CSE 352: Machine Learning and Pattern Recognition

جامعة مصر للمعلوماتية
EGYPT UNIVERSITY OF INFORMATICS

3: Supervised Learning

FACULTY OF ENGINEERING

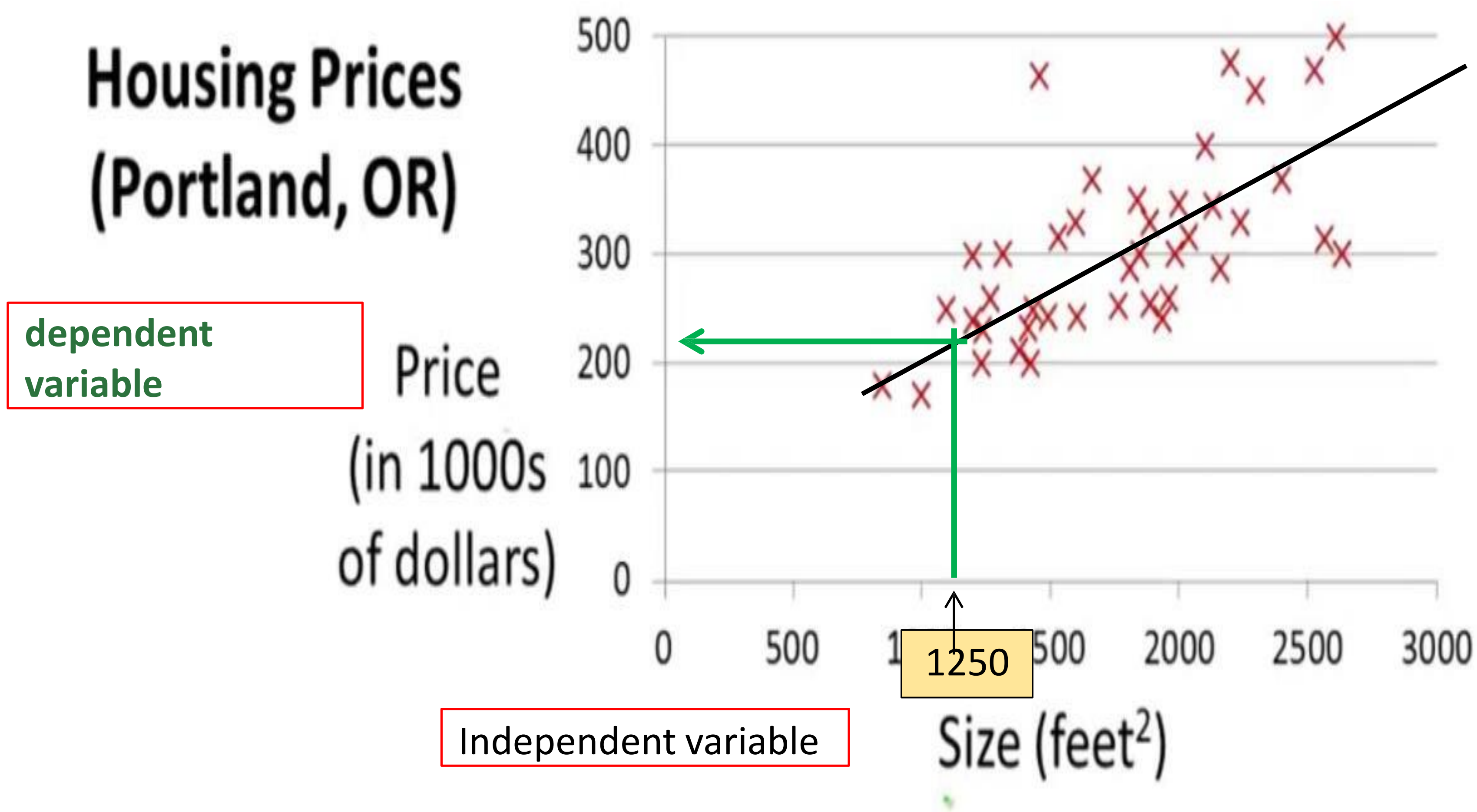
Fall 25

Linear regression with one variable

LINEAR REGRESSION WITH ONE VARIABLE

- Model Representation
- Cost Function
- Gradient Descent

MODEL REPRESENTATION



Supervised Learning

"right answers" or "Labeled data" given

Regression:

