Learn with Google AI | Explore ML

Microsoft Student Partner (MSP)

ML Summit Workshop Day 3

By

Shatadru Majumdar

and

Purbayan Chowdhury

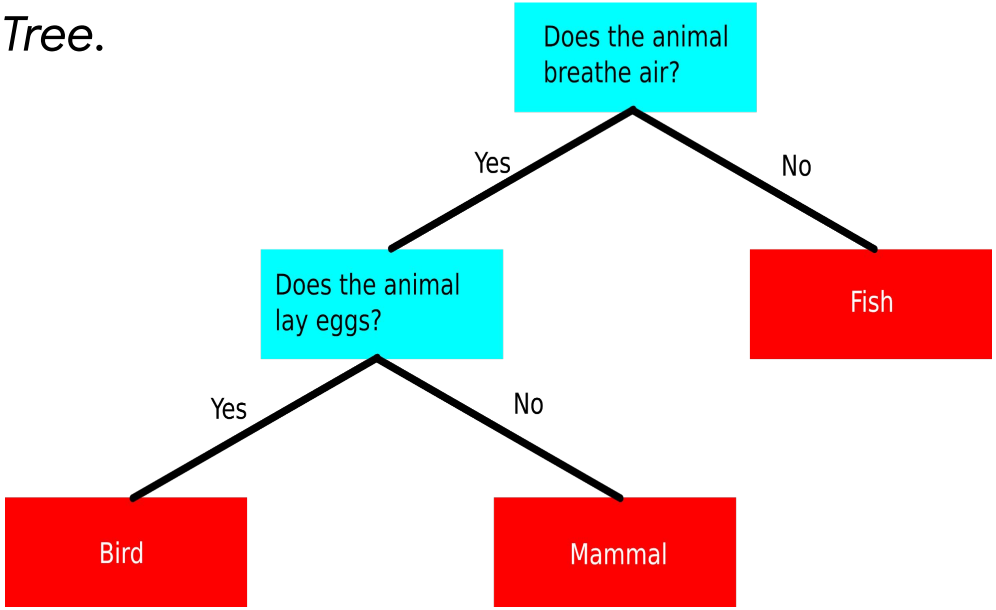
INNOVACIÓN
EMPOWERED BY INNOVATION
2020

What is a Decision Tree?

Predictive Model in tree form that maps items to its target value, starting from root to leaf is known as a *Decision Tree*.

Types of Decision Tree:

- **Classification Tree**
- **Regression Tree**



Entropy & Gini

Metrics for above mentioned algorithm can be:

- **Gini Impurity(GI):** Measure of how a randomly chosen element is incorrectly labelled if it was randomly labelled according to its subset distribution.
- **Information Entropy(IE):** Number of bits required in encoding the given data.

$$I_G = 1 - \sum_{j=1}^{c} p_j^2$$

p_j: proportion of the samples that belongs to class c for a particular node

GI

$$I_H = - \sum_{j=1}^{c} p_j log_2(p_j)$$

p_j: proportion of the samples that belongs to class c for a particular node.

IE

Information Gain

- **Information Gain** is used to decide which features to split at each step while building the tree.
- Our objective is to keep the tree as small as possible, thus we choose the split that results in the purest daughter nodes.
- The information value *represents the expected amount of information that would be needed to specify whether a new instance should be classified* ***yes*** *or* ***no****, given that the example reached that node*.

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{\mathrm{H}(T)}^{\text{Entropy (parent)}} - \overbrace{\mathrm{H}(T|a)}^{\text{Weighted Sum of Entropy (Children)}}$$

$$= -\sum_{i=1}^{J} p_i \log_2 p_i - \sum_a p(a) \sum_{i=1}^{J} -\Pr(i|a) \log_2 \Pr(i|a)$$

How does it work?

Look on the blackboard

Case Study

- Let’s consider the example where our objective is to figure out whether we should go out to play **Tennis** or not.

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Case Study

- First we determine the **information gain** for each candidate attribute(i.e, Outlook, Temperature, Humidity and Wind) and then select the one with **highest** information gain.
- **Attribute Outlook:**

  Outlook_Rain = {2+, 3-}

  Outlook_Sunny = {3+, 2-}

  Outlook_Overcast = {4+, 0-}

Gain(S,Outlook) = Entropy(S) — 5/14 Entropy(S_Outlook_Rain) — 5/14 Entropy(S_Outlook_Sunny) — 4/14 Entropy(S_Outlook_Overcast)

Gain(S,Outlook) = **0.246**

Case Study

- First we determine the **information gain** for each candidate attribute(i.e, Outlook, Temperature, Humidity and Wind) and then select the one with **highest** information gain.
- **Attribute Temperature:**

  Temperature_Hot = {2+, 2-}

  Temperature_Mild = {4+, 2-}

  Temperature_Cool = {3+, 1-}

Gain(S,Temperature) = Entropy(S) — 4/14 Entropy(S_Temperature_Hot) — 6/14 Entropy(S_Temperature_Mild) — 4/14 Entropy(S_Temperature_Cool)

