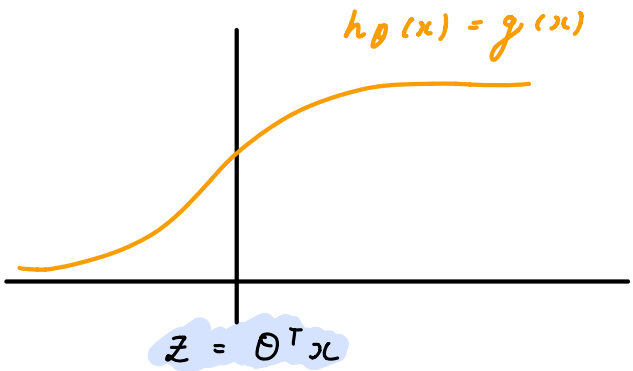
Stanford ML

WEEK 7:

Support Vector Machines (SVMs)

- Modifying logistic regression

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



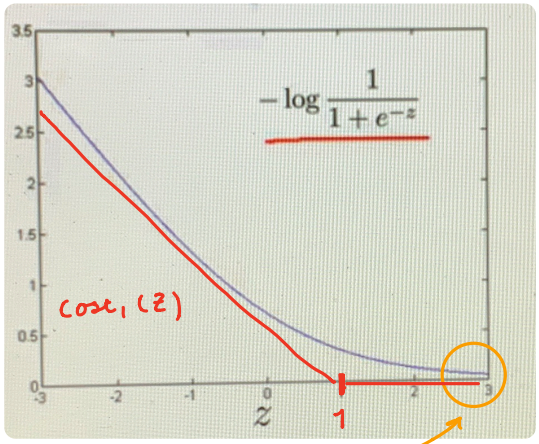
- If $y = 1$, we want $h_\theta(x) \approx 1$ and so $\theta^T x >> 0$
- If $y = 0$, we want $h_\theta(x) \approx 0$ and so $\theta^T x << 0$

COST OF SINGLE EXAMPLE

$$-(y \log h_\theta(x) + (1-y) \log(1 - h_\theta(x))$$

$$\Rightarrow -y \log\left\{\frac{1}{1+e^{-z}}\right\} - (1-y) \log\left(1 - \frac{1}{1+e^{-z}}\right)$$

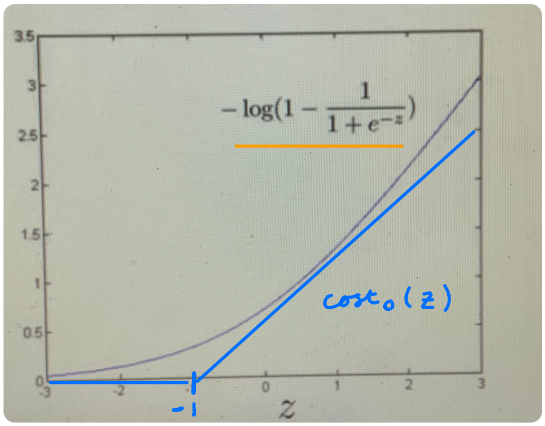
- If $y = 1$ (want $z >> 0$)



large z gives a small value for $-\log\left(\frac{1}{1+e^{-z}}\right)$

- LEFT ONLY WITH 1ST TERM
- Created new cost function consisting of 2 line segments
- This creates computational advantages and allows for a more efficient algorithm

- If $y = 0$ (want $z << 0$):



- LEFT ONLY WITH 2nd TERM
- Similar logic and reasoning applies

COMPARISON:

LOG. REG.

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

SVM

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

↳ Can remove the $1/m$ factor and receive the same parameters via minimization

* Instead of using λ:

LOGREG: $\min\ A + \lambda B$; SVM: $\min\ CA + B$

Different way of modifying training fit / generalization trade-off

SVM COST FUNC:

