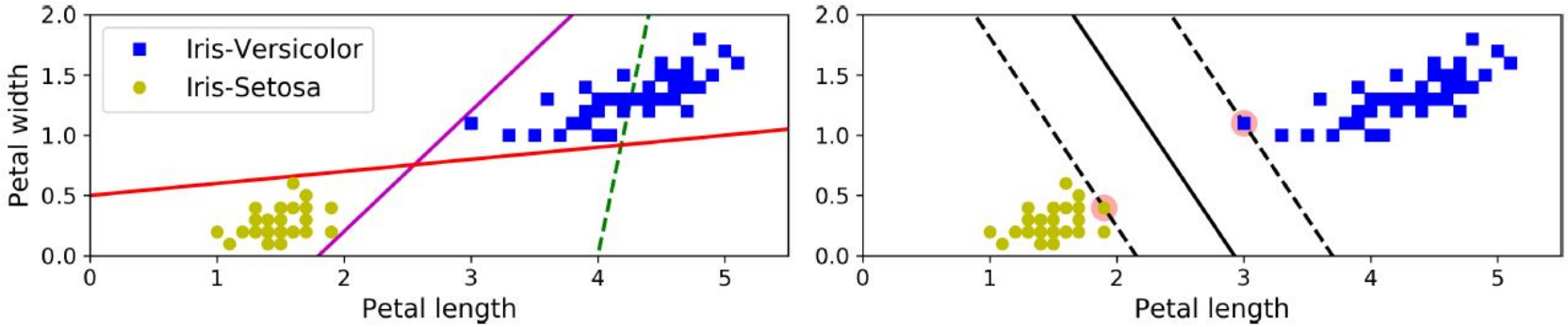
Support Vector Machines
AI4003-Applied Machine Learning

Dr. Mohsin Kamal

Department of Electrical Engineering
National University of Computer and Emerging Sciences, Lahore, Pakistan

- A Support Vector Machine (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
- It is one of the most popular models in Machine Learning, particularly well suited for classification of complex but small- or medium-sized datasets.



- You can think of an SVM classifier as fitting the widest possible street (represented by the parallel dashed lines) between the classes. This is called large margin classification.
- Adding more training instances "off the street" will not affect the decision boundary at all: it is fully determined (or "supported") by the instances located on the edge of the street. These instances are called the support vectors

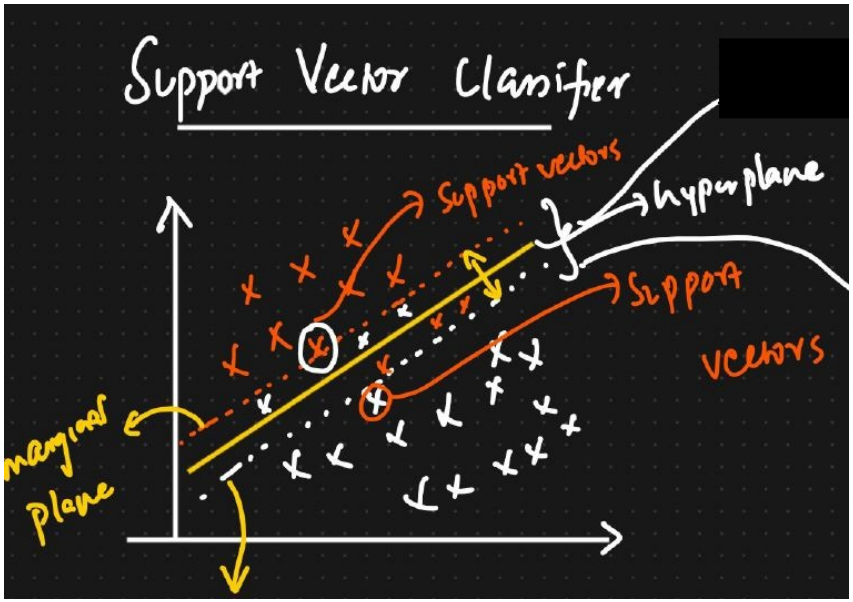
SVM performs the following tasks:

1. Classification Problems (SVM is referred as Support Vector Classifier)
2. Regression Problems (Referred as Support Vector Regressor)

SVM works on the basis of Logistic regression (presented in later slides)

SVM Classification

- In SVM marginal lines are also drawn
- Support vectors help in drawing the marginal lines
- Aim: i) to design hyperplane and marginal lines, and ii) to make the distance of marginal plane as large as possible
- Two types of marginal planes, i) Hard marginal plane, ii) Soft marginal plane



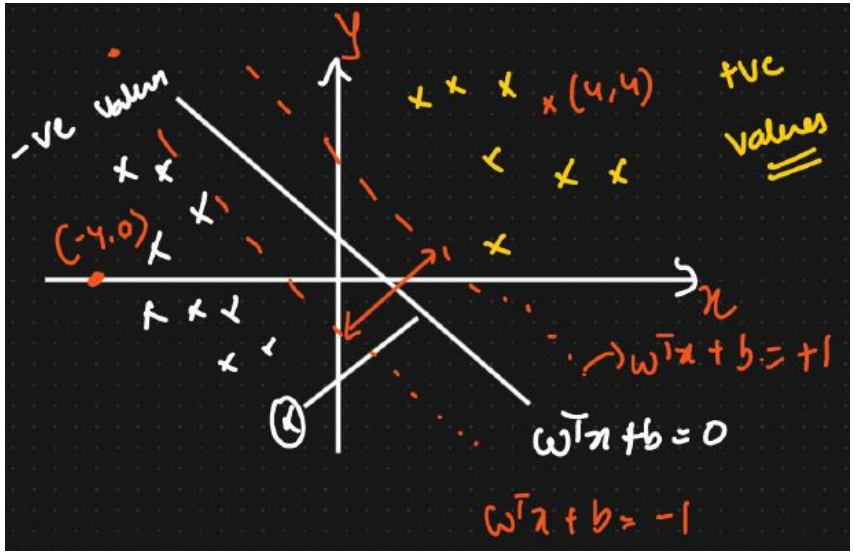
Considering linear SVM/SVC,
We know the straight line equation can be written in different forms

1. $y = \theta_0 + \theta_1 x$, or
2. $y = w_1 x_1 + b$, or
3. $y = w^T x + b$, or
4. $ax + by + c = 0$

let us further solve 4^{th} to understand how this is a straight line equation.

$$by = -ax - c$$
$$y = -\frac{a}{b}x - \frac{c}{b}$$

- We have to look for best fit line with large width of marginal line
- Suppose we get $3x + 2y + 4 = 0$ and have data points on plane at $(-4, 0)$ and $(4, 4)$
- We get -8 and 24, respectively, by putting the data point values in the equation that we have supposed for best fit line.