Gain(S,Temperature) = **0.029**

Case Study

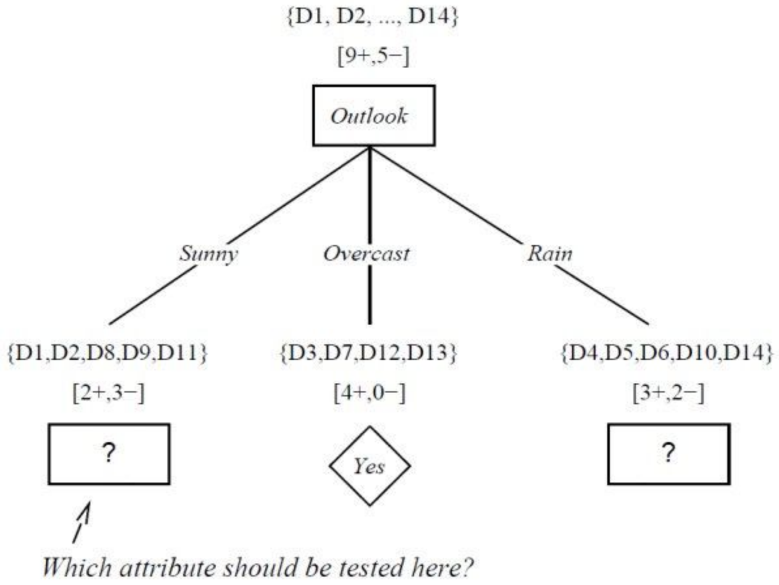
Similarly, the information gain for all the attributes are as follows :

1. Gain(S,Outlook) = **0.246**
2. Gain(S,Temperature) = **0.029**
3. Gain(S,Wind) = **0.048**
4. Gain(S,Humidity) = **0.151**

According to the information gain measure, the Outlook attribute provides the best prediction of the target attribute, PlayTennis, over the training examples. Therefore, Outlook is selected as the decision attribute for the root node, and branches are created below the root for each of its possible values (i.e., Sunny, Overcast, and Rain).

Case Study

The resulting partial decision tree is shown in Figure, along with the training examples sorted to each new descendant node.



Case Study

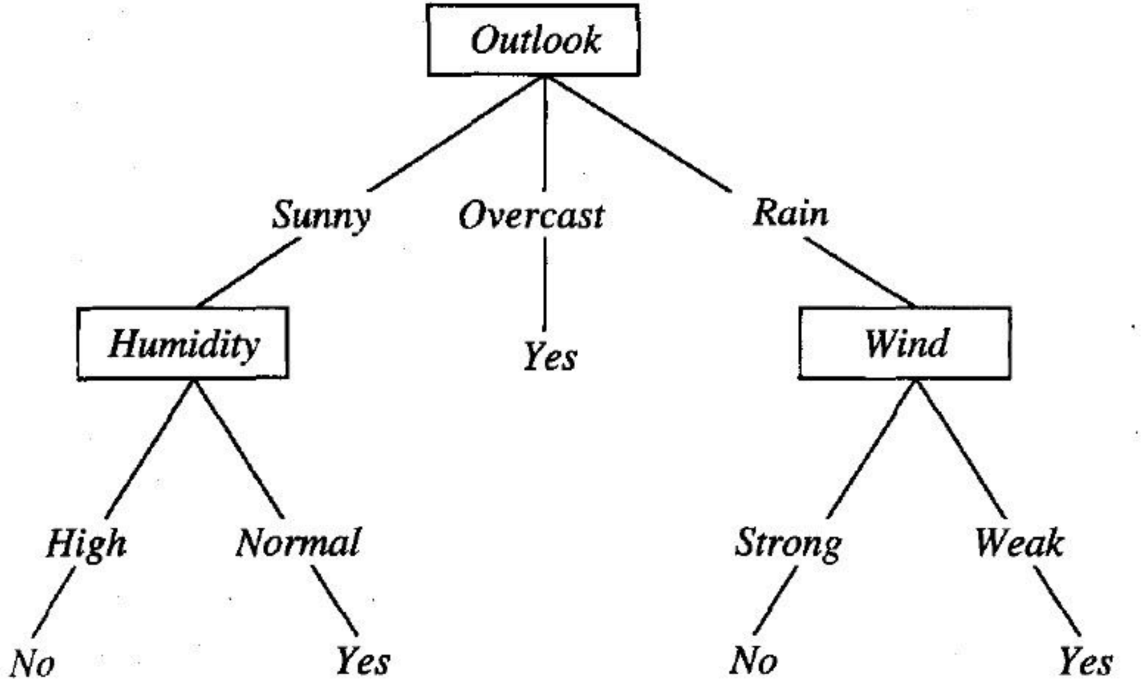
-> Note that every example for which Outlook = Overcast is also a positive example of PlayTennis. Therefore, this node of the tree becomes a leaf node with the classification PlayTennis = Yes. In contrast, the descendants corresponding to Outlook = Sunny and Outlook = Rain still have non-zero entropy, and the decision tree will be further elaborated below these nodes.

-> The process of selecting a new attribute and partitioning the training examples is now repeated for each non terminal descendant node, this time using only the training examples associated with that node.

-> Attributes that have been incorporated higher in the tree are excluded, so that any given attribute can appear at most once along any path through the tree. This process continues for each new leaf node until either of two conditions is met: (1) every attribute has already been included along this path through the tree, or (2) the training examples associated with this leaf node all have the same target attribute value (i.e., their entropy is zero).

