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

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

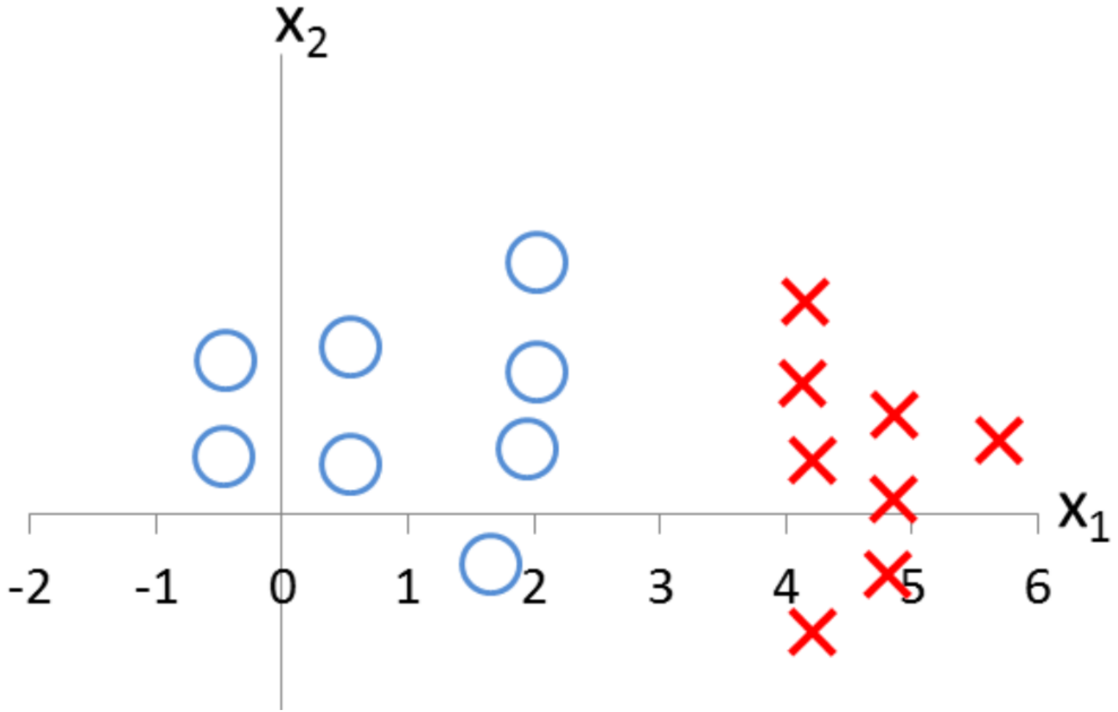
Andrew Ng

SVM Decision Boundary: Linearly separable case



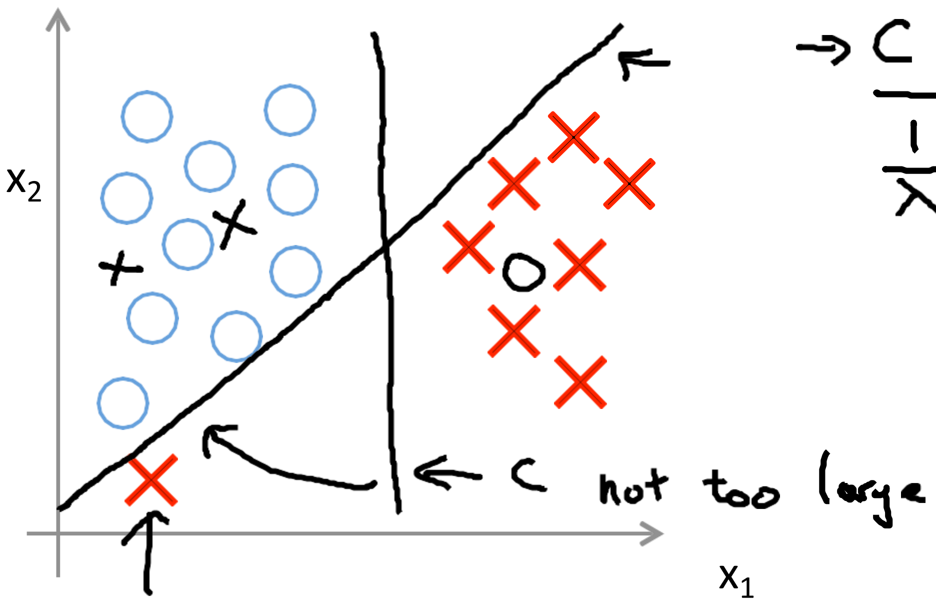
Large margin classifier

Consider the training set to the right, where "x" denotes positive examples ($y = 1$) and "o" denotes negative examples ($y = 0$). Suppose you train an SVM (which will predict 1 when $\theta_0 + \theta_1 x_1 + \theta_2 x_2 \geq 0$). What values might the SVM give for θ_0, θ_1, and θ_2?