-

This shows that any value above the line will be positive while value below the line will always be negative.

- To get the marginal plane: $w^T x + b = k$
- Let $k = 1, -1$, so the upper plane and lower plane becomes, respectively,

$$w^T x + b = 1$$
$$w^T x + b = -1$$

- To find the width of marginal plane, we can subtract upper marginal plane from the lower marginal plane, we get,

$$w^T(x_1 - x_2) = 2 \tag{1}$$

- w is the slope which has two components i.e., i) magnitude, ii) vector
- If eq. (1) is divided by its magnitude, we get the vector of it i.e.,

$$\frac{w^T}{\|\mathbf{w}\|}(x_1 - x_2) = \frac{2}{\|\mathbf{w}\|}$$

- R.H.S of the above equation is representing the marginal plane distance and our goal is to maximize it, so,

$$objective: \max_{(w,b)} \frac{2}{\|\mathbf{w}\|} \tag{2}$$

$$\text{subject to } y_i = \begin{cases} 1\ (positive\ class) & w^T x + b \geq 1 \\ -1\ (negative\ class) & w^T x + b \leq -1 \end{cases}$$

-For all accurate datapoints,

$$y \times (w^T + b) \geq 1$$

- because

$$1 \times (w^T + b) \geq 1$$
$$= (w^T + b) \geq 1$$

and,

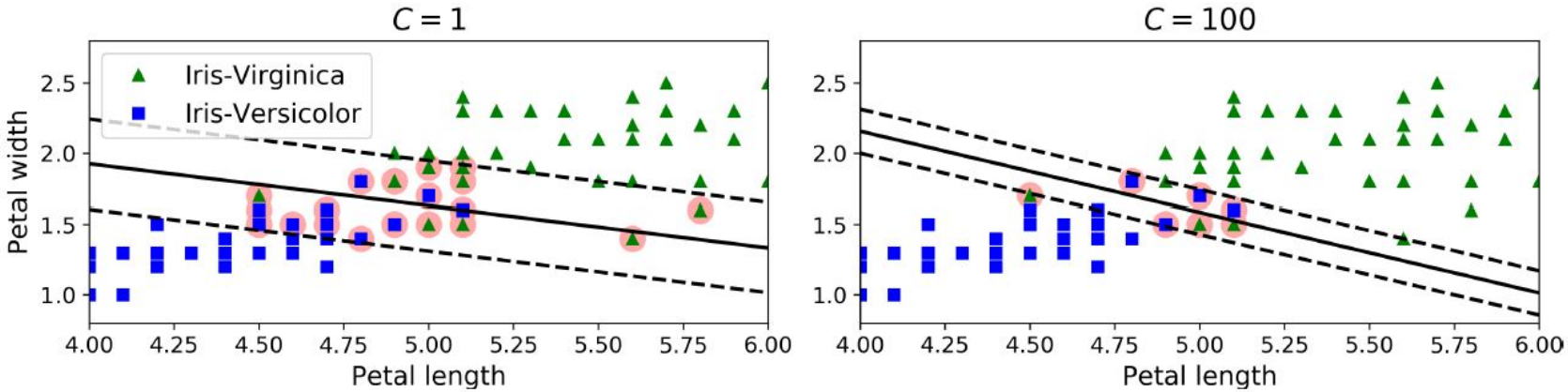
$$-1 \times (w^T + b) \leq -1$$
$$= (w^T + b) \geq 1$$