Predict continuous valued output (price)

MODEL REPRESENTATION

Training set of housing prices (Portland, OR)

Size in feet^2 (x)	Price (\$) in 1000's (y)
2104	460
1416	232
1534	315
852	178
...	...

$$\left.\begin{matrix}460\\232\\315\\178\\\ldots\end{matrix}\right\}\ m$$

Notation:

- **m** = Number of training examples
- **x**'s = "input" variable / features
- **y**'s = "output" variable / "target" variable

(x,y) one training example (one raw)

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

Example

$x^{(1)}$	2104
$y^{(2)}$	232
$x^{(4)}$	852

MODEL REPRESENTATION

$$\begin{tikzcd}
 & \text{Training set} \arrow[d] & \\
 & \text{Learning algorithm} \arrow[d] & \\
x \arrow[r] & h \arrow[r] & y
\end{tikzcd}$$

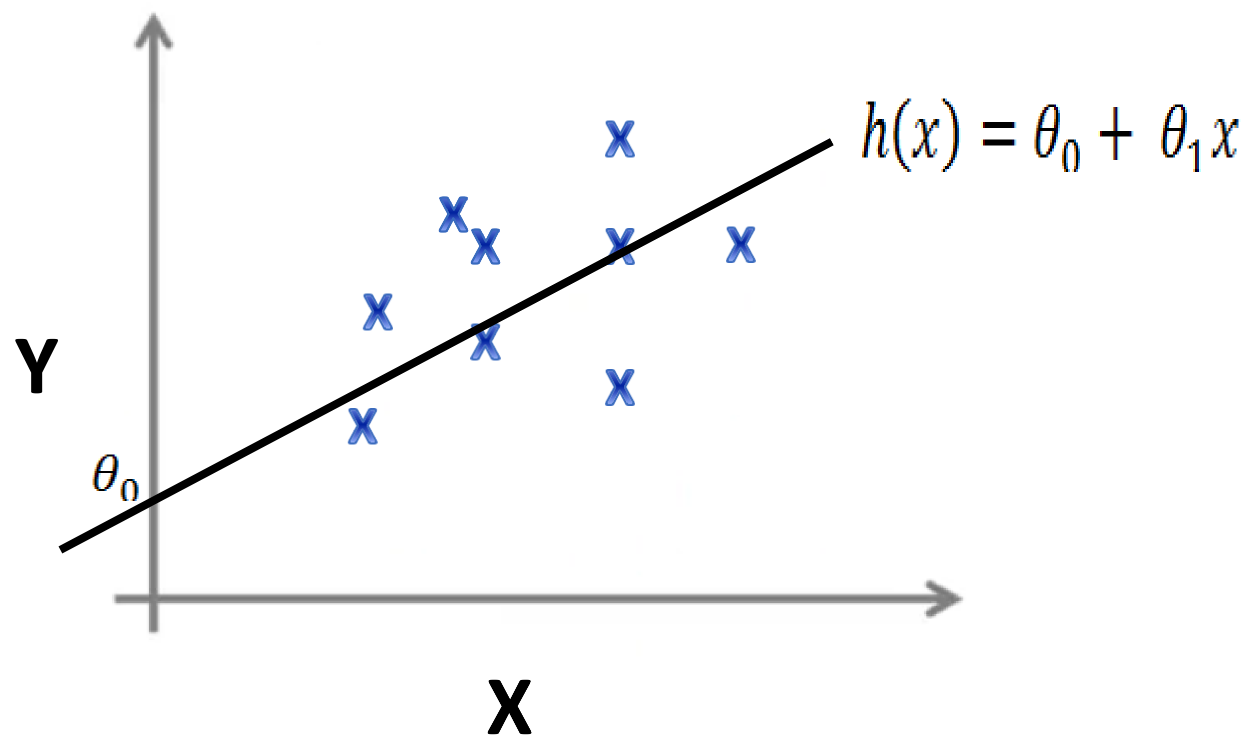
the job of a learning algorithm to output a function is usually denoted lowercase **h** and **h** stands for hypothesis

the job of a hypothesis function is taking the value of x and it tries to output the estimated value of y. So h is a function that maps from x's to y's