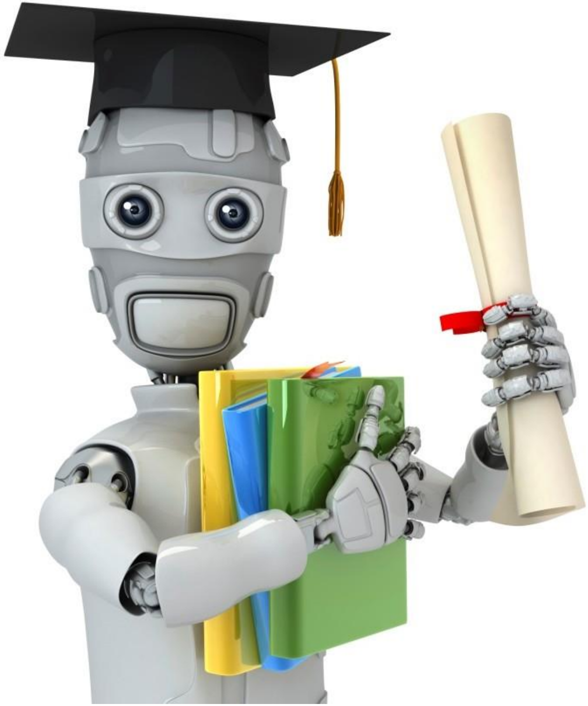
- $\theta_0 = 3, \theta_1 = 1, \theta_2 = 0$
- $\boxed{\theta_0 = -3, \theta_1 = 1, \theta_2 = 0}$
- $\theta_0 = 3, \theta_1 = 0, \theta_2 = 1$
- $\theta_0 = -3, \theta_1 = 0, \theta_2 = 1$

Large margin classifier in presence of outliers



$\rightarrow C$ very large

$\frac{1}{\lambda}$

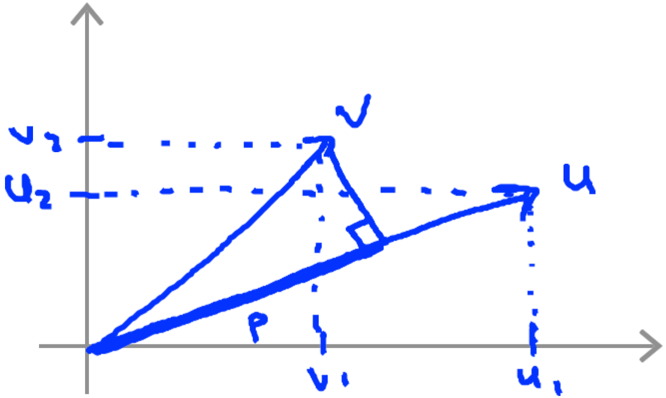


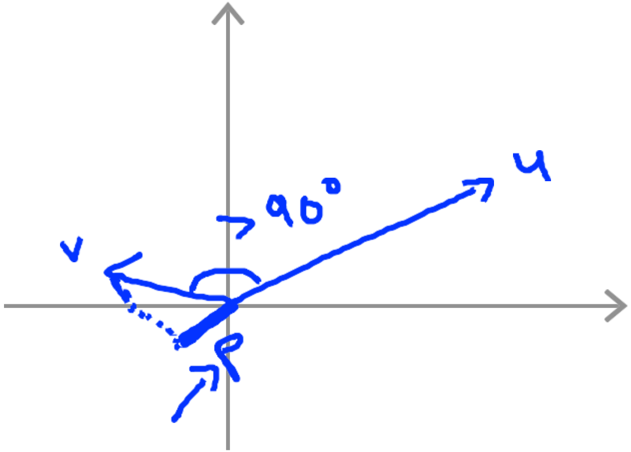
Machine Learning

Support Vector Machines

The mathematics behind large margin classification (optional)

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

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

p = length of projection of v onto u.