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

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

HYPOTHESIS

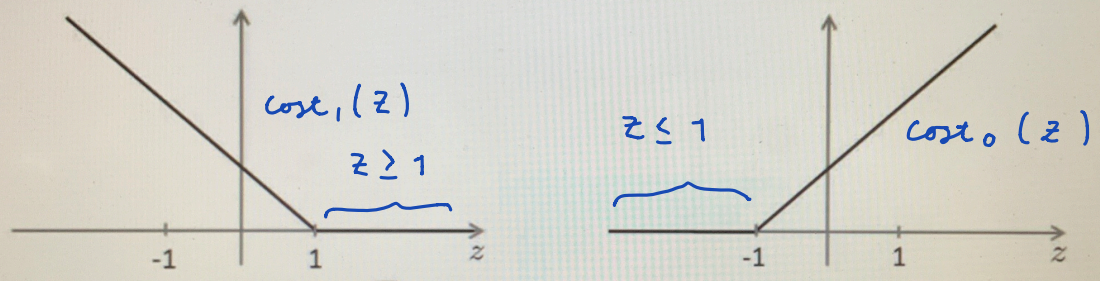
- Rather than the hypothesis outputting a probability of an example being positive ($y=1$), the SVM will output 1 or 0 directly

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

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)

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

Builds in extra safety factor for correct classification

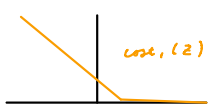
- Let $C \sim 10^5$

$$\min_\theta C \underbrace{\sum_{i=1}^{m} \left\{ y^{(i)} cost_1(z) + (1 - y^{(i)}) cost_0(z^{(i)}) \right\}}_{\sim 0} + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

- For large C, we want the $\{\}$ term to be ~ 0 to minimize the cost function

$\Rightarrow$ Whenever $y^{(i)} = 1$; we need $\theta^T x^{(i)} \geq 1$

$\Rightarrow$ Whenever $y^{(i)} = 0$; we need $\theta^T x^{(i)} \leq -1$



- The SVM optimization problem boils down to:

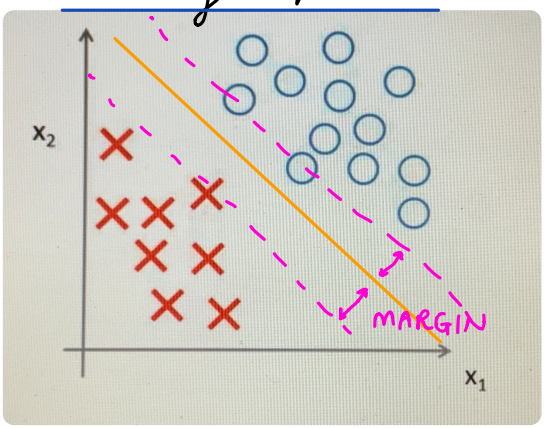
$$\therefore \min_{\theta} \quad C \times 0 + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

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

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

DECISION BOUNDARY

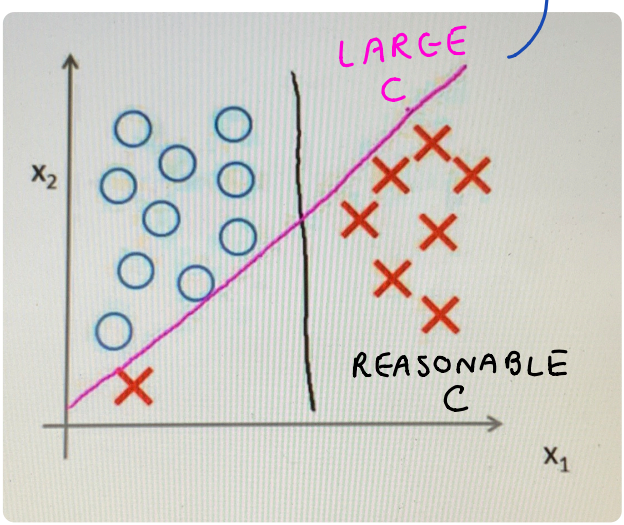
Linearly Separable



LARGE MARGIN CLASSIFIER

- The SVM will create a decision boundary — which maximizes the minimum distance to the +VE and -VE training example
- MAXIMIZES THE MARGINS

OUTLIER PRESENCE



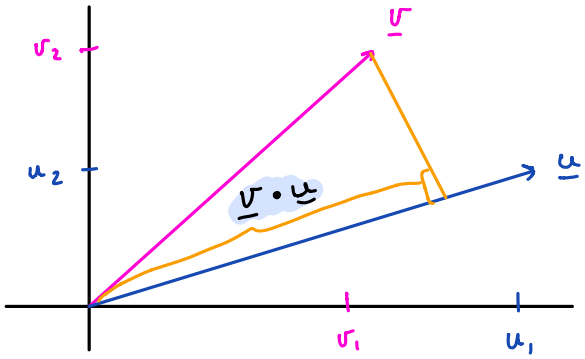
$C \sim 10^5$

— : no outlier

— : with outlier

- Boundaries created using ONLY large margin classifier
- If C is NOT TOO LARGE, we remain with the BLACK decision boundary