- Objective function in eq. (2) can be written as,

$$\min_{(w,b)} \frac{\|\mathbf{w}\|}{2}$$

The cost function of SVM becomes

$$cost: \min_{(w,b)} \frac{\|\mathbf{w}\|}{2} + c_i \sum_{i=1}^{n} \xi_i$$

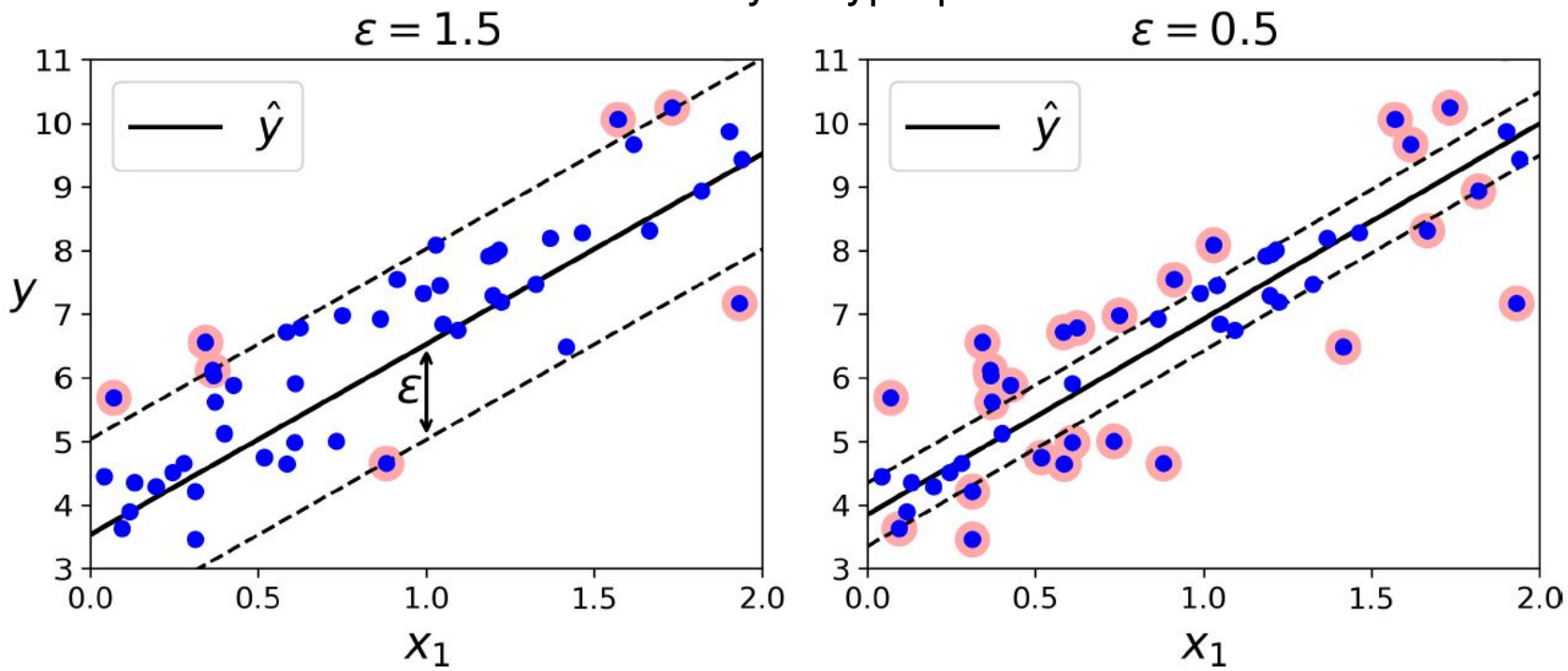


- c_i is representing "How many points can be avoided to misclassify".
- $\sum_{i=1}^{n} \xi_i$ represents the summation of the distances of the misclassified points from marginal plane.
- It is preferable to use a more flexible model. The objective is to find a good balance between keeping the street as large as possible and limiting the margin violations (i.e., instances that end up in the middle of the street or even on the wrong side). This is called soft margin classification.
-The term $c_i \sum_{i=1}^{n} \xi_i$ is classed hinge loss function.

SVM Regression:

- The SVM algorithm is quite versatile: not only does it support linear and nonlinear classification, but it also supports linear and nonlinear regression.
- The trick is to reverse the objective: instead of trying to fit the largest possible street between two classes while limiting margin violations, SVM Regression tries to fit as many instances as possible on the street while limiting margin violations (i.e., instances off the street).

The width of the street is controlled by a hyperparameter ϵ.



- Marginal planes are equidistant i.e., $w^T x + \epsilon$ and $w^T x - \epsilon$ from the hyper plane (considering $b = 0$).
- Suppose the predicted point is $w^T x_i$, so, the constraint is:

$$|y - w^T x_i| \leq \epsilon$$

where $w^T x_i$ is the predicted output $\hat{y}$

- If this is the condition then the output lies in the street which represents a good predicted output.
- For the predicted output outside margins,

$$cost : \min_{(w,b)} \frac{\|\mathbf{w}\|}{2} + c \sum_{i=1}^{n} \xi_i$$

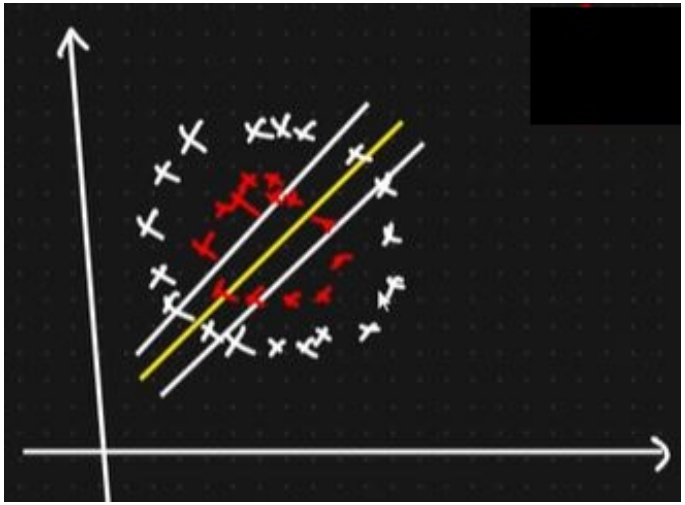
- The above constraint becomes:

$$|y - w^T x_i| \leq \epsilon + |\xi_i|$$

- This represents that the distance of data points outside marginal plane is added with ϵ.

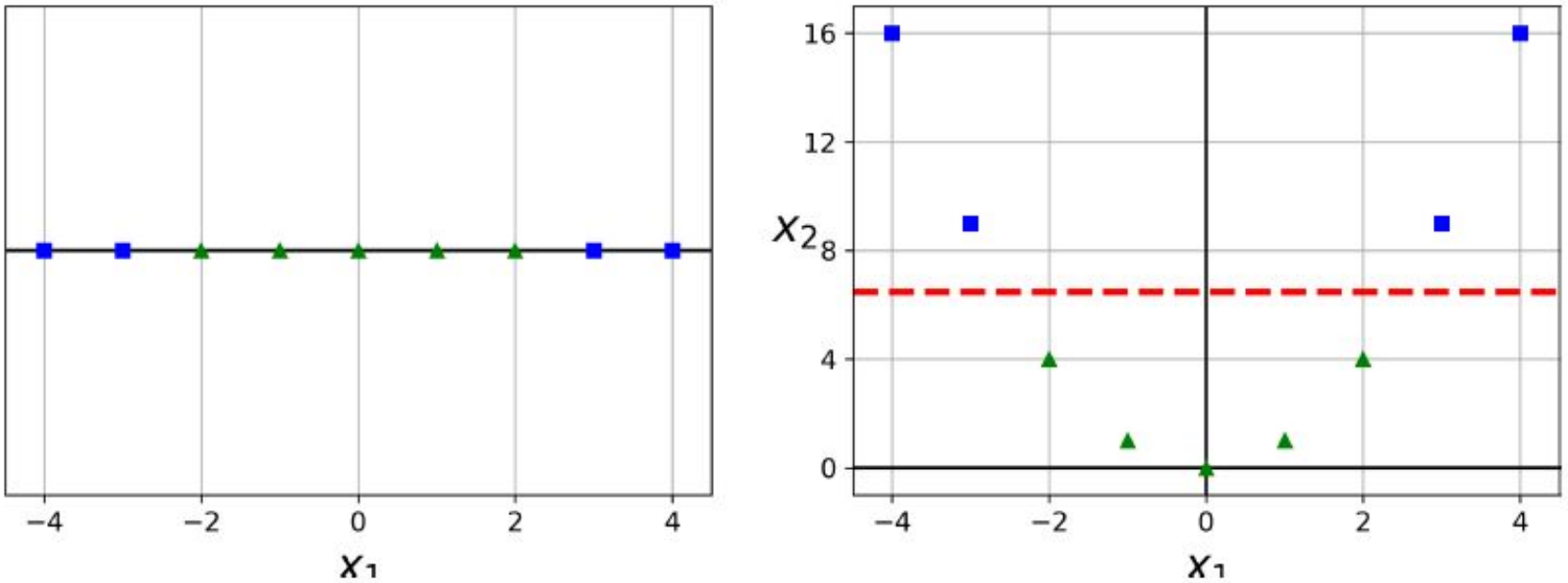
SVM Kernel:

-Suppose we have a non-linear dataset. Applying the linear SVM will reduce the accuracy.