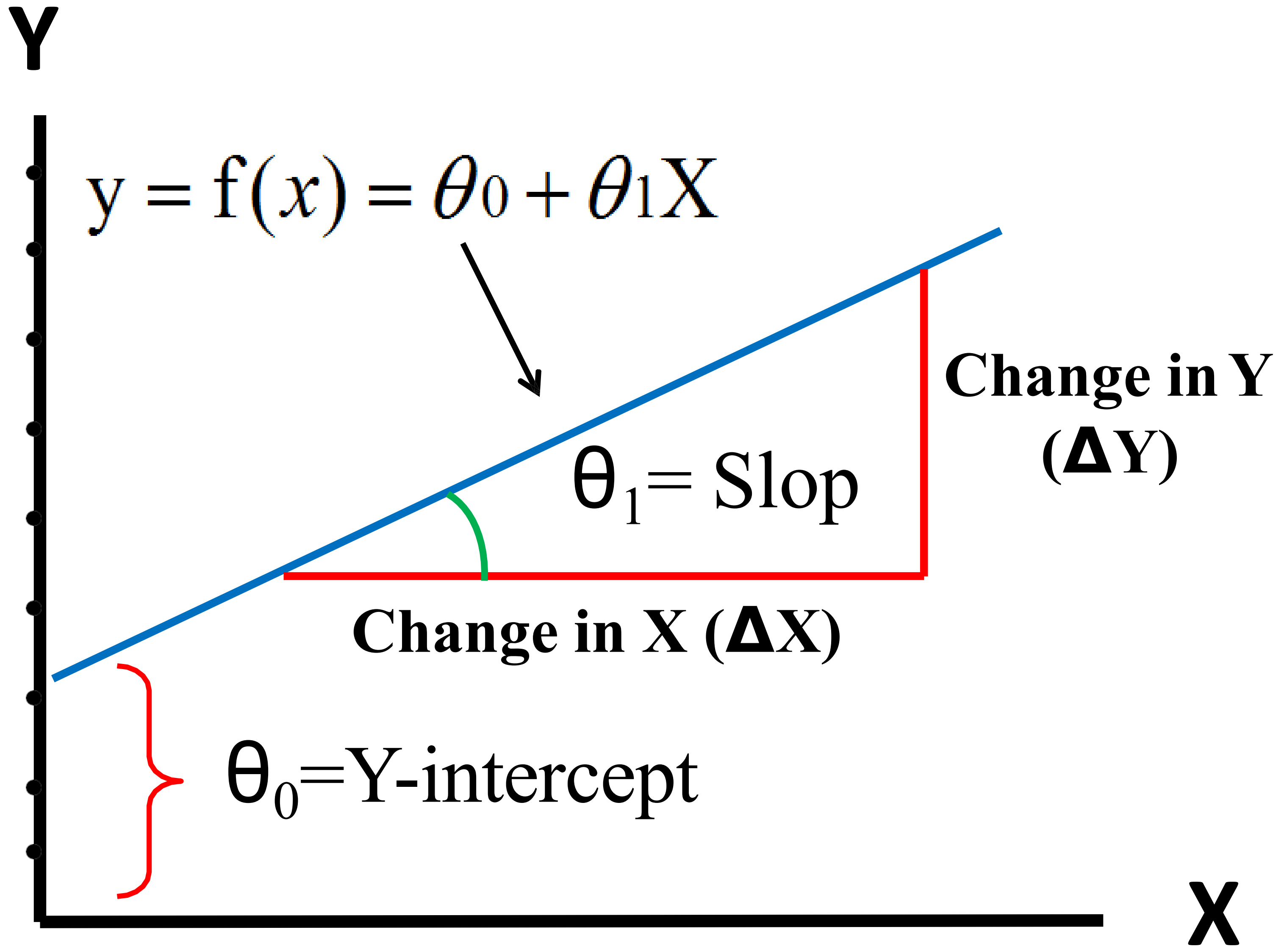
MODEL REPRESENTATION

How do we represent *h* ?