Mathematics of L.M Classifier



- p = length of projection of v onto u

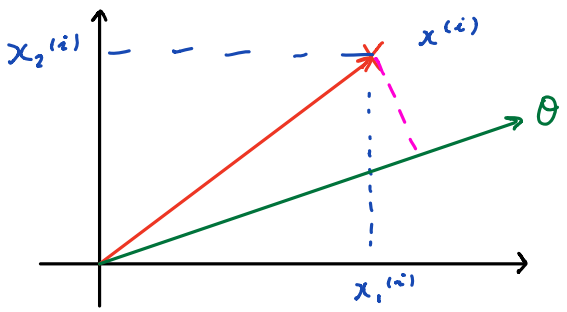
$$u^T v = p \cdot \|u\| = u_1 v_1 + u_2 v_2 = \underline{u} \cdot \underline{v}$$

SVM DECISION BOUNDARY

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

- Simplification: $\theta_0 = 0$ and $n = 2$ → decision boundary must pass through origin.

$$\therefore \min_{\theta} \frac{1}{2} (\theta_1^2 + \theta_2^2) = \frac{1}{2} \|\theta\|^2$$

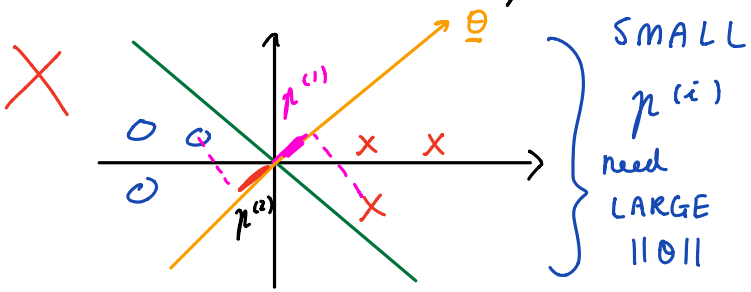


∴ SVM decision boundary is created by minimizing the length squared of the parameter vector.

$$\therefore \theta^T x^{(i)} = p \cdot \|\theta\| = \underline{x} \cdot \underline{\theta} = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

$$\Rightarrow \min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 \quad \text{s.t} \quad \begin{cases} p^{(i)} \cdot \|\theta\| \geq 1 & \text{if } y^{(i)} = 1 \\ p^{(i)} \cdot \|\theta\| \leq -1 & \text{if } y^{(i)} = 0 \end{cases}$$

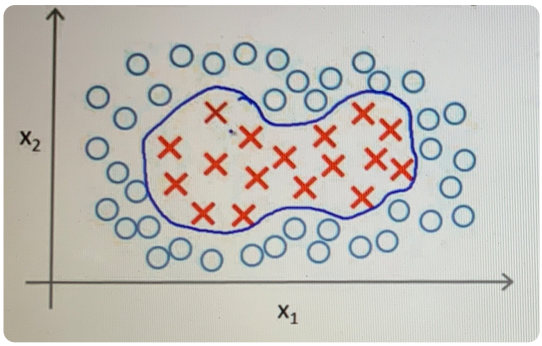
where $p^{(i)}$ is the projection of training example $x^{(i)}$ onto the parameter vector $\underline{\theta}$.



Kernels I

- Developing NON-LINEAR decision boundaries

NON-LINEAR DECISION BOUNDARY



- Predict $y = 1$ if

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

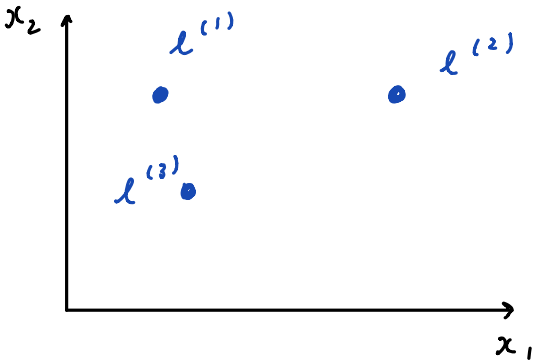
$$\Rightarrow h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 \cdots \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

- Let $f_1 = x_1$, $f_2 = x_2$,

$f_3 = x_1 x_2$ etc.

- Is there a better choice of features $\{f_i\}$?