Signed

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

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

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

$p < 0$

SVM Decision Boundary

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

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

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

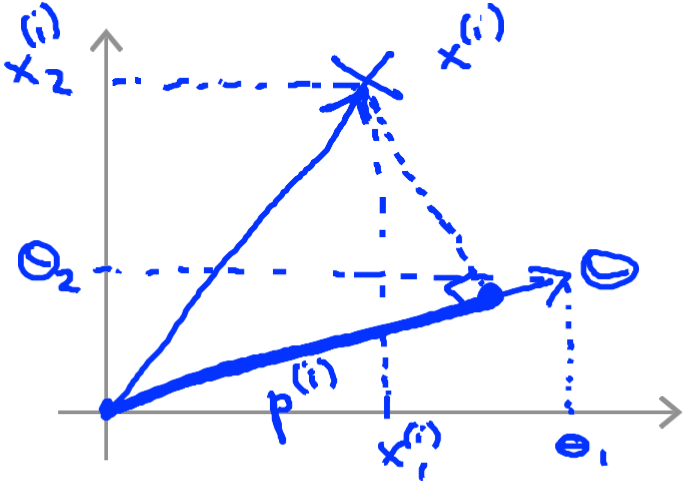
$\begin{bmatrix} \cancel{\theta_0} \\ \theta_1 \\ \theta_2 \end{bmatrix}$ $\theta_0 = 0$

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

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

$\uparrow \quad \uparrow$

$u^T v$



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

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

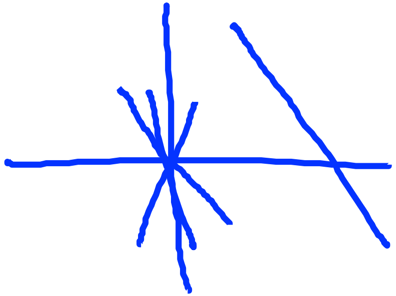
Andrew Ng

SVM Decision Boundary

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

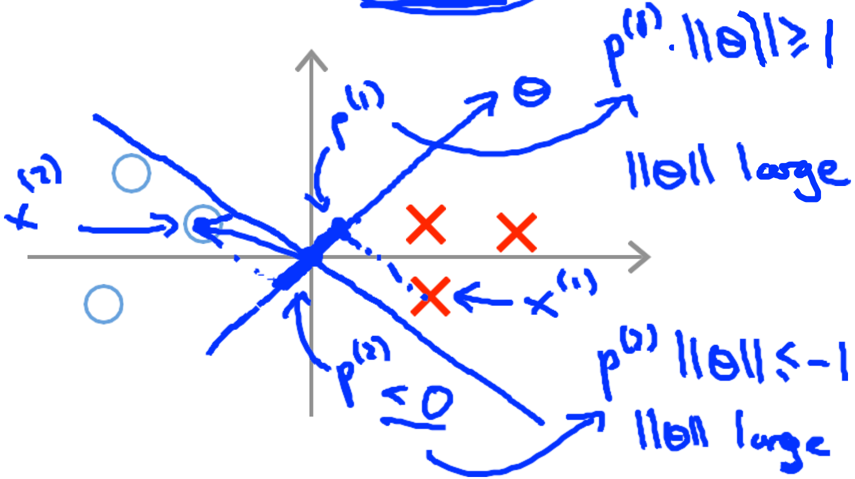
$$\left.\begin{array}{lll} \text{s.t.} & p^{(i)}\cdot\|\theta\| \geq 1 & \text{if } y^{(i)} = 1 \\ & p^{(i)}\cdot\|\theta\| \leq -1 & \text{if } y^{(i)} = 1 \end{array}\right\}$$