$$h(x) = \theta_0 + \theta_1 x$$

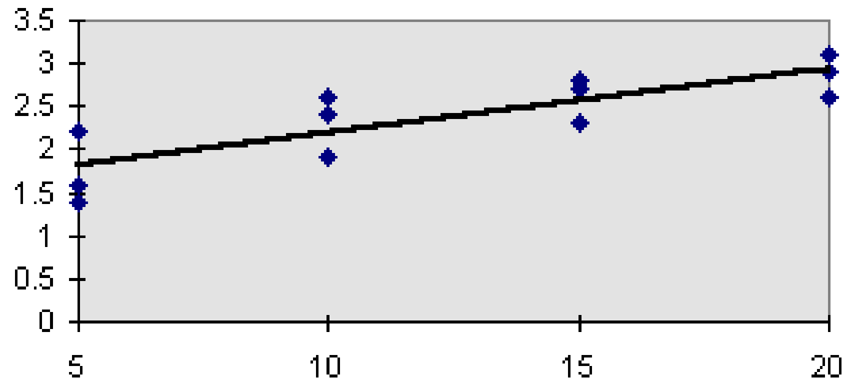


Linear Equations

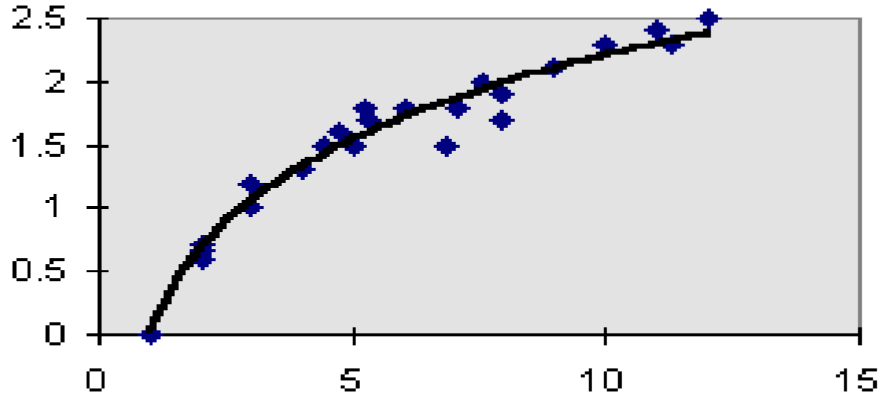


Types of Regression Models

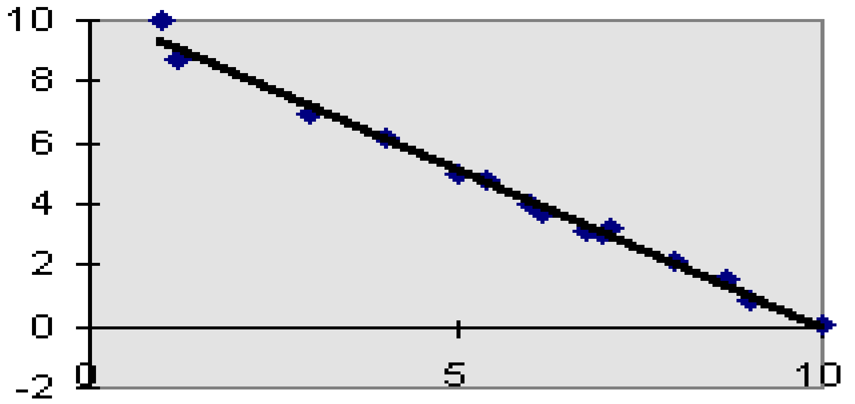
Positive Linear Relationship