Case Study

– Eventually, our Final decision tree turns out to be like this:



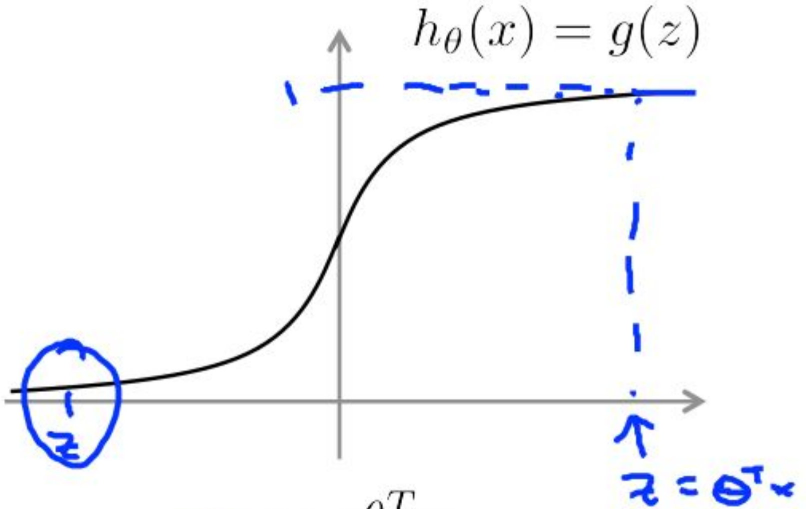
Notebook

What are Support Vector Machines?

- **Support Vector Machines** construct a hyperplane or set of hyperplanes in a high or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.
- An **SVM** model represents examples as points in space, mapped in such a way that the examples of the separate categories are divided by a clear gap as wide as possible.
- In addition to performing linear classification, **SVMs** can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Alternative view of logistic regression

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



$z = \theta^T x$

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

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

(x,y)

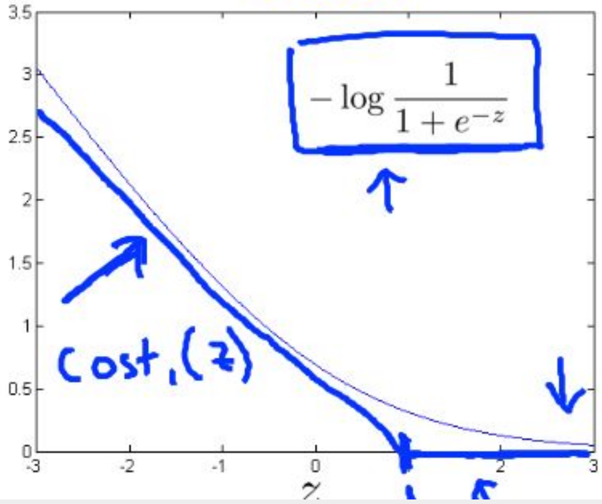
Alternative view of logistic regression

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

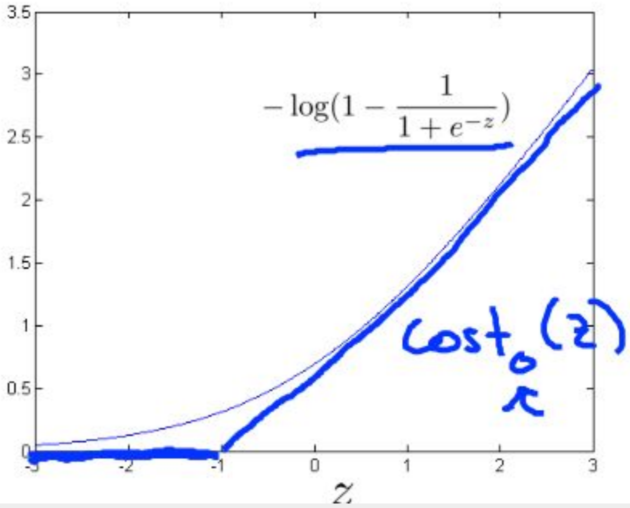
$$= \boxed{-y \log \frac{1}{1+e^{-\theta^T x}}} - \boxed{(1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})}$$

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

$z = \theta^T x$



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:

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

SVM hypothesis

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

SVM Decision Boundary

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

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

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

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

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

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

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

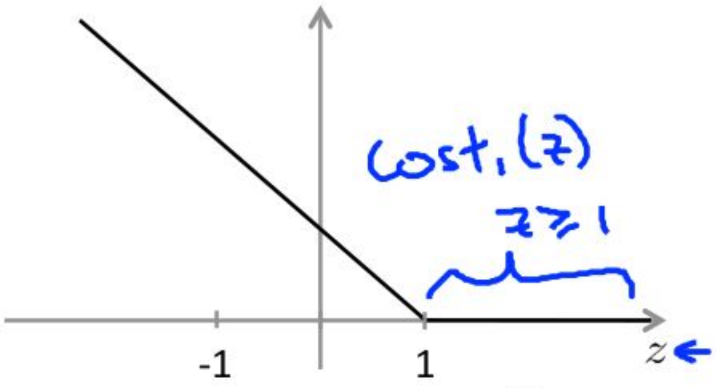
$\theta^T x^{(i)} \leq -1$

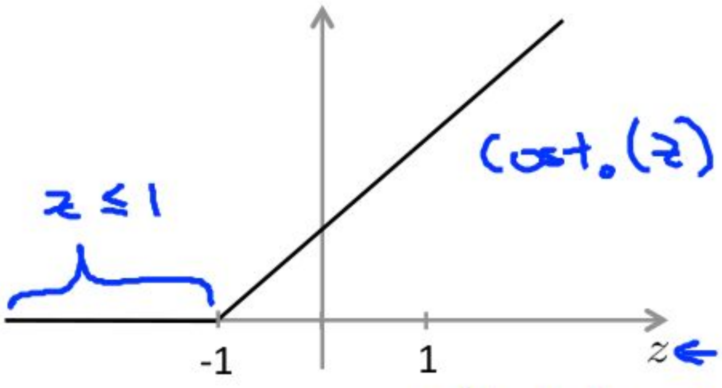
Hard Margin

- If the training data is linearly separable, we can select two parallel hyperplanes that separate the two classes of data, so that the distance between them is as large as possible. The region bounded by these two hyperplanes is called the "margin", and the maximum-margin hyperplane is the hyperplane that lies halfway between them.

Support Vector Machine

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

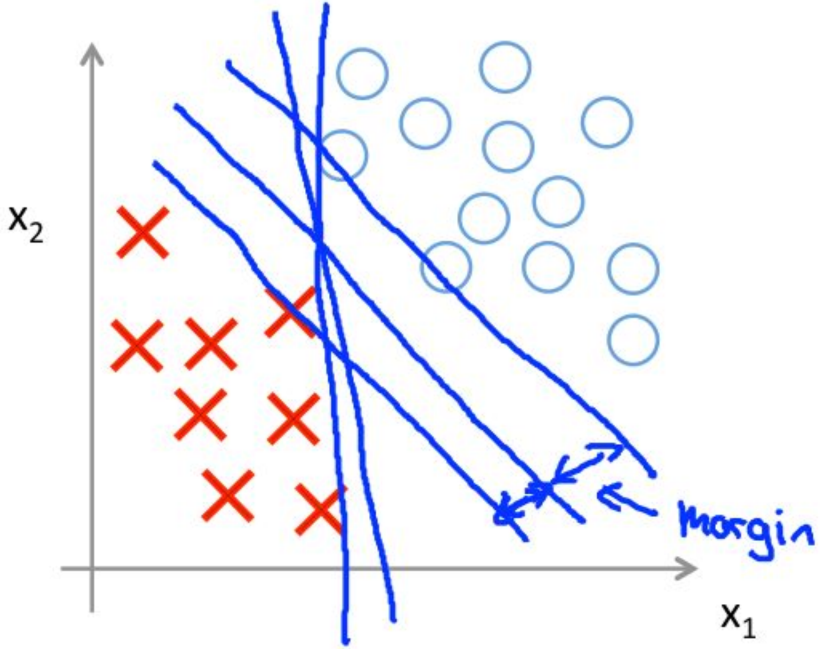




- If $\underline{y = 1}$, we want $\underline{\theta^T x \geq 1}$ (not just ≥ 0) $\quad$ $\theta^T x \geq \cancel{0}\ 1$
- If $\underline{y = 0}$, we want $\underline{\theta^T x \leq -1}$ (not just < 0) $\quad$ $\theta^T x \leq \cancel{0}\ -1$

