

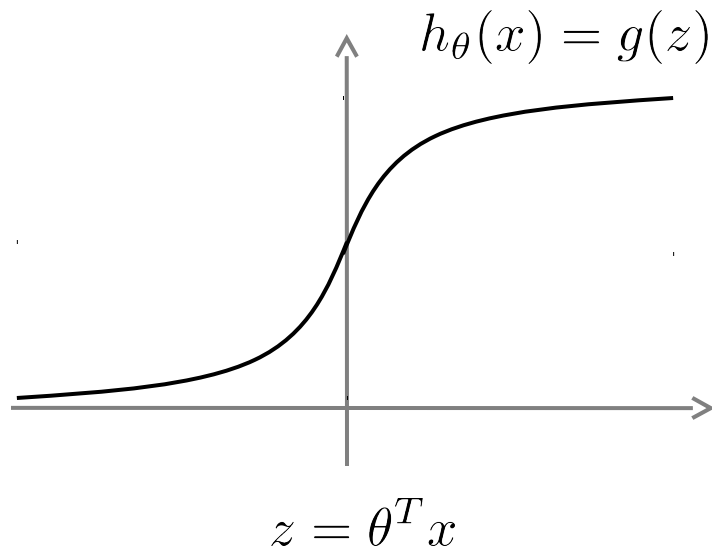


Machine Learning

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  $h_{\theta}(x) \approx 1$   $\theta^T x \gg 0$

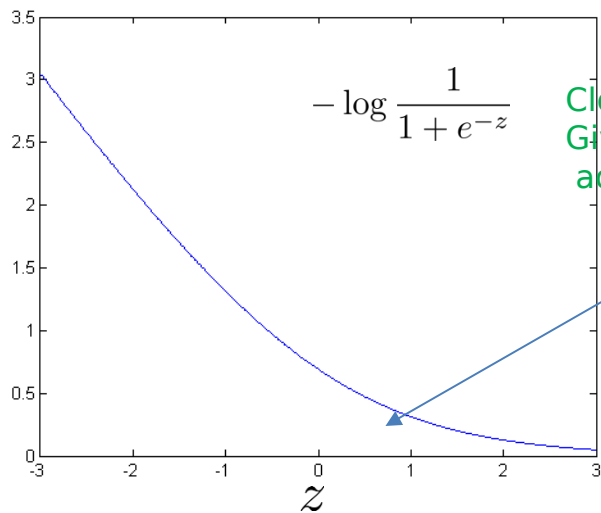
If  $y = 0$  , we want  $h_{\theta}(x) \approx 0$   $\theta^T x \ll 0$

# Alternative view of logistic regression

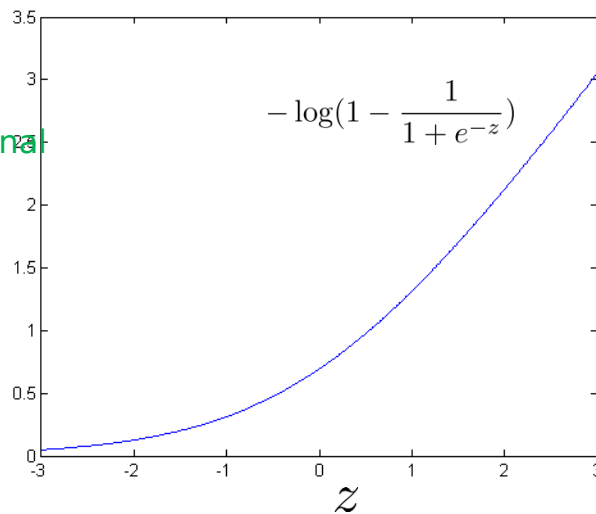
Cost of example:  $(y \log h_{\theta}(x) + (1 - y) \log(1 - h_{\theta}(x)))$

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

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



Close approximation :  
Gives SVM computational  
advantage



# 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:

Example

Removing m will not degrade Optimal value much ?

