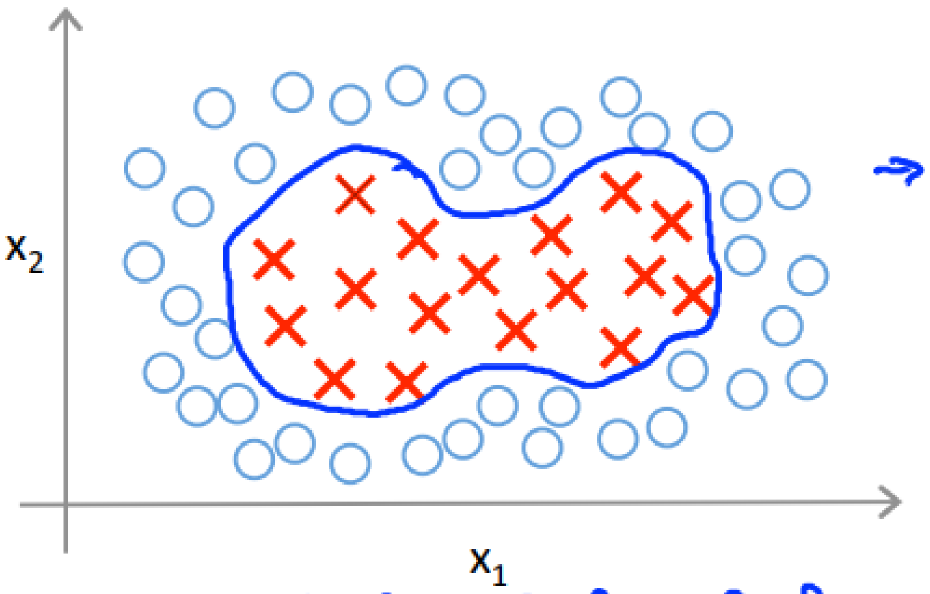
# Week 7:
# Support Vector Machines
# Kernels

# Non-linear Decision Boundary




Predict $y = 1$ if

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

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

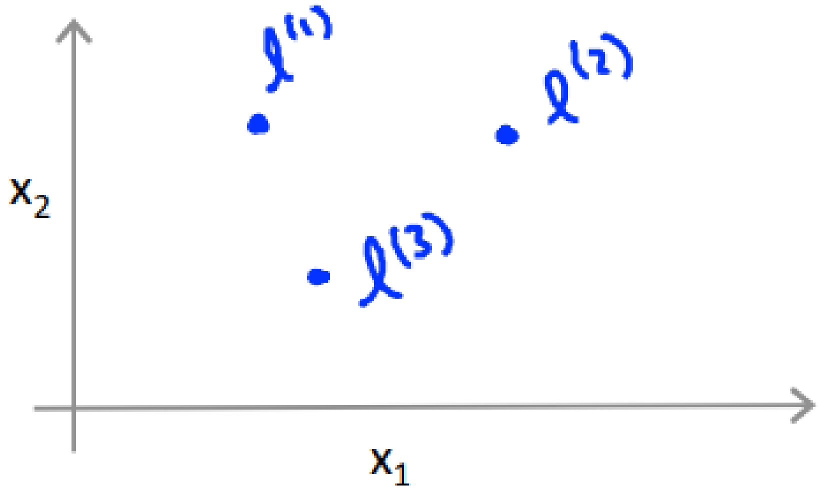
$$\rightarrow \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$$