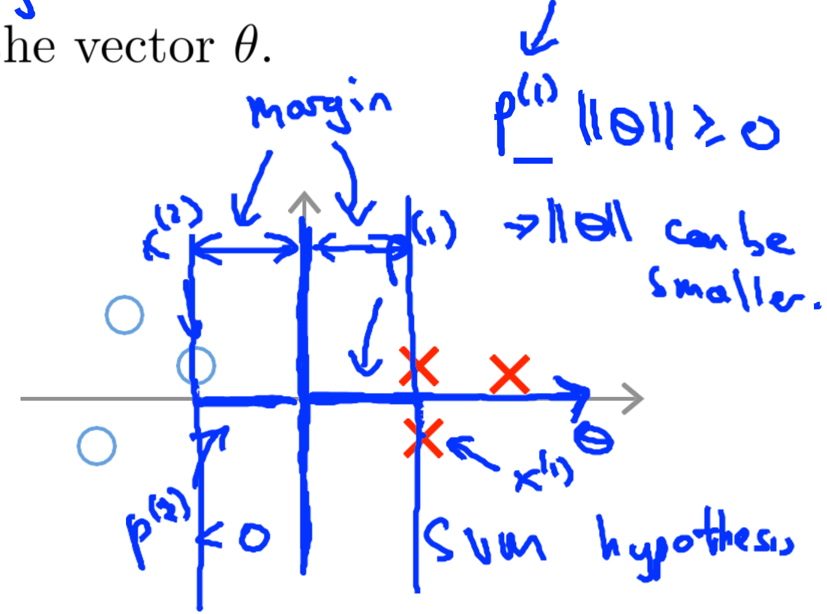
C very large



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

Simplification: $\theta_0 = 0$



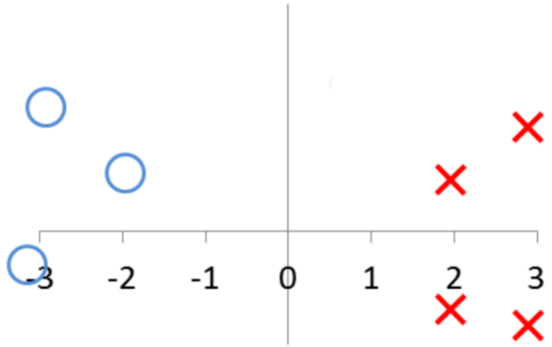


Quiz

The SVM optimization problem we used is:

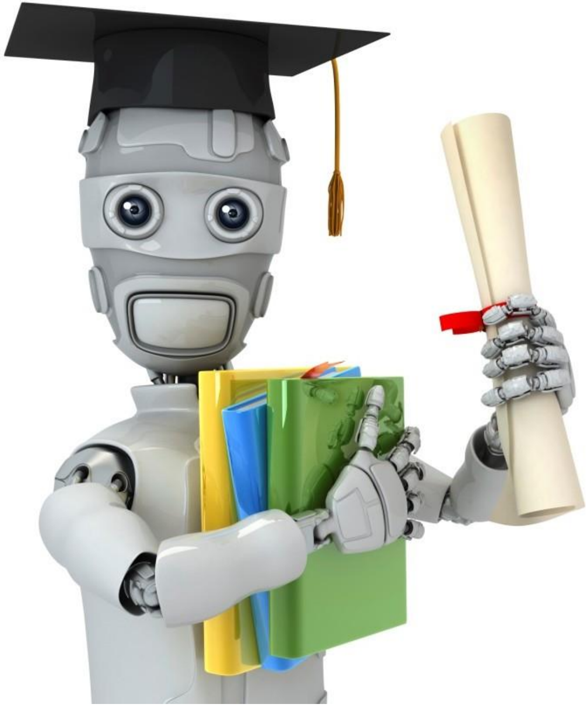
$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

$$\text{s.t. } \|\theta\| \cdot p^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$
$$\|\theta\| \cdot p^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$



where $p^{(i)}$ is the (signed - positive or negative) projection of $x^{(i)}$ onto θ. Consider the training set above. At the optimal value of θ, what is $\|\theta\|$?