In Logistic Regression we care more about 2<sup>nd</sup> term i.e. B

In SVM we care more about first term i.e. A

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

Handwritten examples:

$\min_u \frac{(u-5)^2 + 1}{10} \rightarrow u=5$

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

Diagram illustrating the relationship between the cost function terms:

$A + \lambda B \leftarrow$

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

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

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

# SVM hypothesis

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

Hypothesis:

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

Consider the following minimization problems:

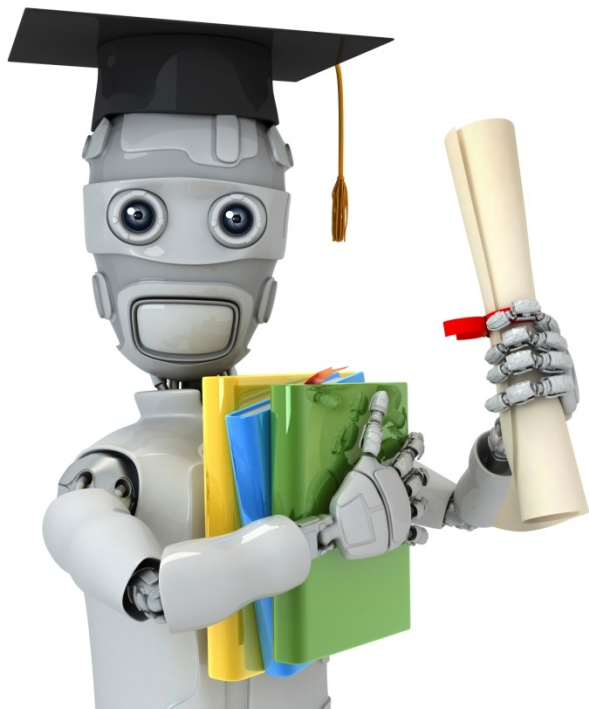
1.  $\min_{\theta} \frac{1}{m} \left[ \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T x^{(i)}) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$
2.  $\min_{\theta} C \left[ \sum_{i=1}^m y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1 - y^{(i)}) \text{cost}_0(\theta^T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

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

- ☐  $C = \lambda$
- ☐  $C = -\lambda$
- ☒  $C = \frac{1}{\lambda}$

**Correct Response**

- ☐  $C = \frac{2}{\lambda}$



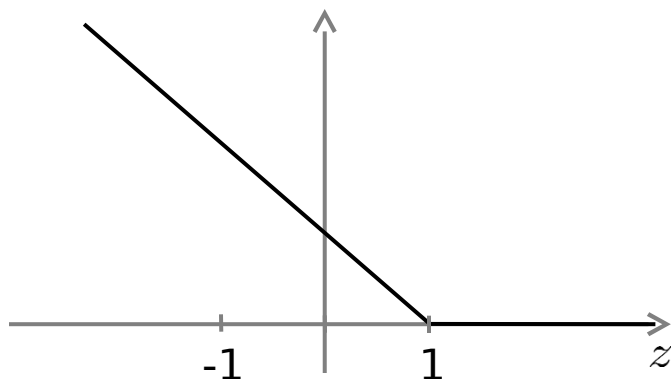
Machine Learning

Support  
Vector  
Machines  
Large Margin  
Intuition

---

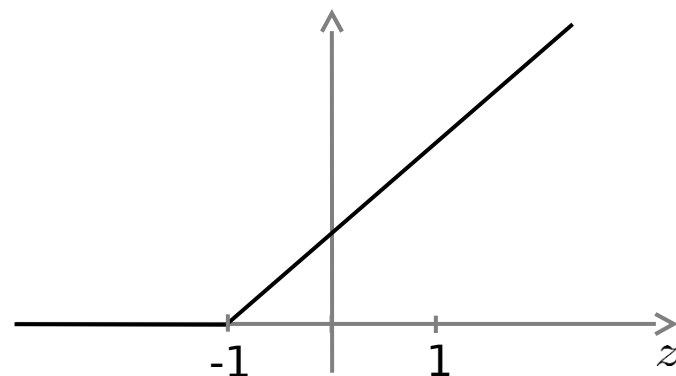
# Support Vector Machine

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



If  $y = 1$  , we want  $\theta^T x \geq 1$

If  $y = 0$  , we want  $\theta^T x \leq -1$



(not just )

(not just )