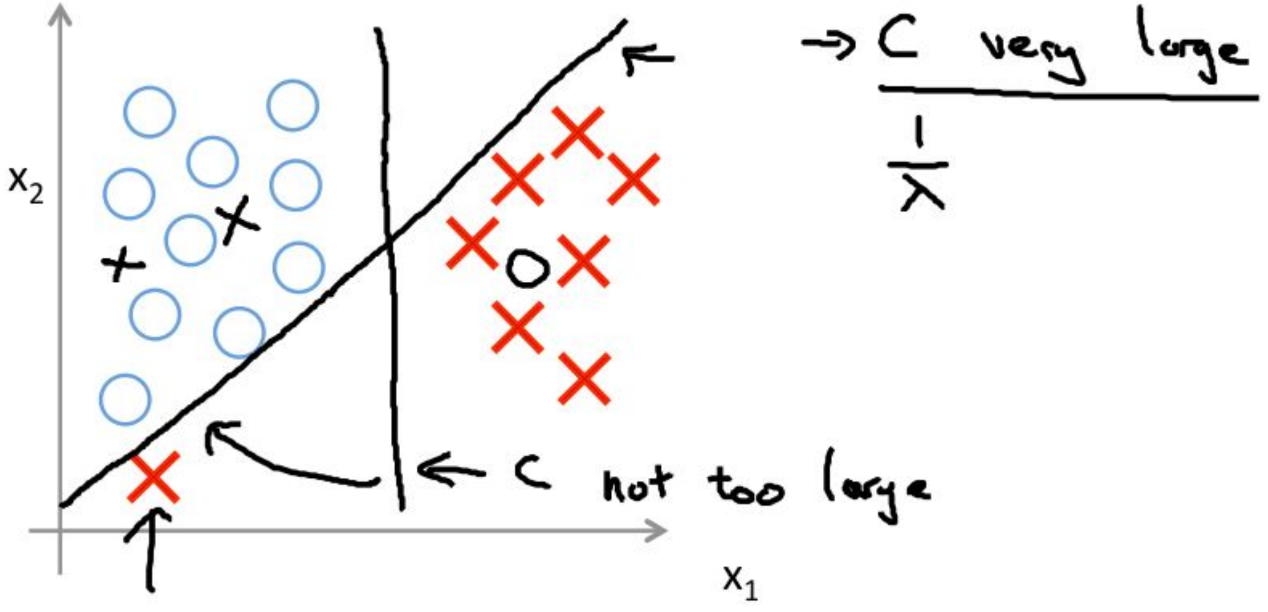
$C = 100,000$

SVM Decision Boundary: Linearly separable case



Large margin classifier

Large margin classifier in presence of outliers



Soft Margin

- To extend SVM to cases in which the data are not linearly separable, we introduce the **hinge loss** function,

$$\max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right).$$

- y_i is the i-th target and $\vec{w} \cdot \vec{x}_i - b$ is the current output.
- This function is zero if the constraint in (1) is satisfied, in other words, if vector x_i lies on the correct side of the margin.
- For data on the wrong side of the margin, the function's value is proportional to the distance from the margin.

Soft Margin

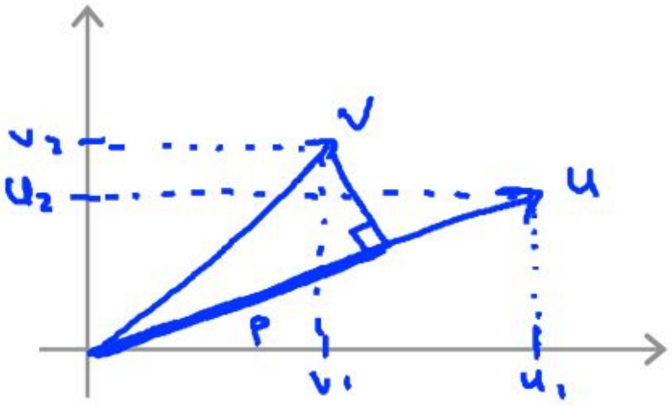
- We then wish to minimise:

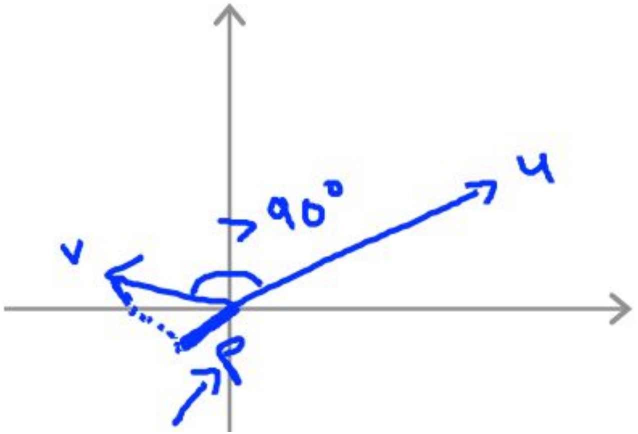
$$\left[\frac{1}{n}\sum_{i=1}^{n}\max\left(0, 1 - y_i(\vec{w}\cdot\vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2,$$

where the parameter λ determines the trade-off between increasing the margin size and ensuring that vector x_i lie on the correct side of the margin. Thus, for sufficiently small values of λ, the second term in the loss function will become negligible, hence, it will behave similar to the hard-margin SVM, if the input data is linearly classifiable.

Decision Boundary

Vector Inner Product





$$\rightarrow u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad \rightarrow v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$$u^T v = ? \qquad [u_1 \;\; u_2] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$\|u\|$ = length of vector u

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

p = length of projection of v onto u.

signed

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

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

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

$$p < 0$$

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

SVM Decision Boundary

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

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

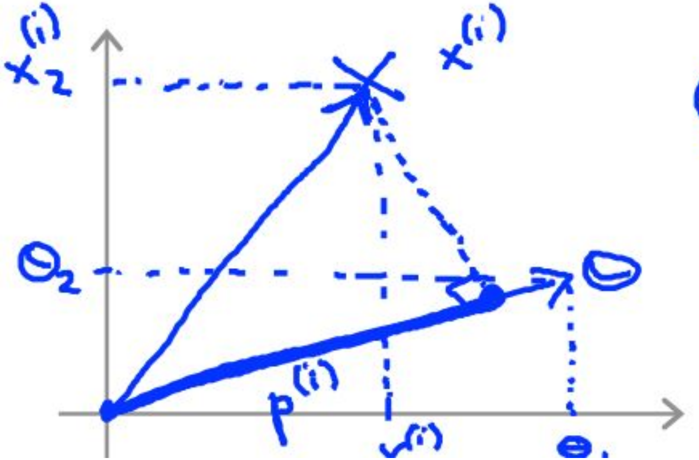
$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