- 1/4
- $\boxed{\text{1/2}}$
- 1
- 2

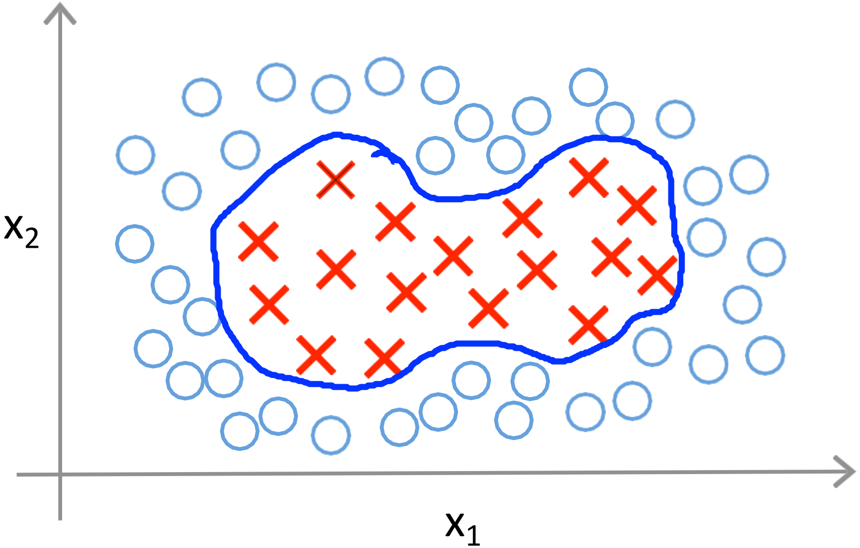


Machine Learning

Support Vector Machines

Kernels I

Non-linear Decision Boundary



Predict $y = 1$ if

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

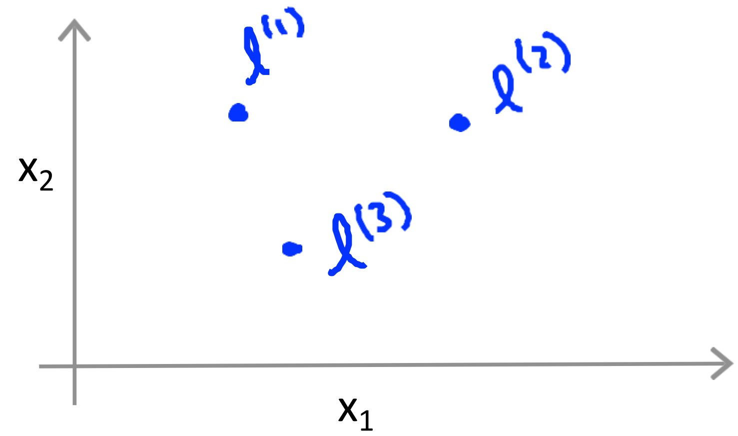
$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \cdots \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

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

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

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

Kernel



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

Given x:

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

$$f_2 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$$

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

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

Kernels and Similarity

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

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

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

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

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

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

x

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

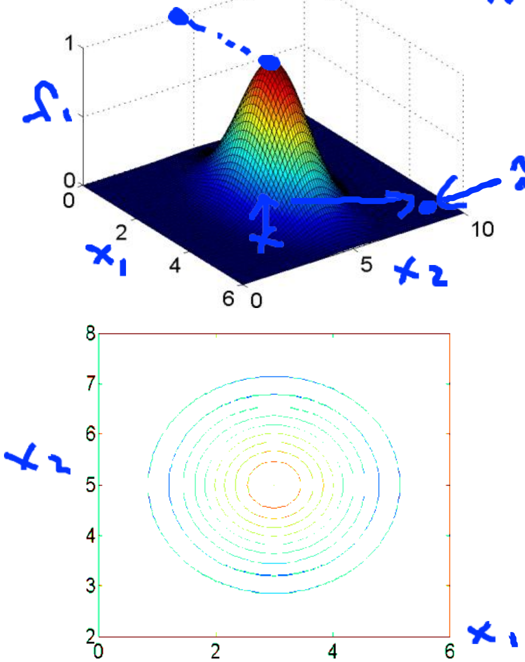
$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \approx 0.$$

Example:

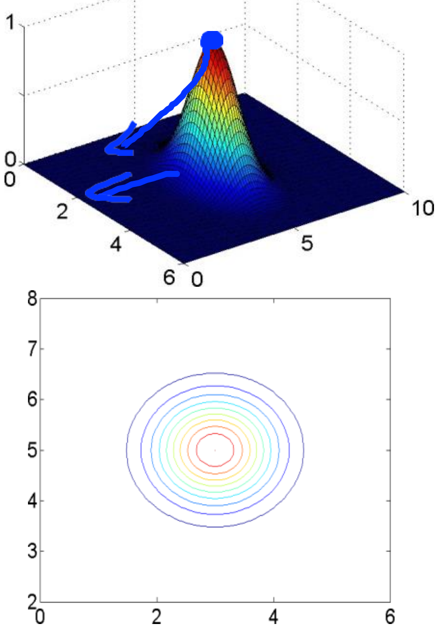
$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$\sigma^2 = 1$

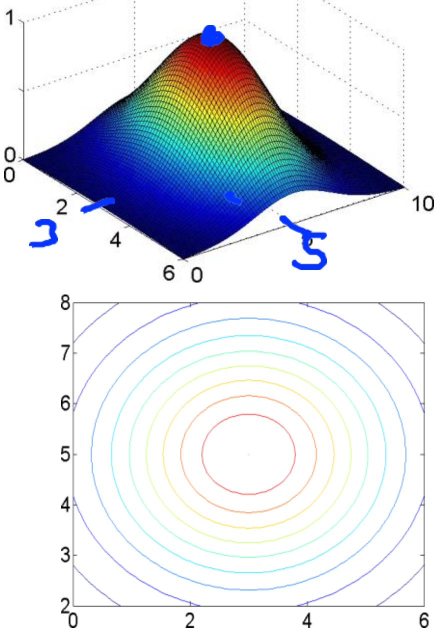
$x = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$