- Fortunately, when using SVMs you can apply an almost miraculous mathematical technique called the kernel trick.
- We will require a transformation method to transform it into another form where it can be linearly separable.
- One way of doing it would be to transform it into 3-D plane.

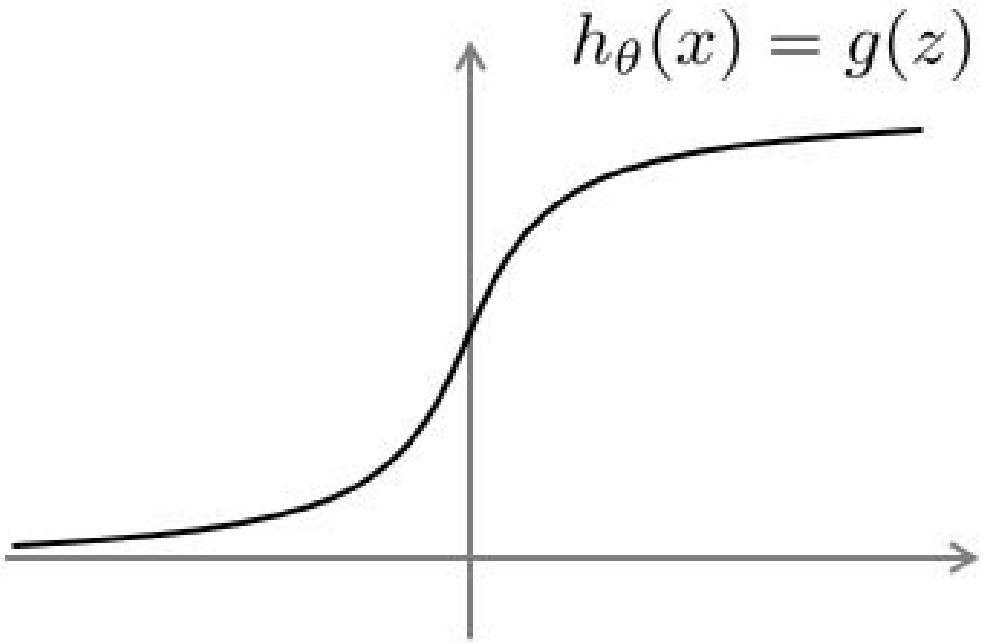
- Figure below represents a simple dataset with just one feature x_1.
- This dataset is not linearly separable, as you can see. But if you add a second feature $x_2 = x_1^2$, the resulting 2D dataset is perfectly linearly separable.



- This is an example of simple Kernel. Other Kernels include, Polynomial Kernel, RBF Gaussian Kernel, Sigmoid Kernel etc.

Alternative view of logistic regression

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



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

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

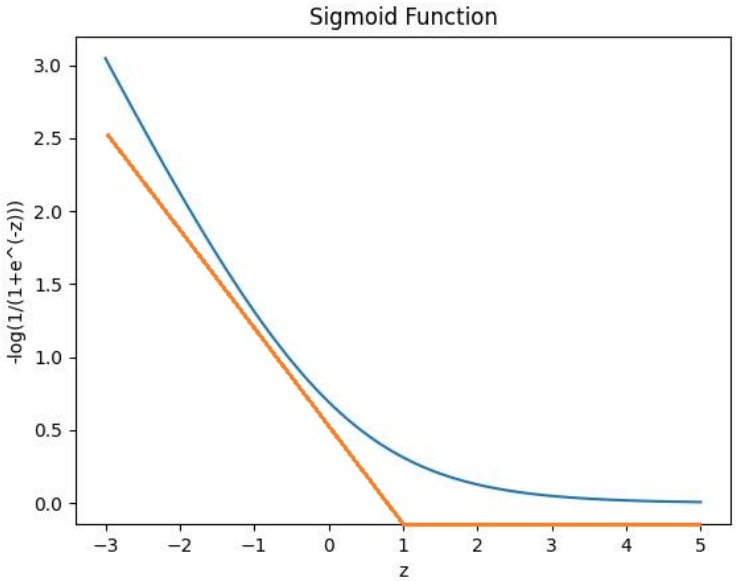
Alternative view of logistic regression

For an example i.e., (x, y)

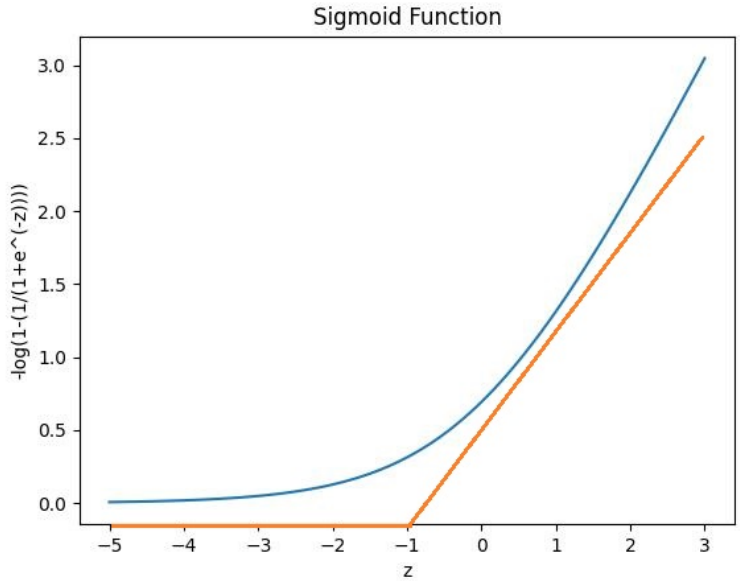
Cost: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

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

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



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



Support vector machine

Logistic regression:

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

Here, the equation looks like $A + \lambda B$ in which if λ is high, we give high weight to B

Support vector machine:

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

In SVM, the equation is of the form: $CA + B$. This shows if C is small, we give high weight to B where C is playing the similar role as $\frac{1}{\lambda}$. Multiply m to the equation of SVM as it does not effect the cost function. The above equation becomes:

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

SVM hypothesis

cost function

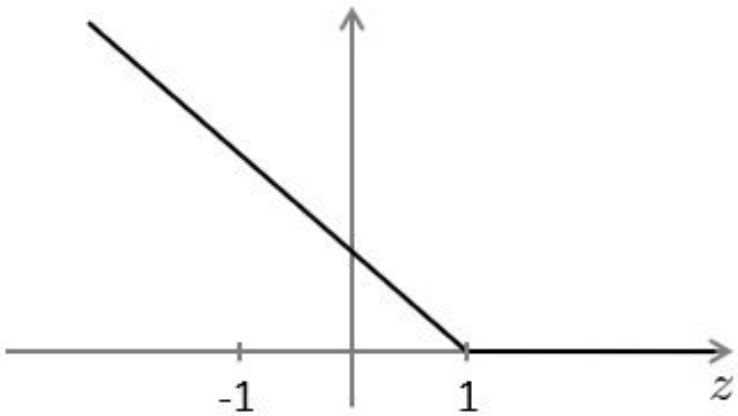
$$= \min_{\theta} C \left[\sum_{i=1}^{m} y^{(i)} cost_1(\theta^T x^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T x^{(i)}) \right] + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

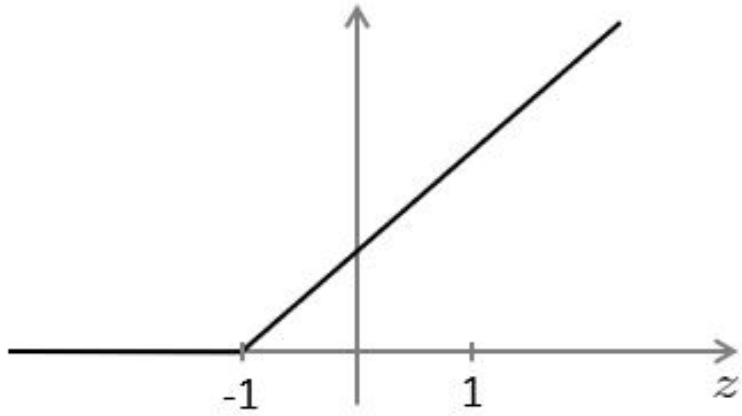
Hypothesis:

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

As presented in previous slides, **Support vector machine**

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





If $y = 1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

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

SVM decision boundary

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

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

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

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

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

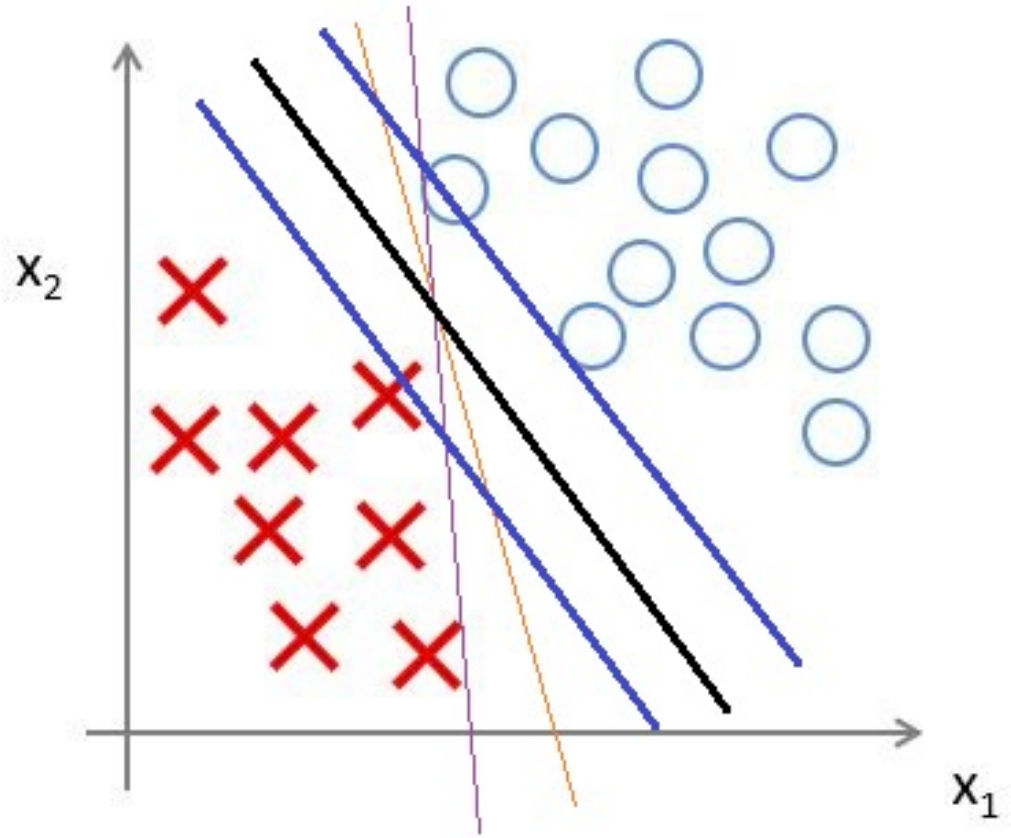
If C is very large, the term multiplied to C becomes almost equal to 0.

So the SVM minimization objective becomes:

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

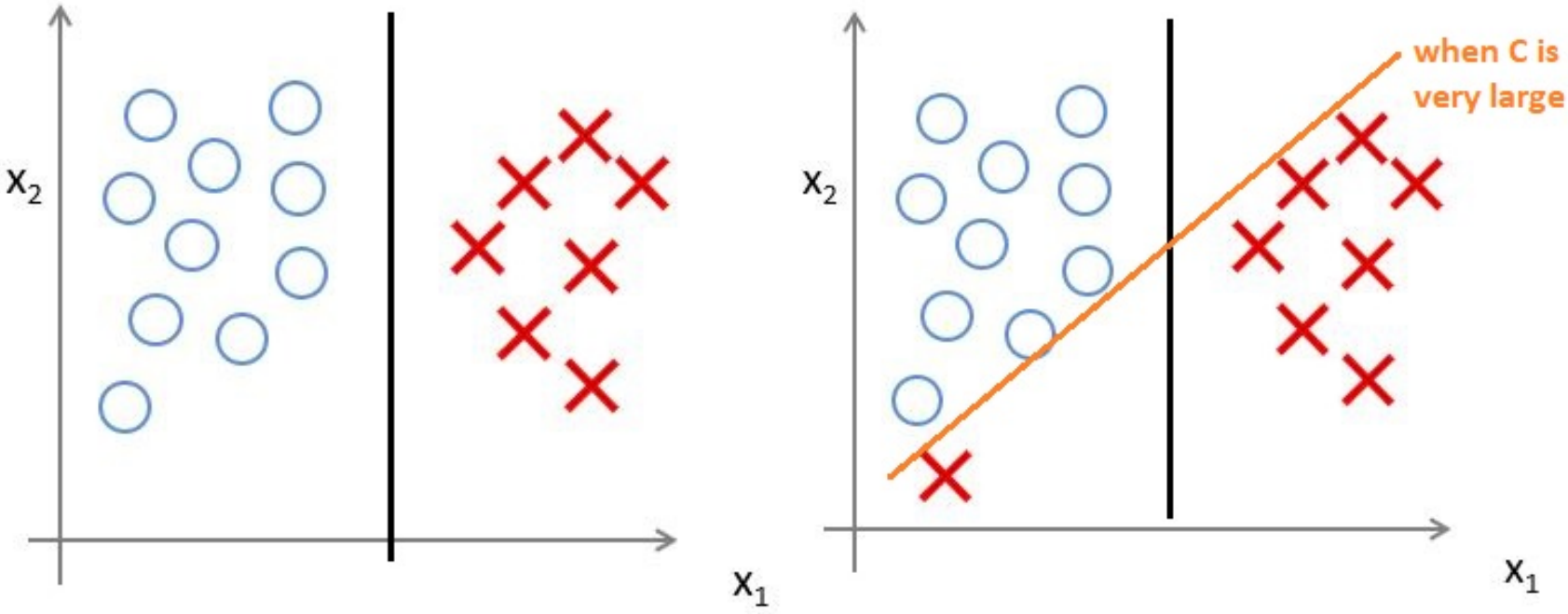
$$\text{subject to } \begin{cases} \theta^T x^{(i)} \geq 1 & \text{if } y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 & \text{if } y^{(i)} = 0 \end{cases}$$

SVM Decision Boundary: Linearly separable case

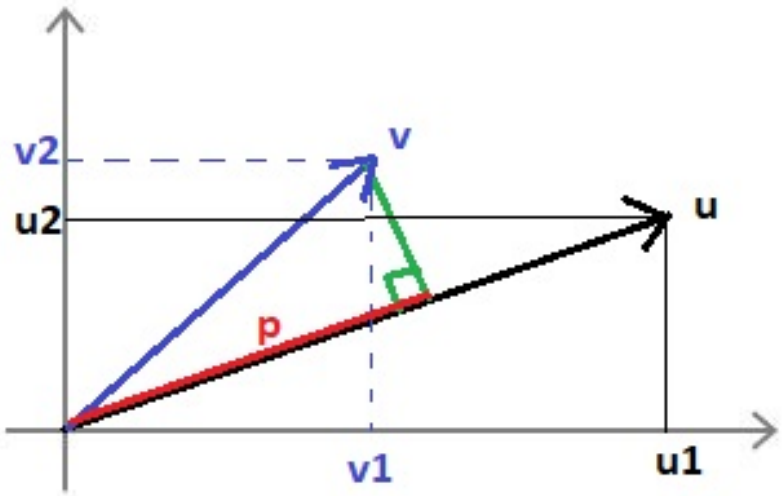


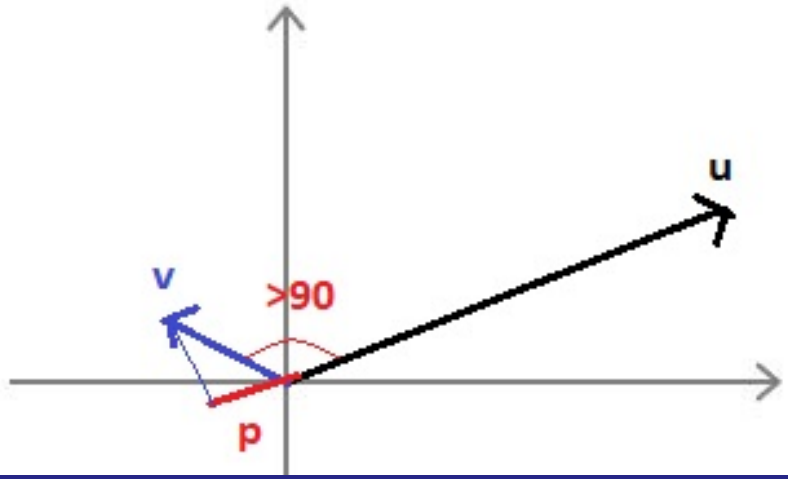
Large margin classifier

Large margin classifier in presence of outliers



Vector inner product





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

$$u^T v = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

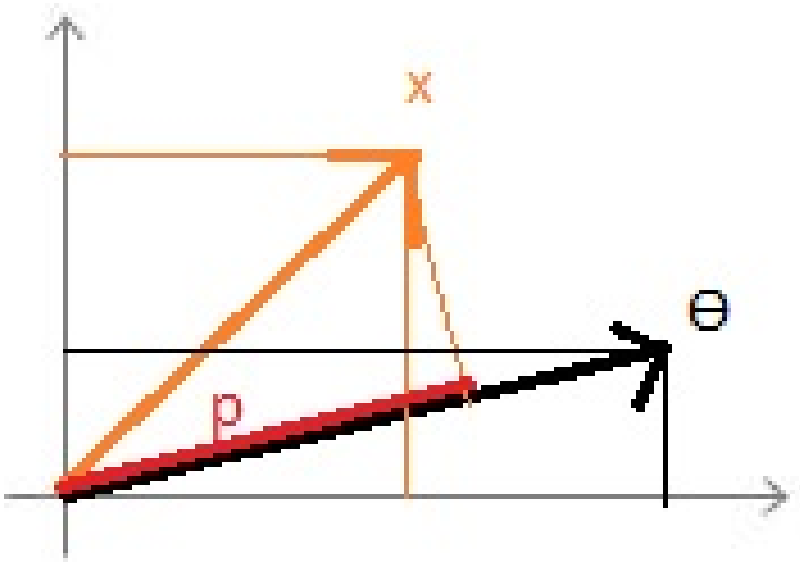
$\|u\|$ = length of vector u

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

p = signed length of projection of v onto u

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

SVM Decision Boundary



$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 \qquad (3)$$

$$\text{s.t.} \begin{cases} \theta^T x^{(i)} \geq 1 & \text{if } y^{(i)} = 1 \\ \theta^T x^{(i)} \leq -1 & \text{if } y^{(i)} = 0 \end{cases}$$