# SVM Decision Boundary

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

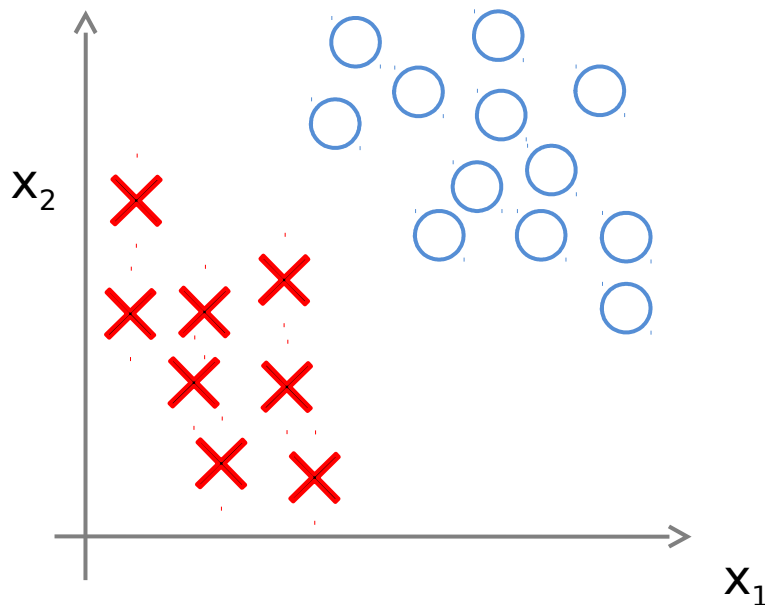
Whenever  $y^{(i)} = 1$  :

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

Optimization  
objective  
Of SVM

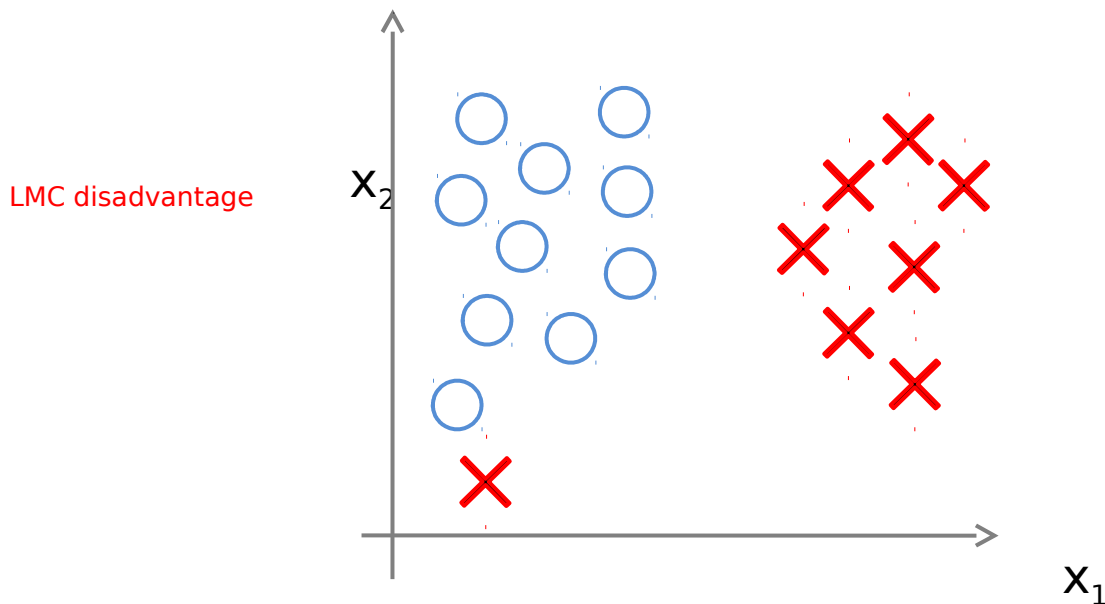
Suppose we have set C to large value, Now in order to minimize this equation we have to make term (under box) to zero. Which can be done by setting  $\text{cost}_i() = 0$ , which can be achieved by manipulating (Theta' x)