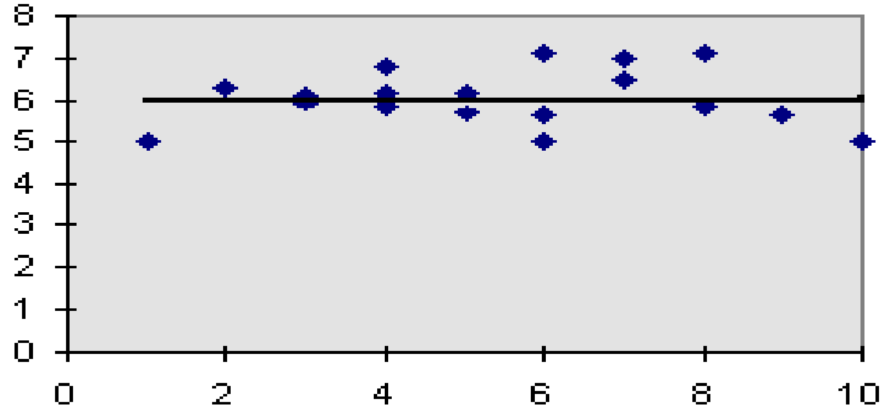
Relationship NOT Linear



Negative Linear Relationship

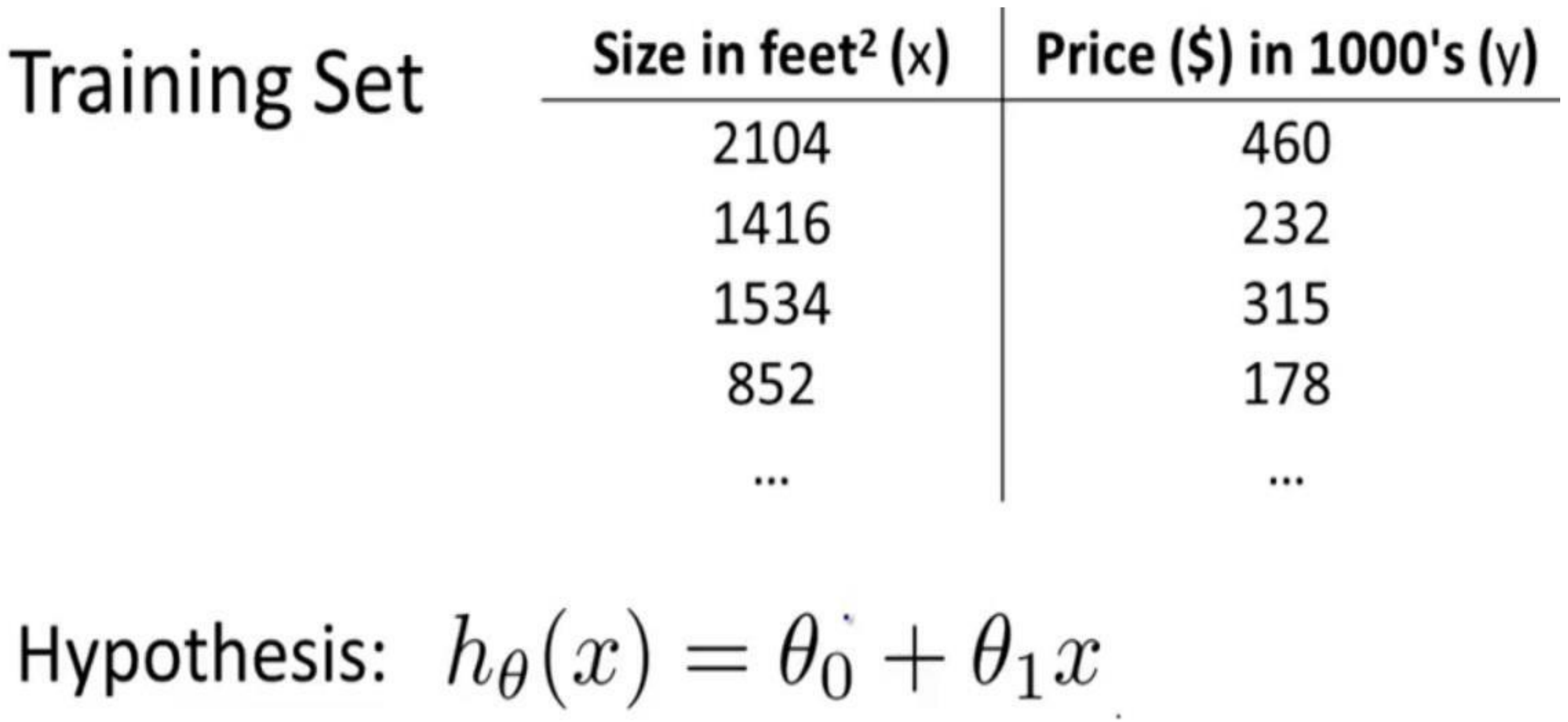


No Relationship



COST FUNCTION

- *The cost function*, let us figure out how to fit the best possible straight line to our data.