KERNEL



- Given x, compute new features based off proximity landmarks $l^{(1)}, l^{(2)}, l^{(3)}$

- Given x: $f_1 = \text{SIMILARITY}(x, l^{(1)})$

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

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

$$\| x - l^{(1)} \|^2 \sim 0$$

$$\Rightarrow f_1 \approx e^{-0/2\sigma^2} \approx 1$$

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

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

$$\Rightarrow f_1 \approx e^{-\text{LARGE}/2\sigma^2} \approx 0$$

$\therefore f_{(i)} \in [0, 1]$ ℝ

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

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

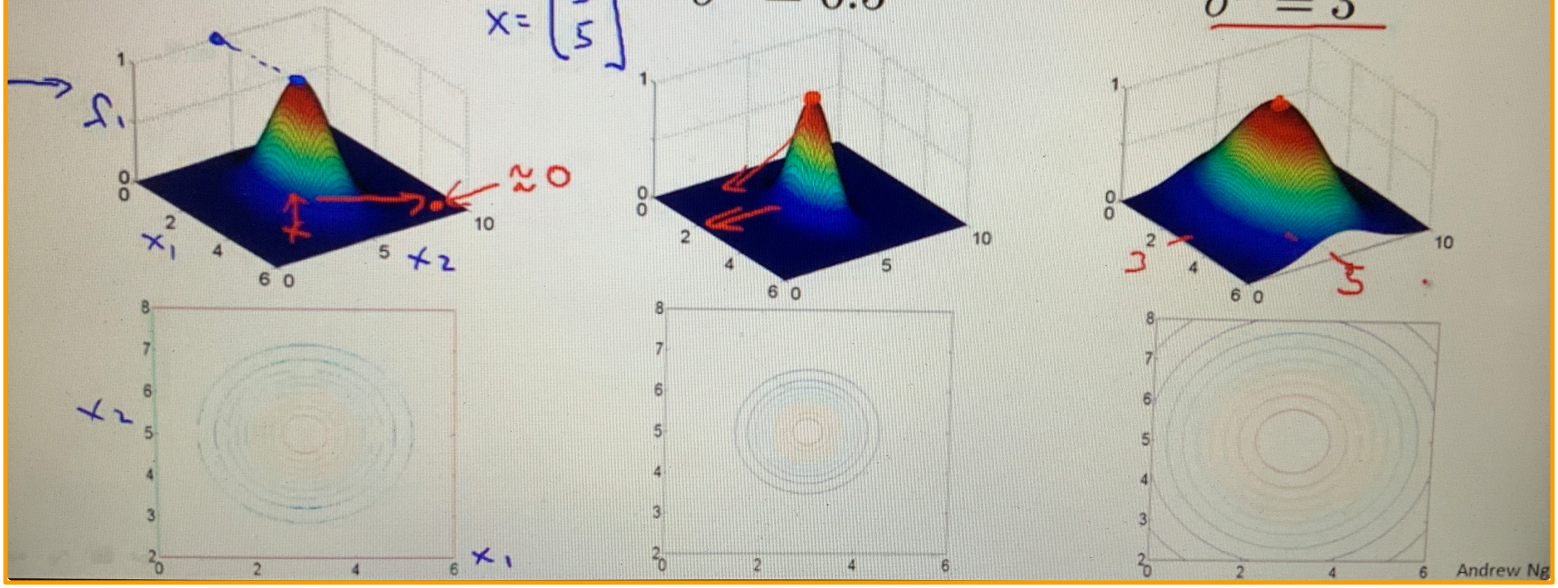
GAUSSIAN KERNELS

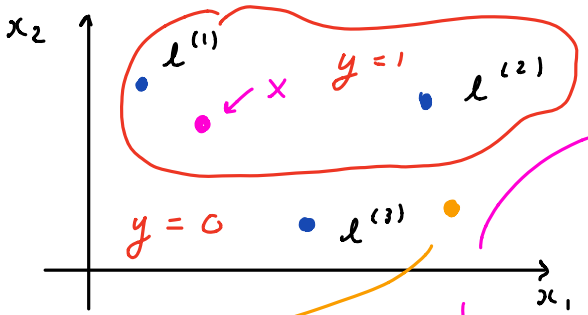
$k(x, l^{(i)})$

Example:

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

$\sigma^2 = 1$ $\quad x = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$ $\quad \sigma^2 = 0.5$ $\quad \sigma^2 = 3$