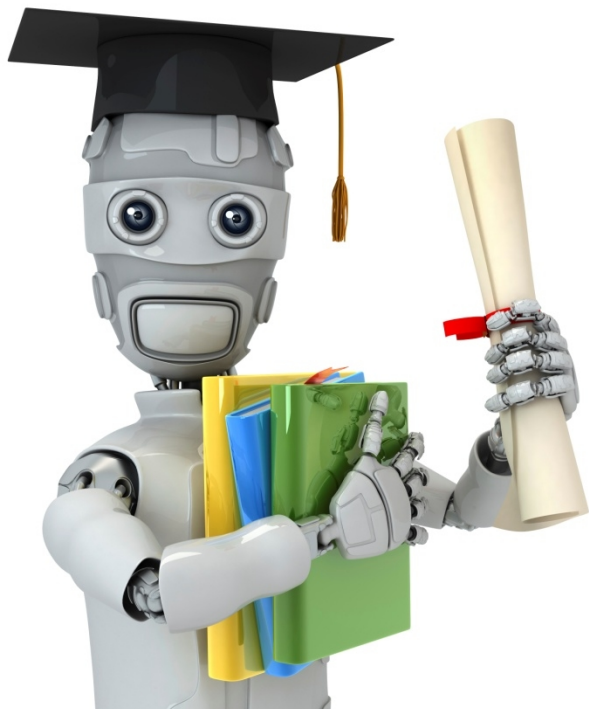
# SVM Decision Boundary: Linearly separable case



Large margin classifier

# Large margin classifier in presence of outliers





Machine Learning

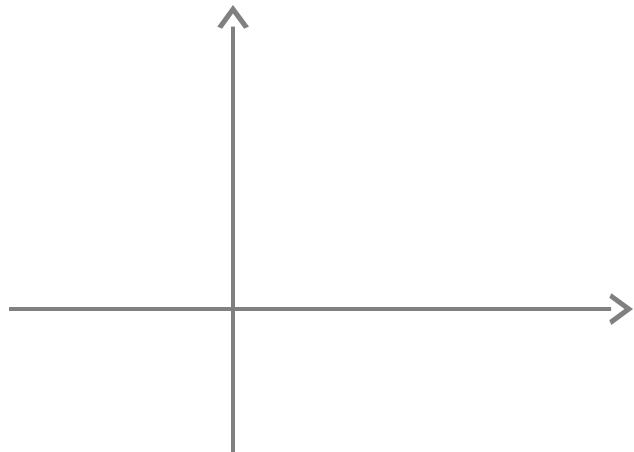
Support  
Vector  
The mathematics  
Machines  
behind large  
margin  
classification  
(optional)

# Vector Inner Product



$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$



Projecting  $u$  on to  $v$   
Will also give same  
result



# SVM Decision Boundary

Optimization objective  
Of SVM

$$\min_{\theta} \frac{1}{2} \sum_{j=1}^n \theta_j^2$$
$$\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$
$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$



# SVM Decision Boundary

If we set  $\theta = 0$  then we are allowing  
Decision boundaries that pass through  
the origin

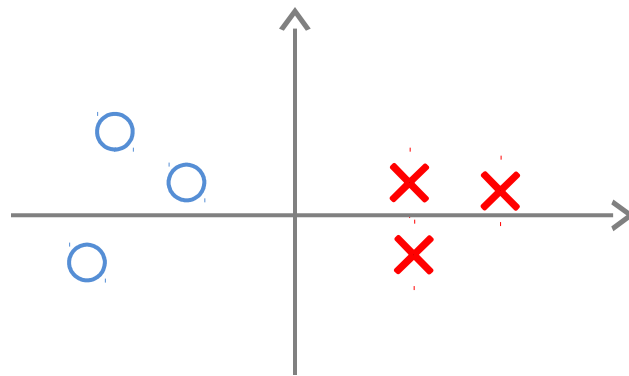
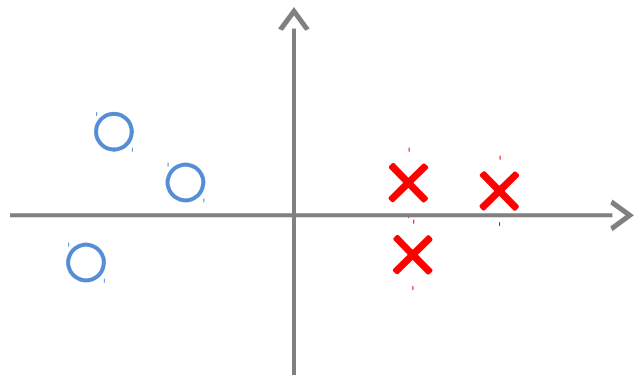
$$\min_{\theta} \frac{1}{2} \sum_{j=1}^n \theta_j^2$$