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

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

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

$u^T v$



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

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

SVM Decision Boundary

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

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

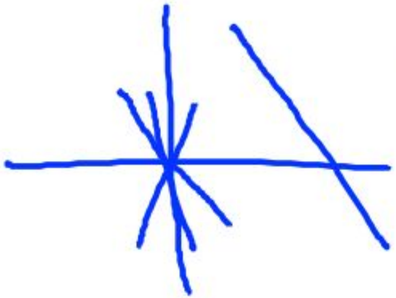
$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 1$

C very large

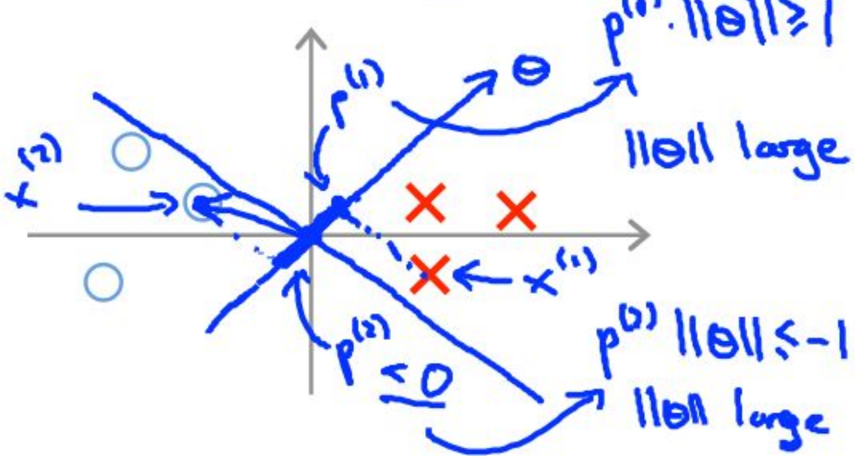
where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ.