* Predict "1" when $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$
* Through learning we receive $\theta_0 = 0.5$, $\theta_1 = 1.0$, $\theta_2 = 1$, $\theta_3 = 0$

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

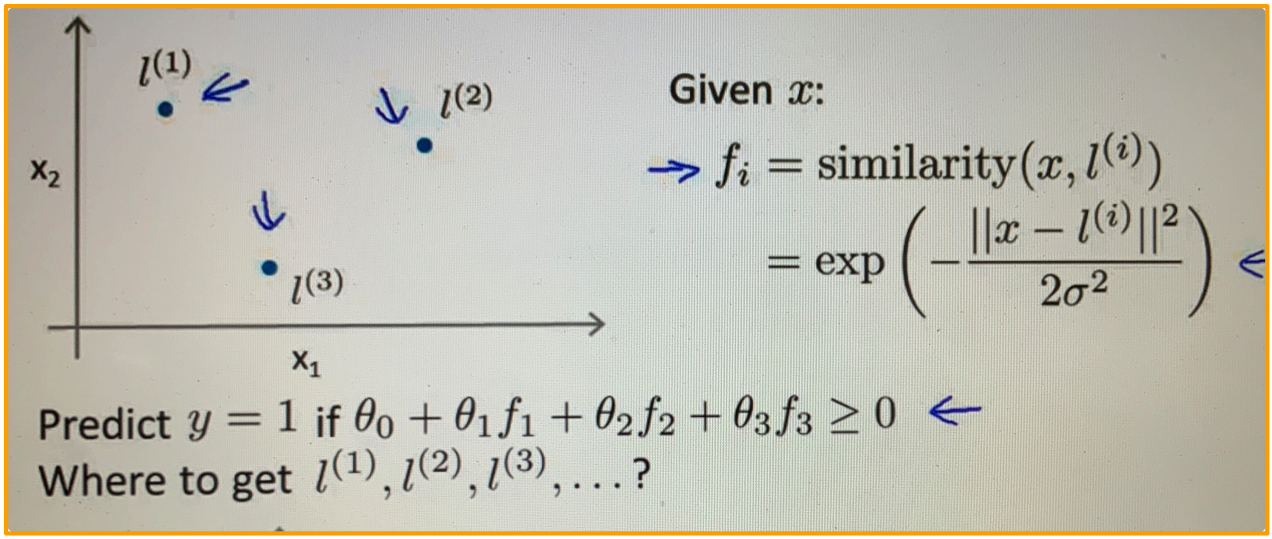
$\Rightarrow -0.5 + 1 = 0.5 \geq 0$ PREDICT 1

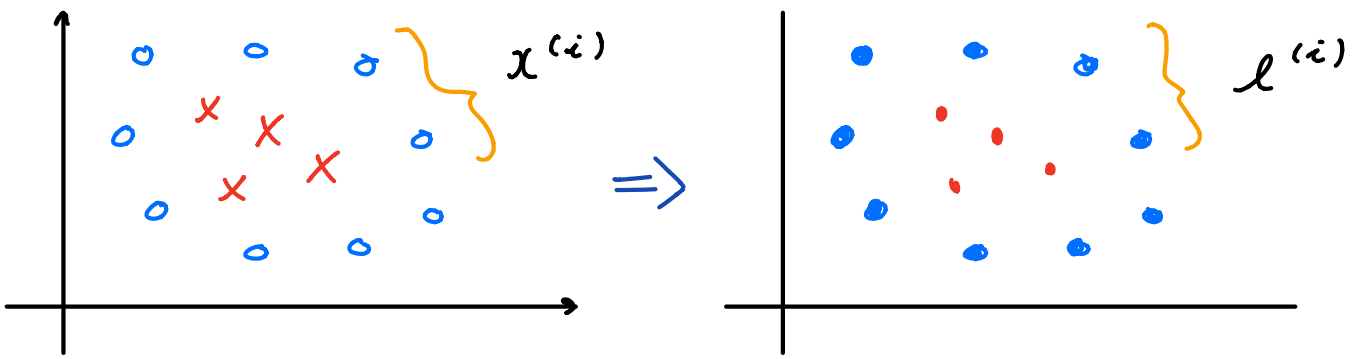
$f_1, f_2, f_3 \approx 0 \Rightarrow \theta_0 + \theta_1 f_1 + \dots \approx -0.5 \leq 0$ PREDICT 0

- Overall, points near $l^{(1)}$ and $l^{(2)}$ will end up being predicted as 1 whilst ones far away will be 0.
- Allows a complex non-linear boundary to be formed