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

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

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

Simplification:  $\theta_0 = 0$

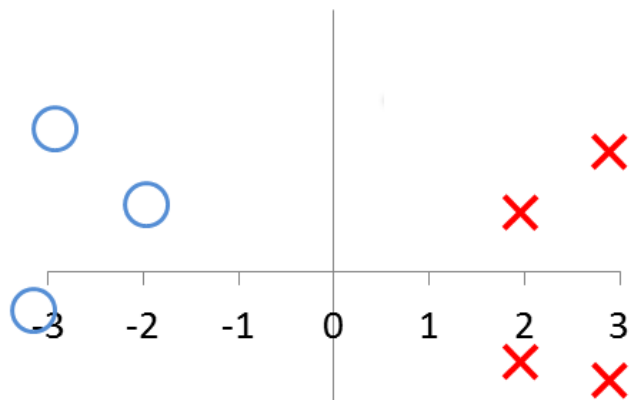


The SVM optimization problem we used is:

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

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

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



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

☐ 1/4

☒ 1/2

Correct Response

☐ 1

☐ 2



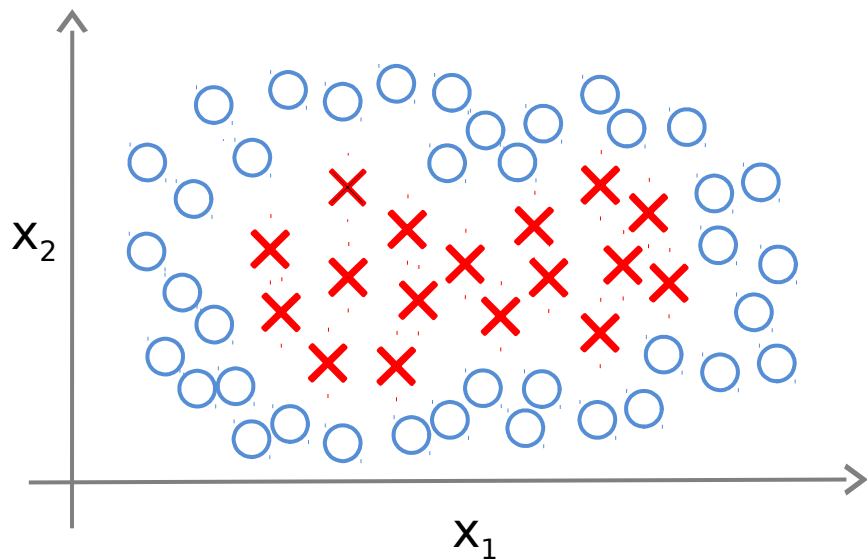


Machine Learning

# Support Vector Machines Kernels I

---

# Non-linear Decision Boundary



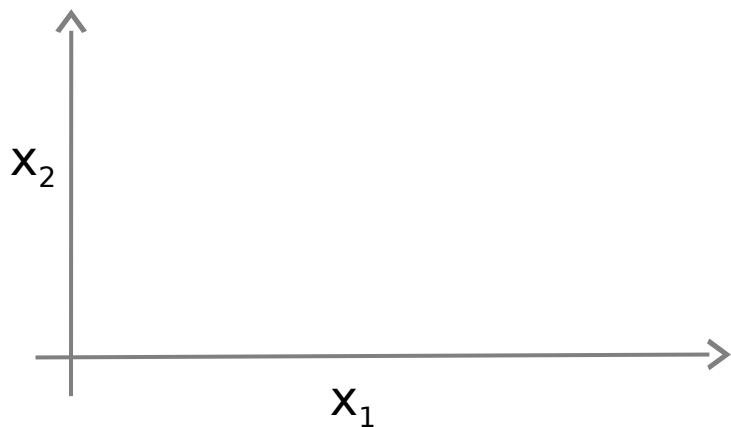
Predict  $y = 1$  if

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

Is there a different / better choice of the features, ...  
?