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

High order polynomial features -> new features
to reduce computational expense

# Measure similarity using Euclidean distance squared

## Kernel




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

$\|w\|$

$\|x - l^{(1)}\|^2$

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

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

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

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

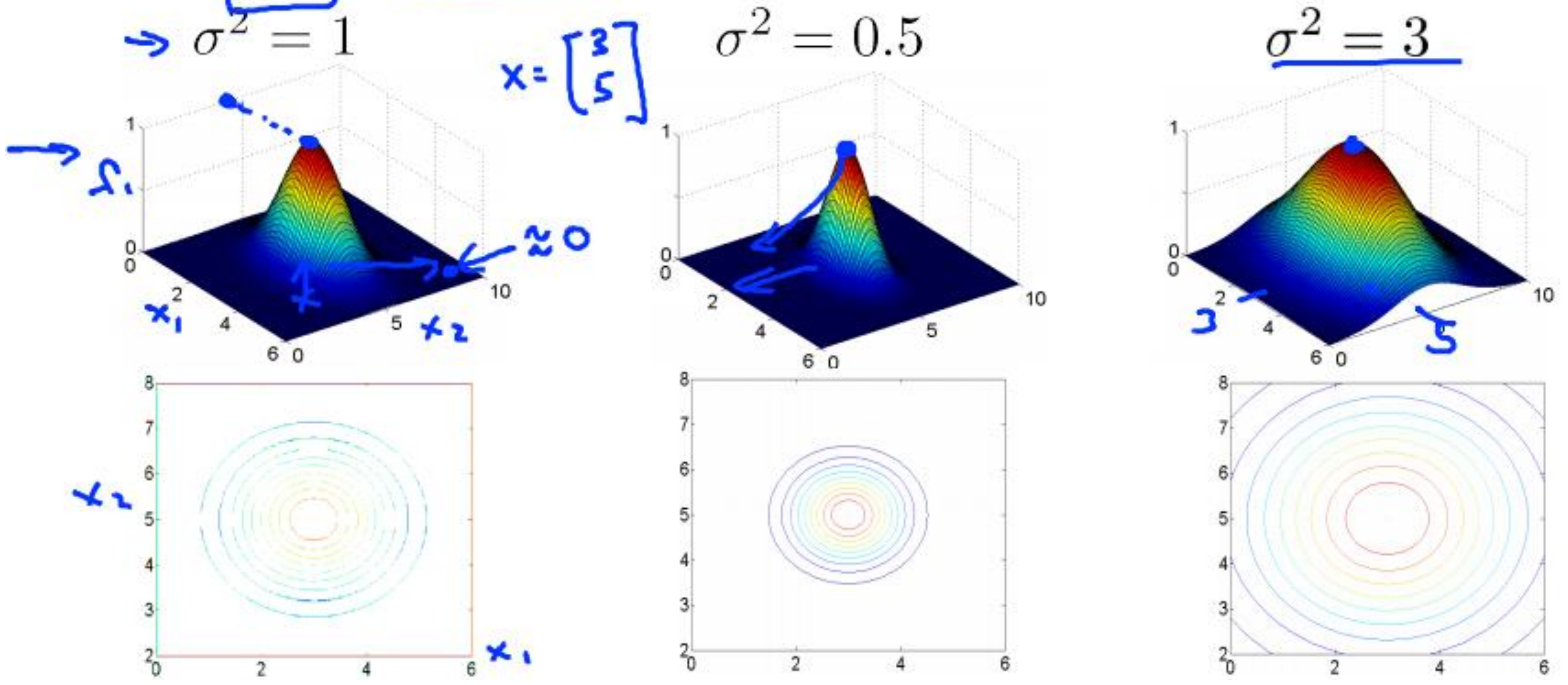
If $x$ if far from $l^{(1)}$ :

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

$$l^{(1)} \rightarrow f_1$$
$$l^{(2)} \rightarrow f_2$$
$$l^{(3)} \rightarrow f_3$$