$\sigma^2 = 0.5$



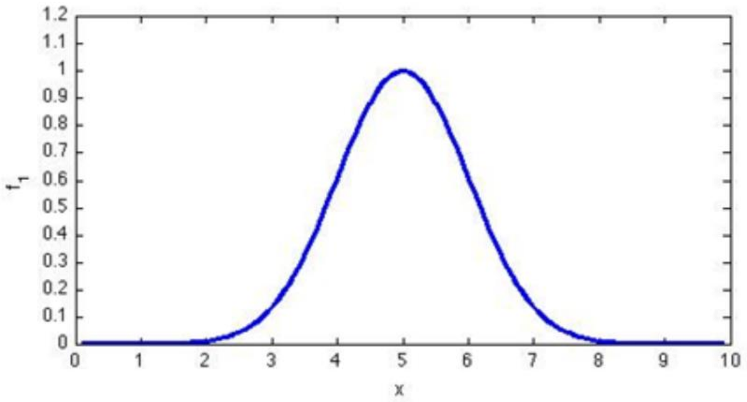
$\sigma^2 = 3$

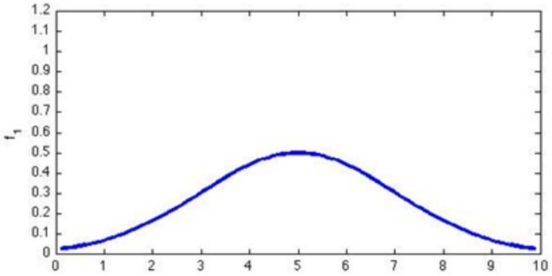


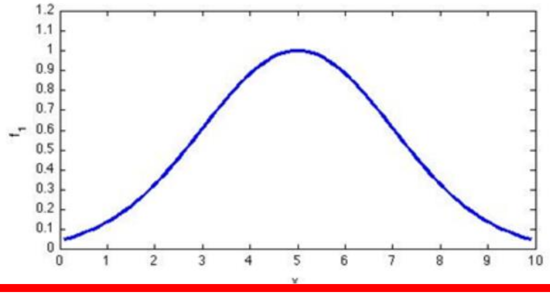
Andrew Ng

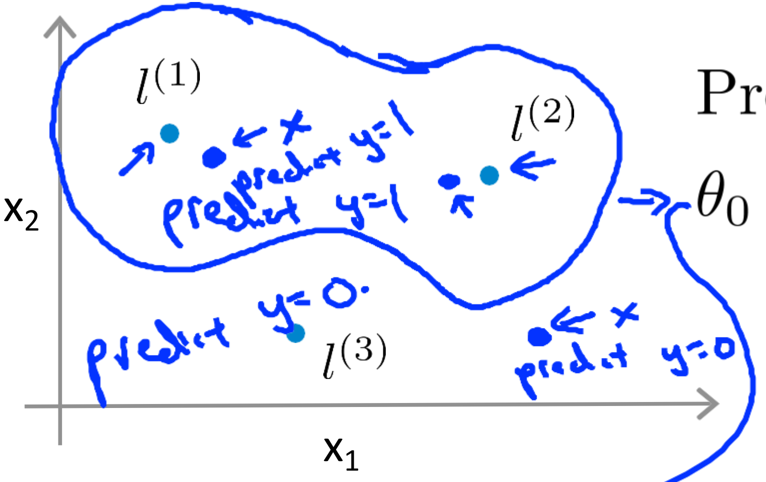
Quiz

Consider a 1-D example with one feature x_1. Suppose $l^{(1)} = 5$. To the right is a plot of $f_1 = \exp(-\frac{\| x_1 - l^{(1)} \|}{2\sigma^2})$ when $\sigma^2 = 1$. Suppose we now change $\sigma^2 = 4$. Which of the following is a plot of f_1 with the new value of σ^2?









Predict "1" when

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

x

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

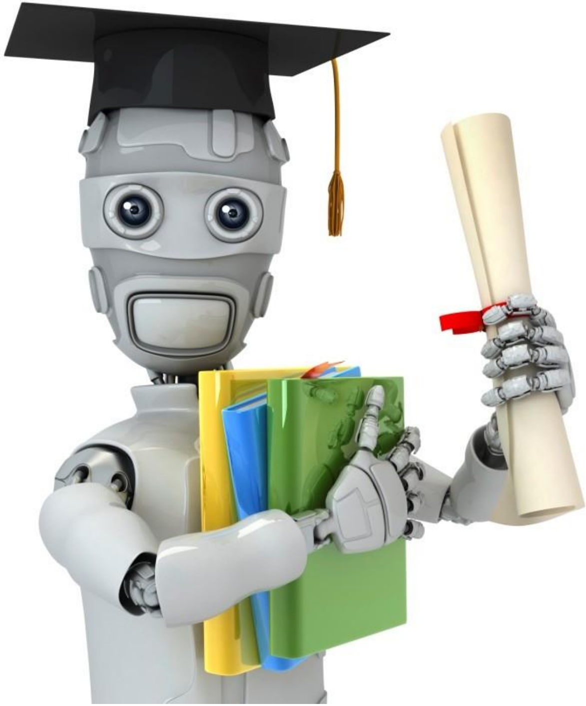
$f_1 \approx 1, \quad f_2 \approx 0, \quad f_3 \approx 0.$