# Kernel

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



# Kernels and Similarity

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

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

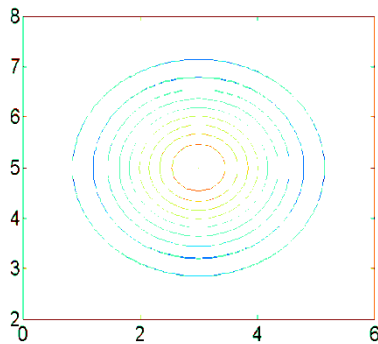
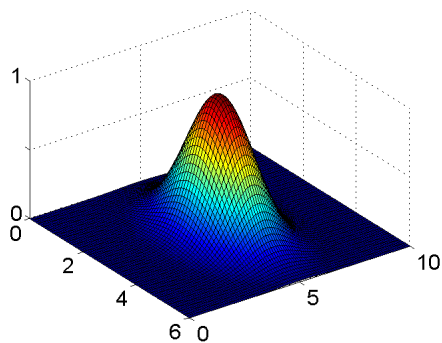
If  $x$  is far from  $l^{(1)}$  :

## Example:

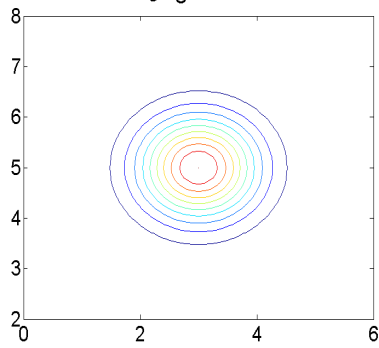
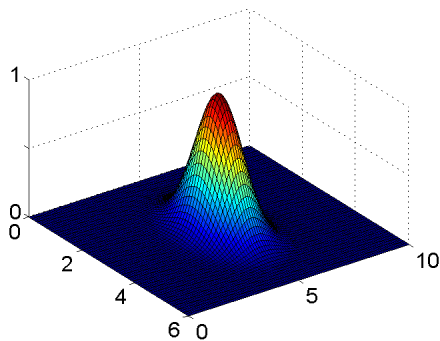
$$l^{(1)} = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad f_1 = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$f_1$  will be maximum i.e. 1,  
when  $x_1 = 3$  and  $x_2 = 5$

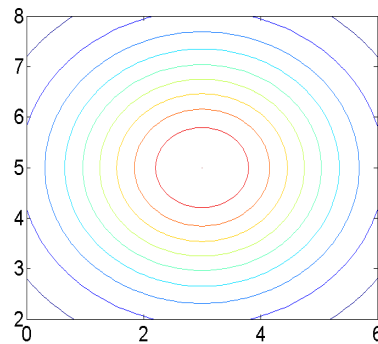
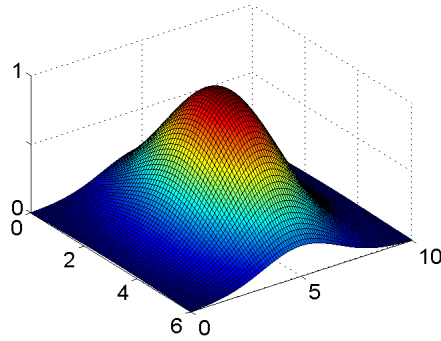
$$\sigma^2 = 1$$