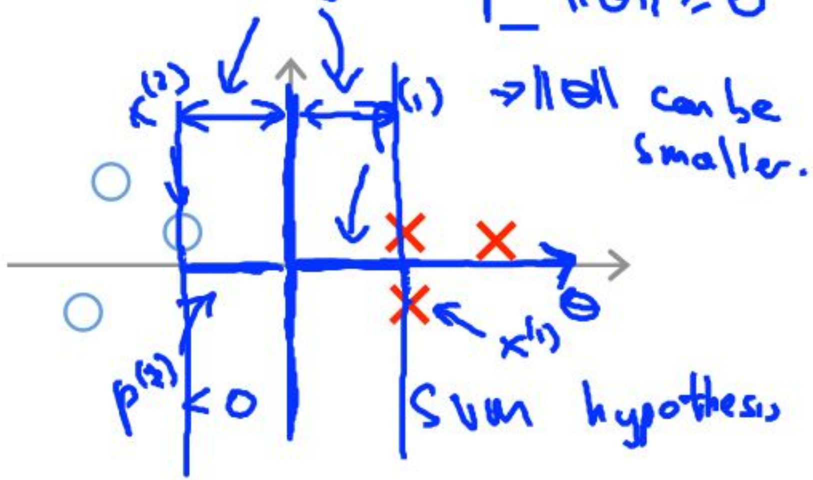
Simplification: $\theta_0 = 0$

$\theta_0 \neq 0$



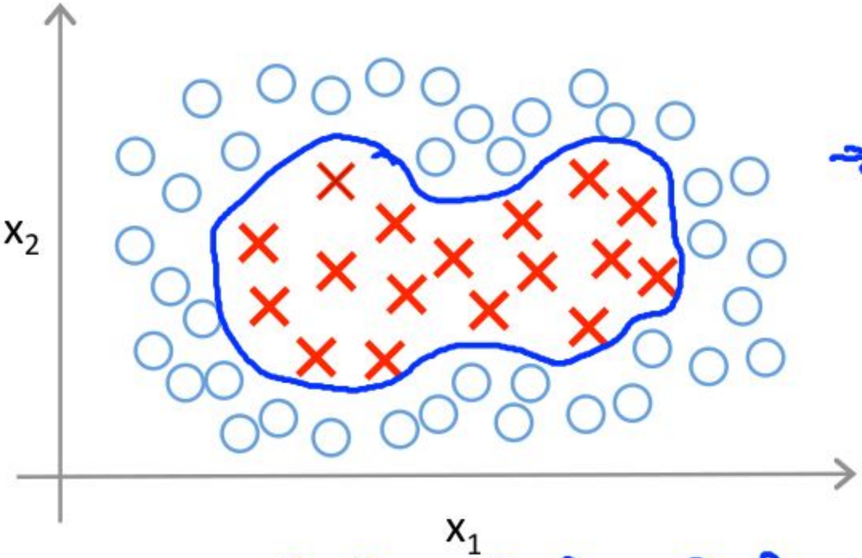
$p^{(1)} \|\theta\| \geq 0$





Kernels

Non-linear Decision Boundary



Predict $y = 1$ if

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

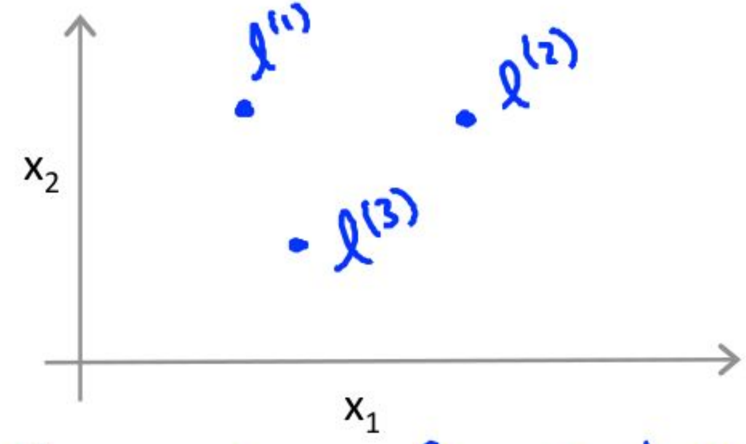
$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta_0 + \theta_1 x_1 + \dots \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

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

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

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 $l^{(1)}, l^{(2)}, l^{(3)}$

$\|w\|$

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

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

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

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

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

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

x

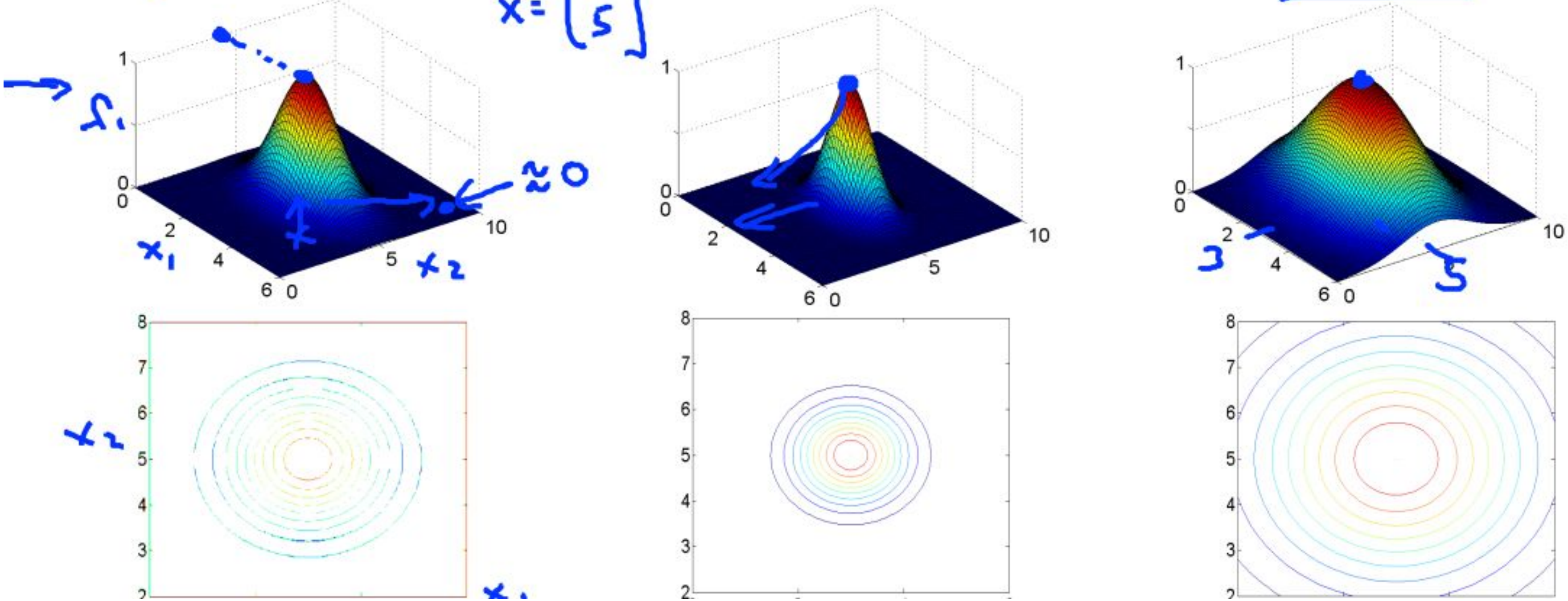
If x if far from $l^{(1)}$:

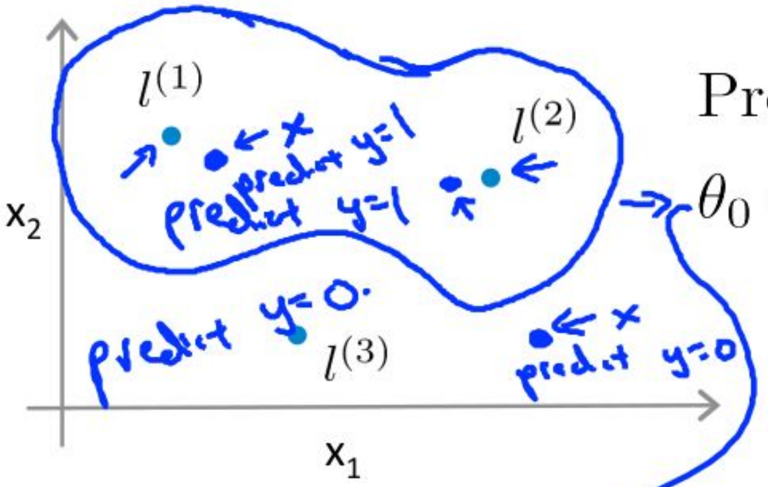
$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \approx 0.$$