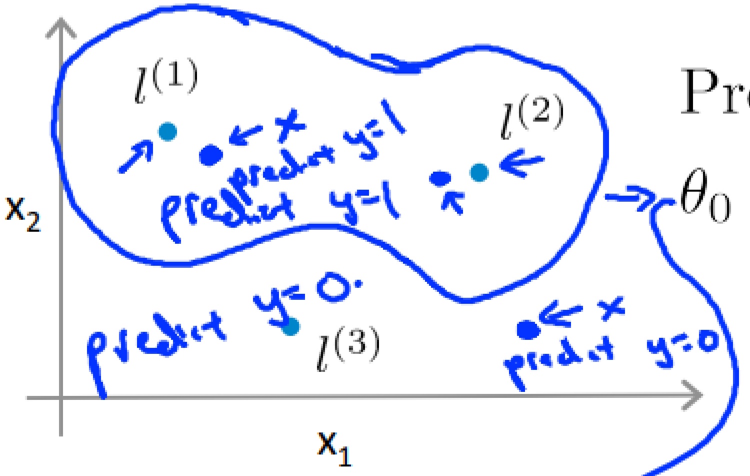
$x$

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




Gaussian kernel graph vary by sigma squared

# If training example is given




Predict "1" when

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

$x$

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

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

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

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

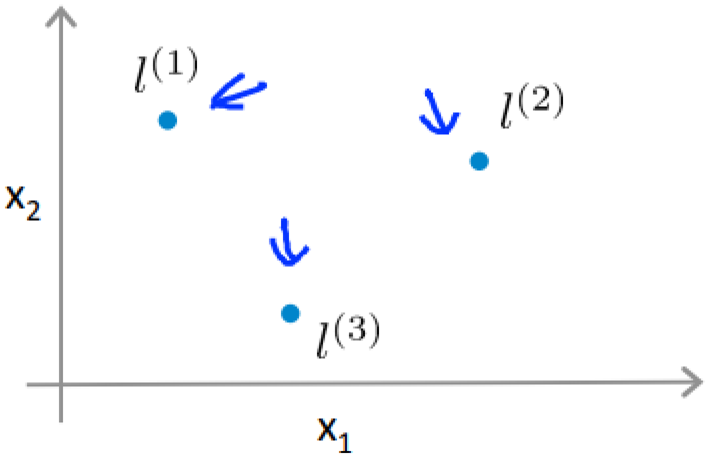
$f_1, f_2, f_3 \approx 0$

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

Inner boundary -> predict as 1

Outer boundary -> predict as 0

# Choosing landmarks

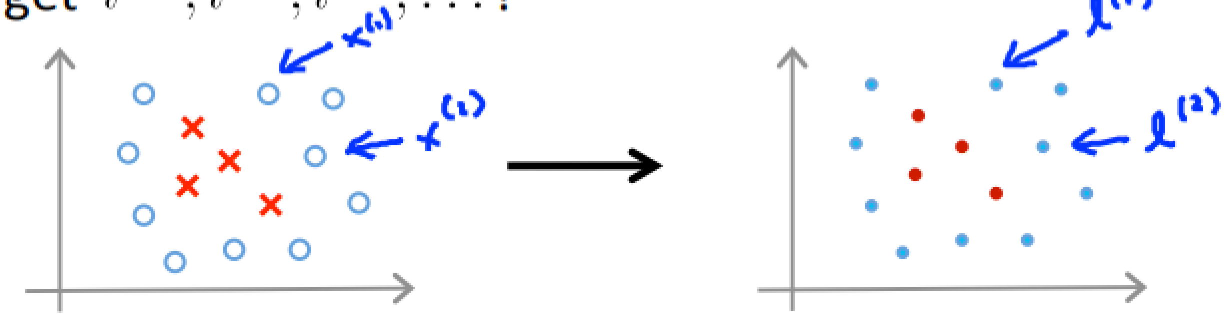



Given $x$:

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

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

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




$l^{(1)}$
$l^{(2)}$
$\vdots$
$l^{(m)}$

Set landmarks exact same location as training data

# Mapping training data with similarity function

Given example $x$: $\leftarrow x^{(i)}$

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

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

...

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

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

$$x^{(i)} \rightarrow \begin{array}{l} f_1^{(i)} = \text{sim}(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = \text{sim}(x^{(i)}, l^{(2)}) \\ \vdots \\ f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)}) \end{array}$$

$$f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp\left(-\frac{0}{2\sigma^2}\right) = 1 \qquad (x^{(i)} = l^{(i)})$$

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

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

Andrew N

M-dimensional x vector -> (M+1)-dimensional feature vector
* f0 = 1

# How to get value of theta

**SVM with Kernels**

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

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

Training:

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

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

To get value of theta and minimize it, we can use cost-function

## Why not use kernel in other algorithm?

- We ‘can’ use kernel in other algorithms such as logistic regression
- But computational tricks doesn’t generalize to other algorithms
- As a result, computation will be very slow and expensive

# Bias variance trade-off

$$\min_{\theta} C \sum_{i=1}^{m} [y^{(i)} (Cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) Cost_0(\theta^T f^{(i)})] + \frac{1}{2} \sum_{i=1}^{n} \theta_j^2$$

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

# Bias variance trade-off

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

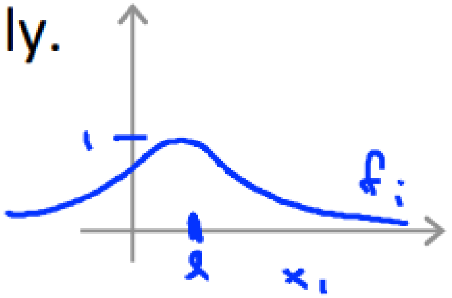
→ Small C: Higher bias, low variance. (large $\lambda$)

$\sigma^2$

Large $\sigma^2$: Features $f_i$ vary more smoothly.

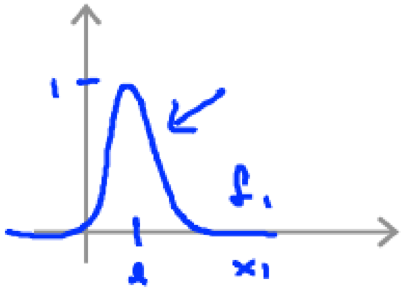
→ Higher bias, lower variance.

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




Small $\sigma^2$: Features $f_i$ vary less smoothly.

Lower bias, higher variance.




Large C = small lambda = tendency to overfit

Small C = large lambda = tendency to underfit

Small sigma = tendency to overfit

Large sigma = tendency to underfit