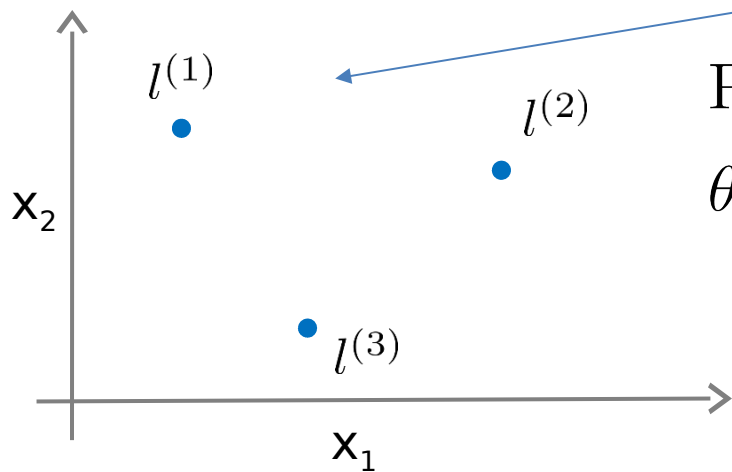


$$\sigma^2 = 0.5$$



$$\sigma^2 = 3$$



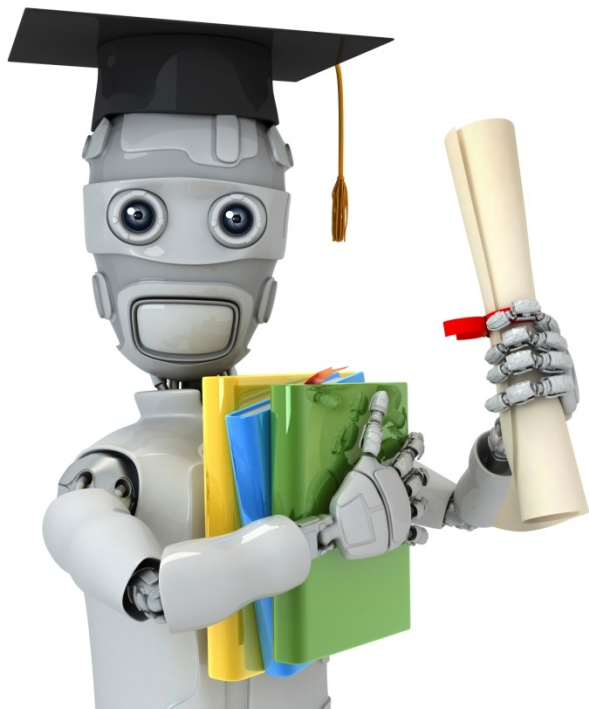


For points near  $l^{(1)}$ ,  $l^{(2)}$  (since  $\theta_1 = \theta_2 = 1$ ),  
We Predict  $y = 1$  otherwise we predict  $y = 0$

Predict "1" when

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

$x$  is close to  $l^{(1)}$  and far  
away from  $l^{(2)}$  and  $l^{(3)}$



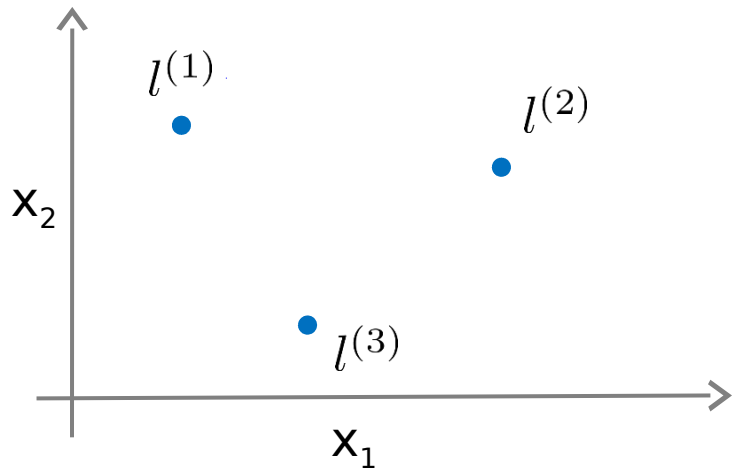
Machine Learning

# Support Vector Machines

---

## Kernels II

# Choosing the landmarks



Given  $x$  :

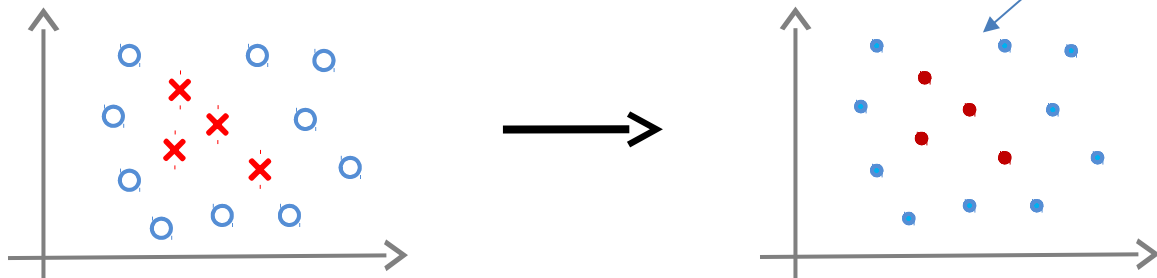
$$f_i = \text{similarity}(x, l^{(i)})$$

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

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

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

landmarks are chosen exactly same as the training example





## 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 :

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

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

...