For simplification, let $\theta_0 = 0$ and $n = 2$

eq. 3 becomes,

$$\frac{1}{2}(\theta_1^2 + \theta_2^2) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\theta\|^2$$

Based on the projection of x on θ,

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

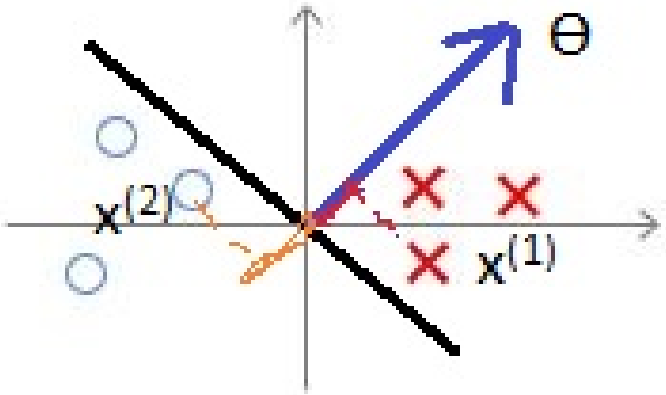
SVM Decision Boundary

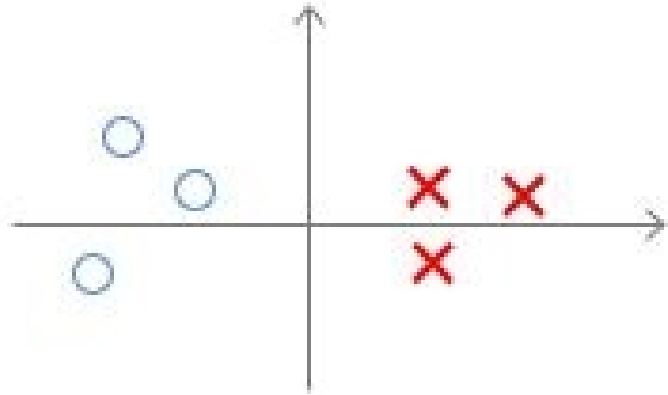
$$\min_{\theta} \frac{1}{2} \sum_{j=1}^{n} \theta_j^2 = \frac{1}{2} \|\theta\|^2$$

$$\text{s.t.} \begin{cases} p^{(i)}.\|\theta\| \geq 1 & \text{if } y^{(i)} = 1 \\ p^{(i)}.\|\theta\| \leq -1 & \text{if } y^{(i)} = 0 \end{cases}$$

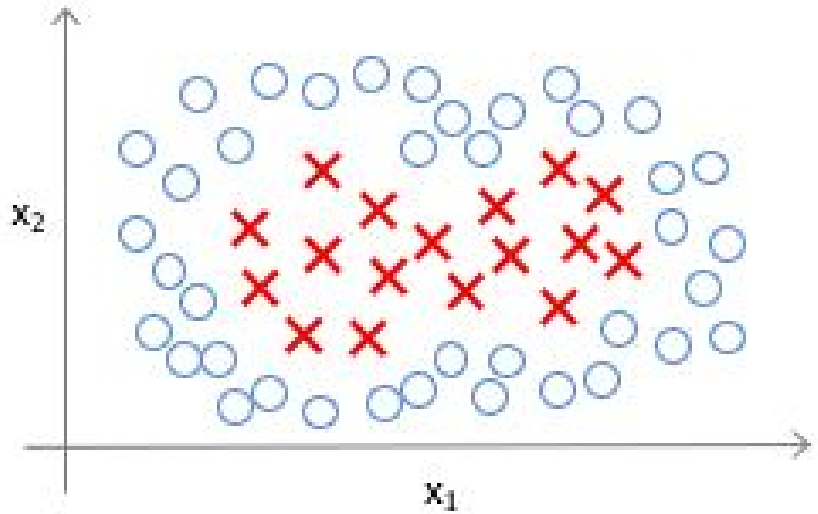
where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ.

For simplification: $\theta_0 = 0$





Non-linear Decision Boundary



Predict $y = 1$, if

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

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

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \ldots$$

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

Is there a different/better choice of 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:

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

$$f_3 = similarity(x, l^{(2)}) = exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$$

$$f_3 = similarity(x, l^{(3)}) = exp\left(-\frac{\|x - l^{(3)}\|^2}{2\sigma^2}\right)$$

Kernels and Similarity

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

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

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

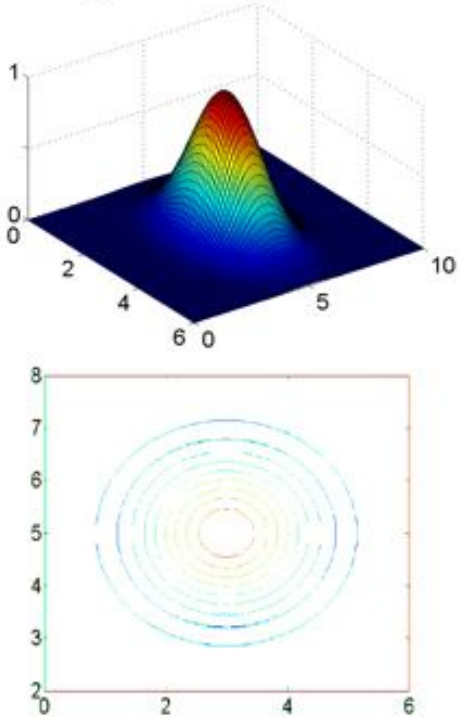
If x is far from $l^{(1)}$:

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

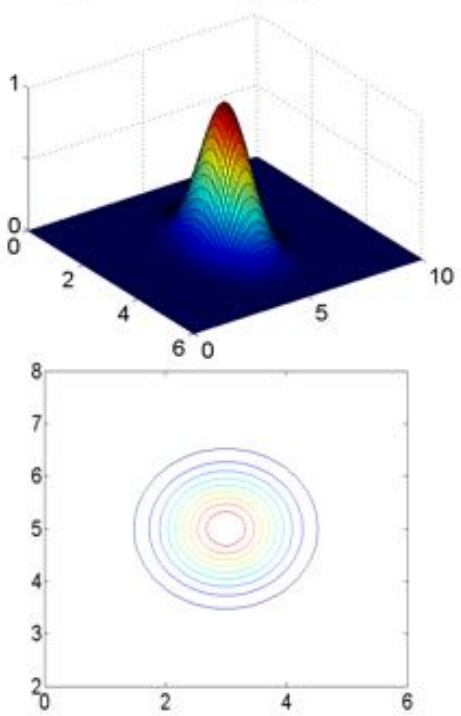
Example

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

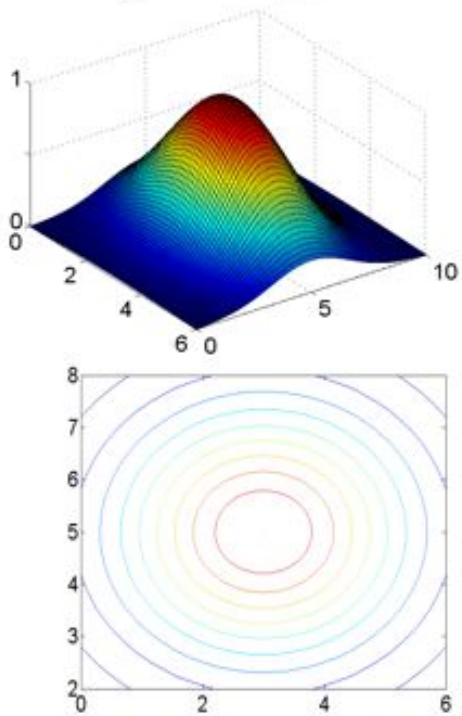
$\sigma^2 = 1$

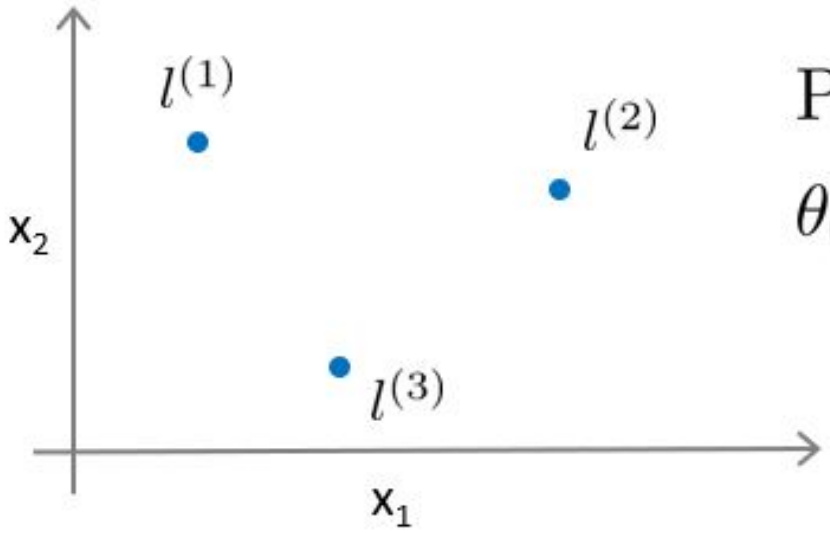


$\sigma^2 = 0.5$



$\sigma^2 = 3$



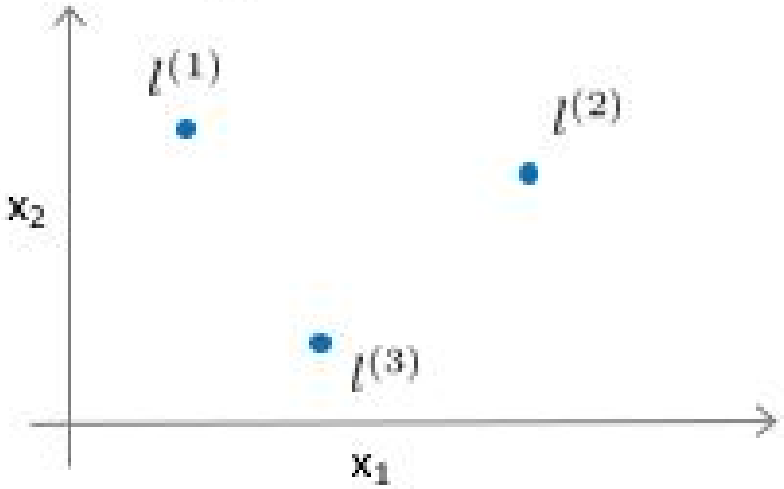


Predict "1" when

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

For example we get $\theta_0 = -0.5, \theta_1 = 1, \theta_2 = 1, \theta_3 = 0$
and, $f_1 \approx 1, f_2 \approx 0, f_3 \approx 0$
So, we get $0.5 \geq 0$

Choosing the landmarks

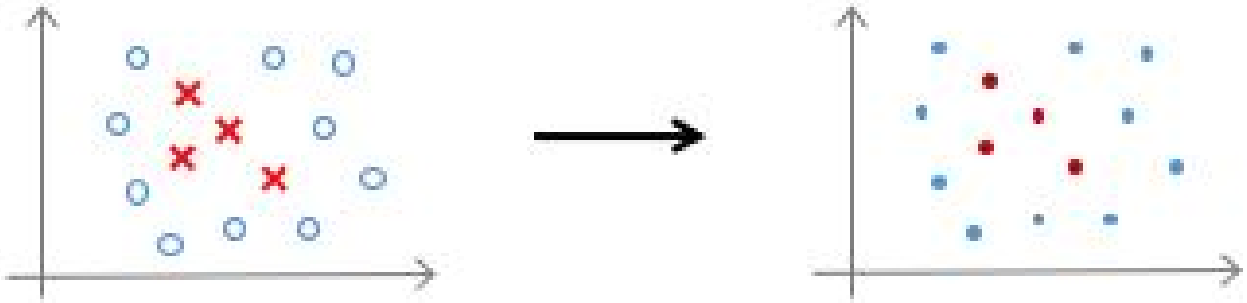


Given x:

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

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 x:

$$f_1 = \text{similarity}(x, l^{(1)})$$
$$f_2 = \text{similarity}(x, l^{(2)})$$
$$\ldots$$

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

SVM with Kernels

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

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

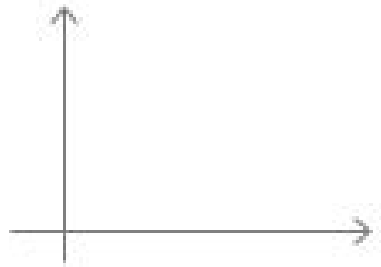
Training:

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

SVM parameters:

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

σ^2 Large σ^2: Features f_i vary more smoothly.
Higher bias, lower variance.



Small σ^2: Features f_i vary less smoothly.
Lower bias, higher variance.