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

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

$f_1, f_2, f_3 \approx 0$

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

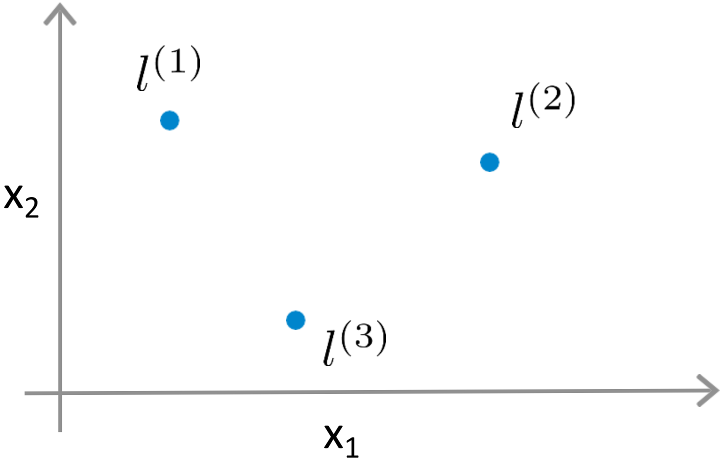


Machine Learning

Support Vector Machines

Kernels II

Choosing the landmarks

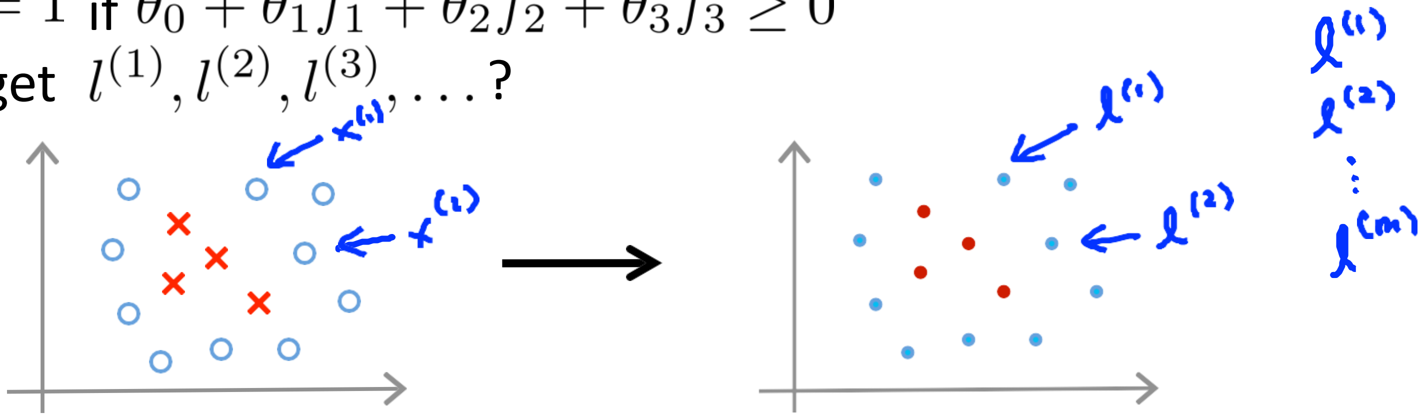


Given x:

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

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

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



Andrew Ng

SVM with Kernels

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:

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

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

$$\ldots$$

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

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

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

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

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

SVM with Kernels

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

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

Training:

$n = m$

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

$\cancel{n} = m$

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

$\rightarrow \theta_0$

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

$\|\theta\|^2$

$\underline{\theta^T M \theta}$

$m = 10{,}000$

SVM parameters:

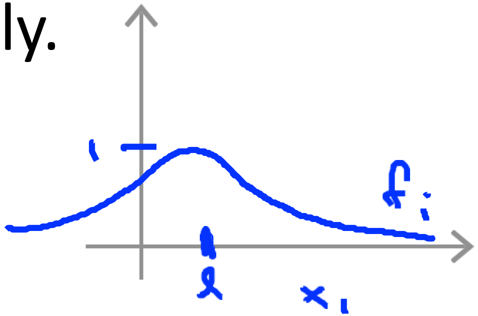
C ($= \frac{1}{\lambda}$). Large C: Lower bias, high variance. (small λ)

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

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

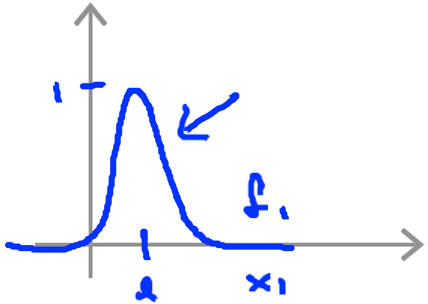
Higher bias, lower variance.

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



Small σ^2: Features f_i vary less smoothly.

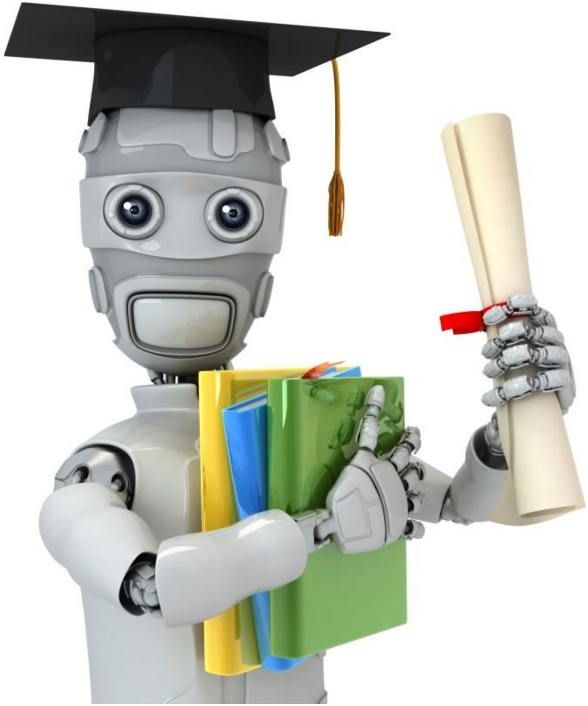
Lower bias, higher variance.



Quiz

Suppose you train an SVM and find it overfits your training data. Which of these would be a reasonable next step? Check all that apply.

- [ ] Increase C
- [ ] Decrease C
- [ ] Increase σ^2
- [ ] Decrease σ^2



Machine Learning

Support Vector Machines

Using an SVM

Use SVM so]ware 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

m large

Quiz

Suppose you are trying to decide among a few different choices of kernel and are also choosing parameters such as C, σ^2, etc. How should you make the choice?

- ○ Choose whatever performs best on the training data.
- ○ Choose whatever performs best on the cross-validation data.
- ○ Choose whatever performs best on the test data.
- ○ Choose whatever gives the largest SVM margin.

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
```

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

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

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

$v = x - l$

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

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

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:

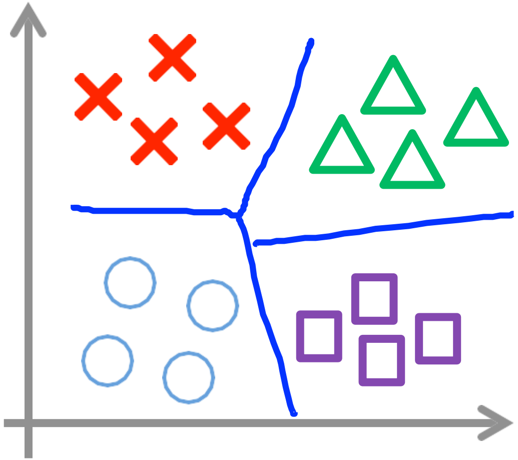
- Polynomial kernel:

$$k(x, l) = (x^T l + c)^d$$

- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, …

$\text{sim}(x, l)$

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

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

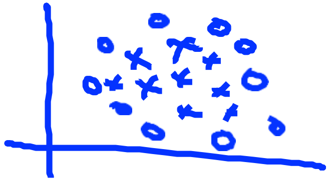
Use logistic regression, or SVM without a kernel ("linear kernel")

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: ($n = 1 - 1000$, $\underline{m} = \underline{50{,}000+}$)

Create/add more features, then use logistic regression or SVM without a kernel



References

- Andrew Ng, Coursera: Machine Learning, https://www.coursera.org/learn/machine-learning