Training Set

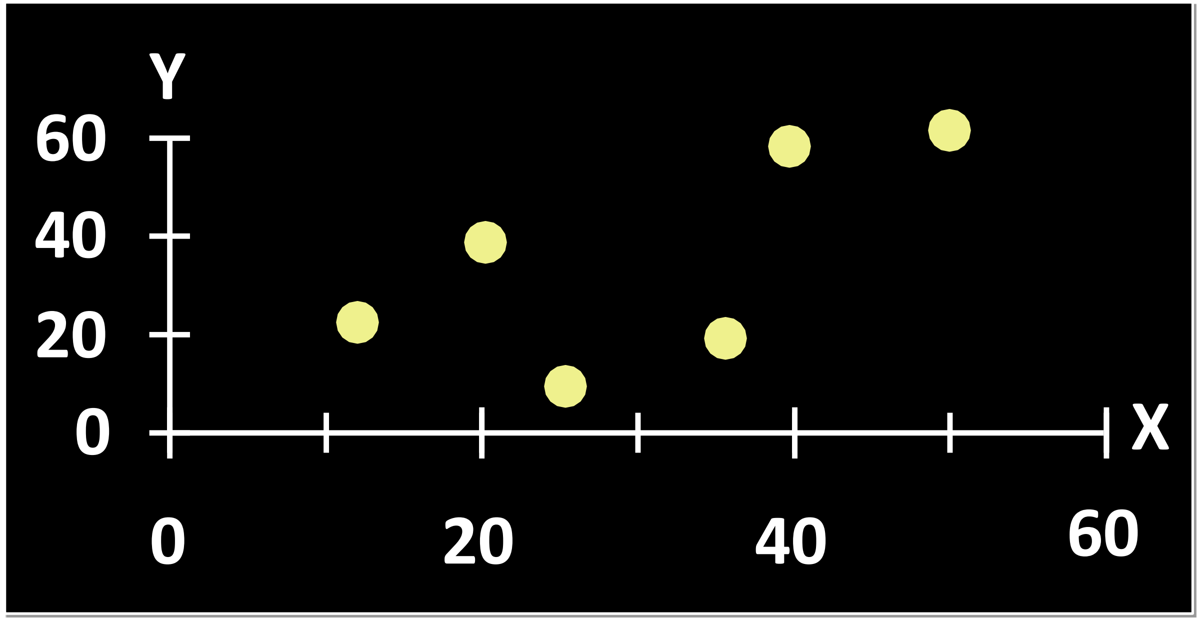
Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$

How to choose $\boldsymbol{\theta}_{\text{i's}}$?