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

# SVM with Kernels

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

Predict “ $y=1$ ” iff  $\theta^T f \geq 0$

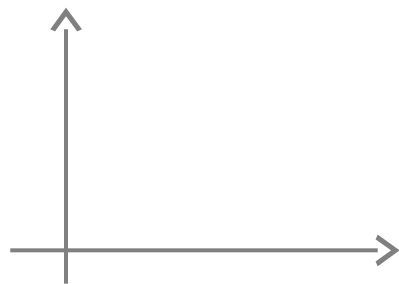
Training:

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

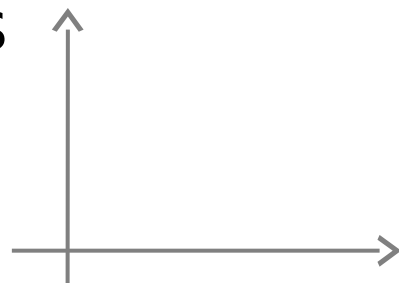
## SVM parameters:

$C (= \frac{1}{\lambda})$ . Large C: Lower bias, high variance.  
Small C: Higher bias, low variance.

Large  $\sigma^2$  : Features vary more smoothly.  
Higher bias, lower variance.



Small  $\sigma^2$  : Features vary less smoothly.  
Lower bias, higher variance.



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  $C$

Correct Response

☒ Decrease  $C$

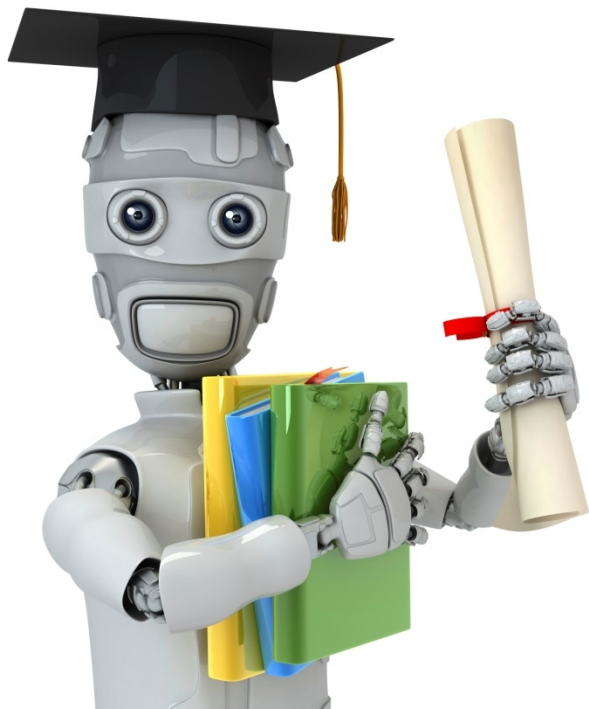
Correct Response

☒ Increase  $\sigma^2$

Correct Response

☐ Decrease  $\sigma^2$

Correct Response



Machine Learning

# Support Vector Machines

---

## Using an SVM

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters  $\theta$ .

Need to specify:

- Choice of parameter C.

- Choice of kernel (similarity function):

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

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

Gaussian kernel:

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

$l^{(i)}$ , where

Need to choose  $\sigma^2$ .

## Kernel (similarity) functions:

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

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

```
return
```

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

## 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:
- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, ...



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

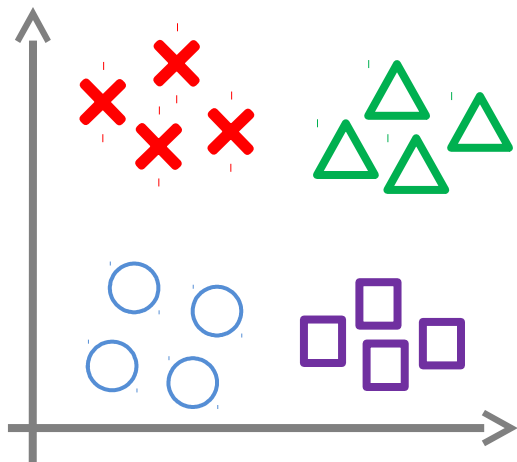
---

- ☐ Choose whatever performs best on the training data.
- ☒ Choose whatever performs best on the cross-validation data.

**Correct Response**

- ☐ Choose whatever performs best on the test data.
- ☐ Choose whatever gives the largest SVM margin.

# 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  $i = 1, 2, \dots, K$ ). For each test point  $x$ , get  $\theta^{(1)}, \theta^{(2)}, \dots, \theta^{(K)}$ . Pick class with largest

# Logistic regression vs. SVMs

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

If  $m$  is large (relative to  $n$ ):

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

If  $n$  is small,  $m$  is intermediate:  
Use SVM with Gaussian kernel

If  $n$  is small,  $m$  is large:

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.