Kernels II

CHOOSING LANDMARKS





- Put landmarks in the exact same spots as the training examples

→ Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$

→ Choose $l^{(i)} = x^{(i)} \quad \forall\, i = 1, \ldots, m$

→ COMPUTE: $f_i = k(x, l^{(i)})$ → $\begin{bmatrix} f_0 = 1 \\ f_1 \\ \vdots \\ f_m \end{bmatrix}$

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

$x^{(i)} \longrightarrow$

$f_1^{(i)} = k(x^{(i)}, l^{(1)})$

$f_2^{(i)} = k(x^{(i)}, l^{(2)})$

$\vdots \quad \leftarrow f_i^{(i)} = k(x^{(i)}, l^{(i)}) = 1$

$f_m^{(i)} = k(x^{(i)}, l^{(m)})$

∴ $n = m$

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

SVMs w/ Kernels

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

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

* TRAINING:

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

θ_0 not regularized

SVM Parameters:

$C \; (= 1/\lambda)$

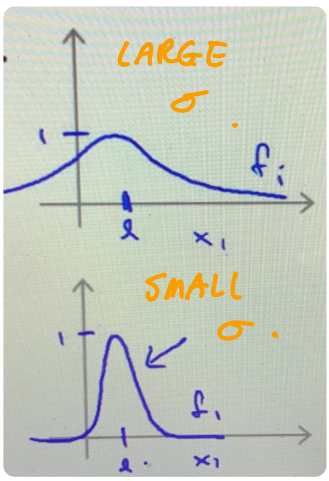
→ Large C: lower bias, high variance (OVER)

→ Small C: higher bias, low variance (UNDER)

σ^2

→ Large σ^2: Features f_i vary more smoothly → higher bias, low variance

→ Small σ^2: Features f_i vary less smoothly → lower bias, higher variance



Using an SVM

* Use SVM software package to solve for parameters θ

* NEED TO SPECIFY:
  - Choice of parameter C
  - Choice of kernel

e.g. NO KERNEL ("linear")

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

- if n = LARGE
- if m = SMALL

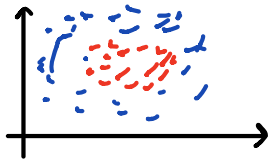
$x \in \mathbb{R}^{n+1}$

↳ Gaussian Kernel

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

where $l^{(i)} = x^{(i)}$ } NEED TO CHOOSE σ^2

- $x \in \mathbb{R}^n$; n = SMALL
- and / or m = LARGE



Kernel (similarity) functions:

$x^{(i)}$ $l^{(i)} = x^{(i)}$

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

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

$f_i \in \mathbb{R}$

$x \longrightarrow f_1, f_2, \vdots, f_m$

```
return
```

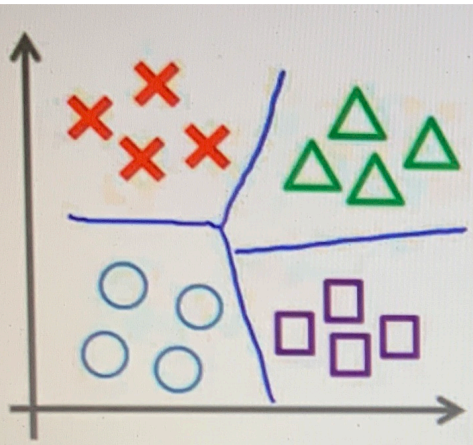
Note: Do perform feature scaling before using the Gaussian kernel.

MUST FEATURE SCALE BEFORE USING KERNEL

TO ENSURE $||x - l||^2$ IS NOT SKEWED

- All kernels must satisfy "Mercer's Theorem" for optimisation reasons

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

$y=1$ $y=2$ $\ldots$ $\theta=K$

Logistic Regression vs. SVMs

* n = # of features, m = training set size
* if n is large (relative to m ~ x10) $n \geq m$
* USE LOG REG, or SVM with LINEAR KERNEL

* if n is small and m intermediate
  (n = 1-1000, m = 10, - 10,000)
  → Use SVM with Gaussian kernel

* if n is small and m LARGE
  → Create / add more features, then use LOG REG or SVM w/ NO KERNEL
  (n = 1-1000, m = 50,000+)

CONVEX OPTIMISATION PROBLEM w/ SVM

- ANN MAY WORK WELL BUT SLOWER TO TRAIN