Example:

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

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





Predict "1" when

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

x

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

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

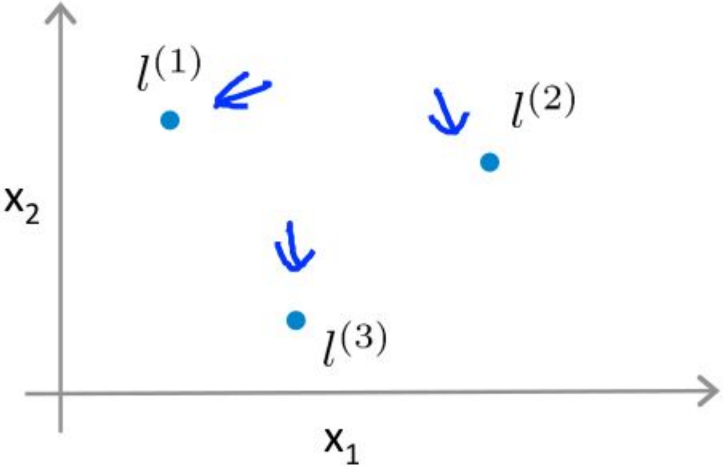
$$\theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$$

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

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

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

Choosing the landmarks

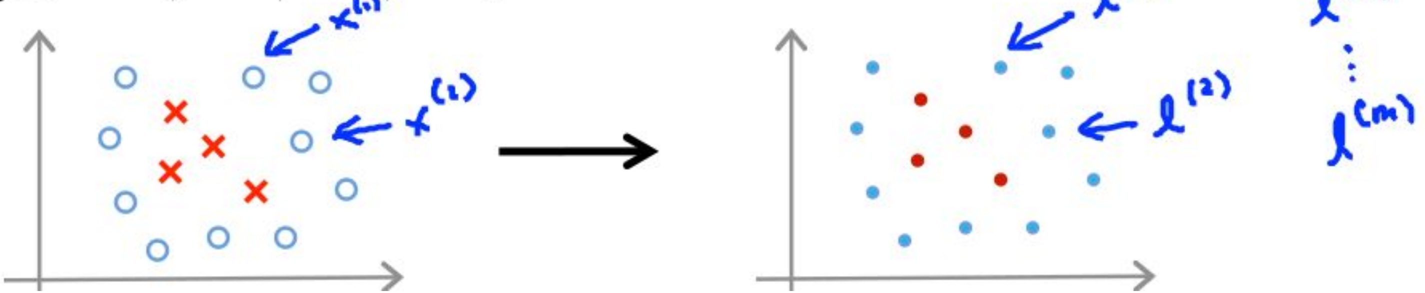


Given x:

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

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

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



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 $\underline{x}$:

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

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

$\ldots$

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

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

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

$l^{(i)} \leftarrow x^{(i)}$

$\underline{x^{(i)}} \in \mathbb{R}^{n+1}$ (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 $f \in \mathbb{R}^{m+1}$ $\qquad \theta \in \mathbb{R}^{m+1}$

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

Training: $\qquad n = m$

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

$\cancel{n} = m$

$\cancel{\theta^T x^{(i)}}$ $\qquad \theta^T f^{(i)}$ $\qquad \to \theta_0$

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

$\|\theta\|^2$ $\qquad m = 10{,}000$

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

SVM parameters:

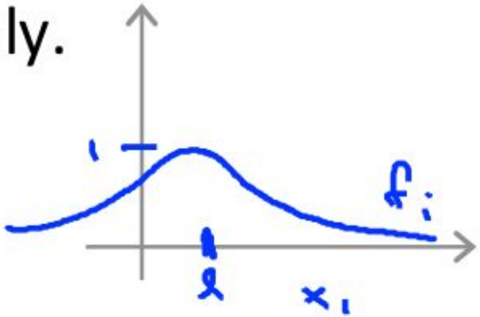
C ($= \frac{1}{\lambda}$). → Large C: Lower bias, high variance. (small λ)

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

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

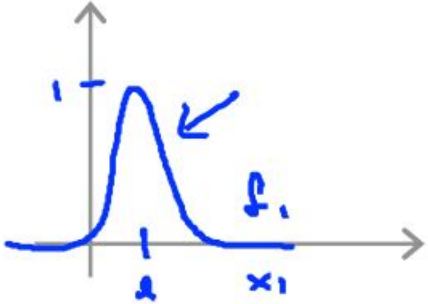
→ Higher bias, lower variance.

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



Small σ^2: Features f_i vary less smoothly.

Lower bias, higher variance.



Logistic Regression vs. SVMs

- Let us consider the following notation:
  - **n** = number of features,   **m** = number of training examples
- If **n** is large relative to **m**:
  - 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

Presented by:
Shatadru Majumdar (https://github.com/shatadru99)
Purbayan Chowdhury (https://github.com/shivishbrahma)
Soham Biswas (https://github.com/Nibba2018)
Balaka Biswas (https://github.com/BALaka-18)
and
Team Explore ML and Team Innovacion