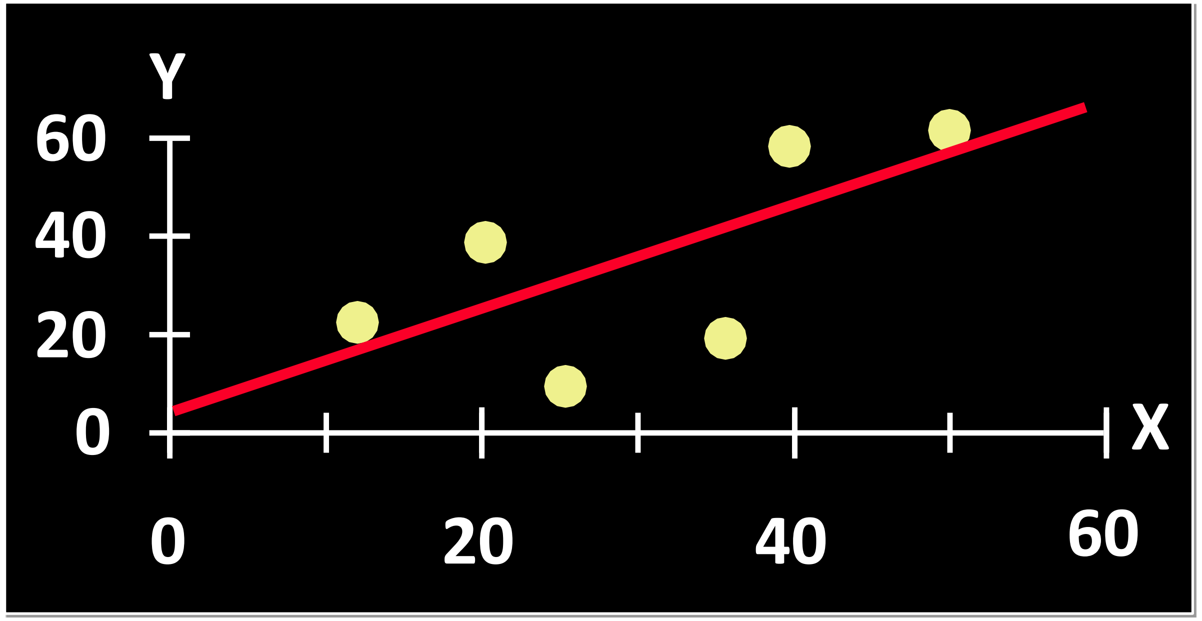
Scatter plot

- 1. Plot of All (X_i, Y_i) Pairs
- 2. Suggests How Well Model Will Fit



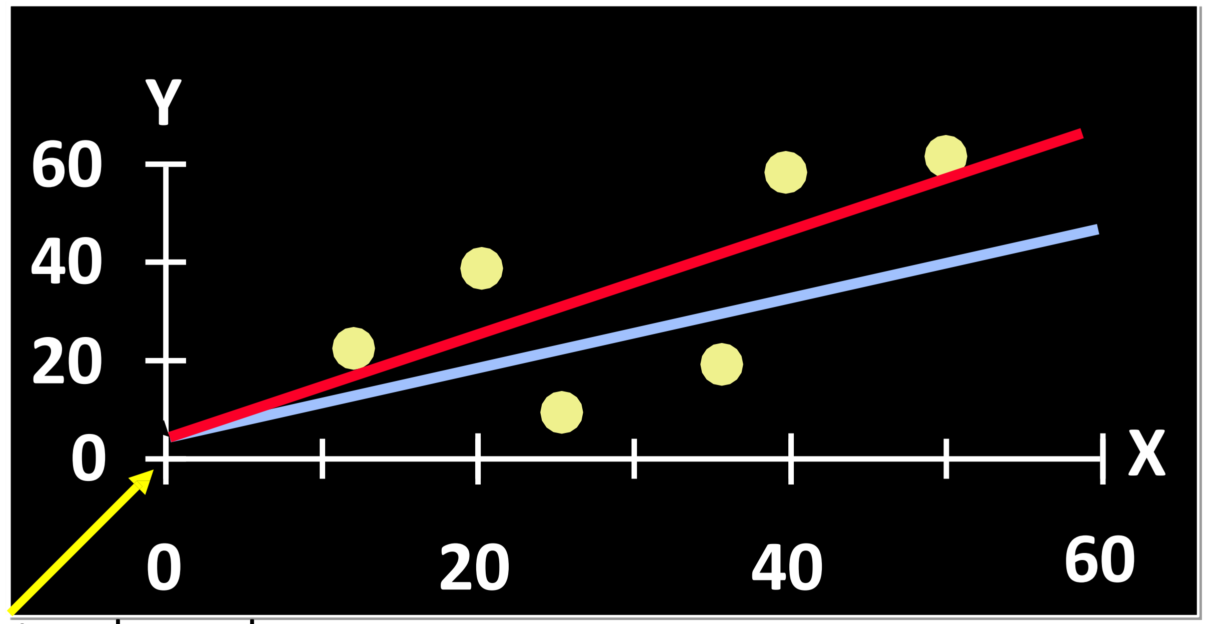
Thinking Challenge

How would you draw a line through the points?
How do you determine which line 'fits best'?



Thinking Challenge

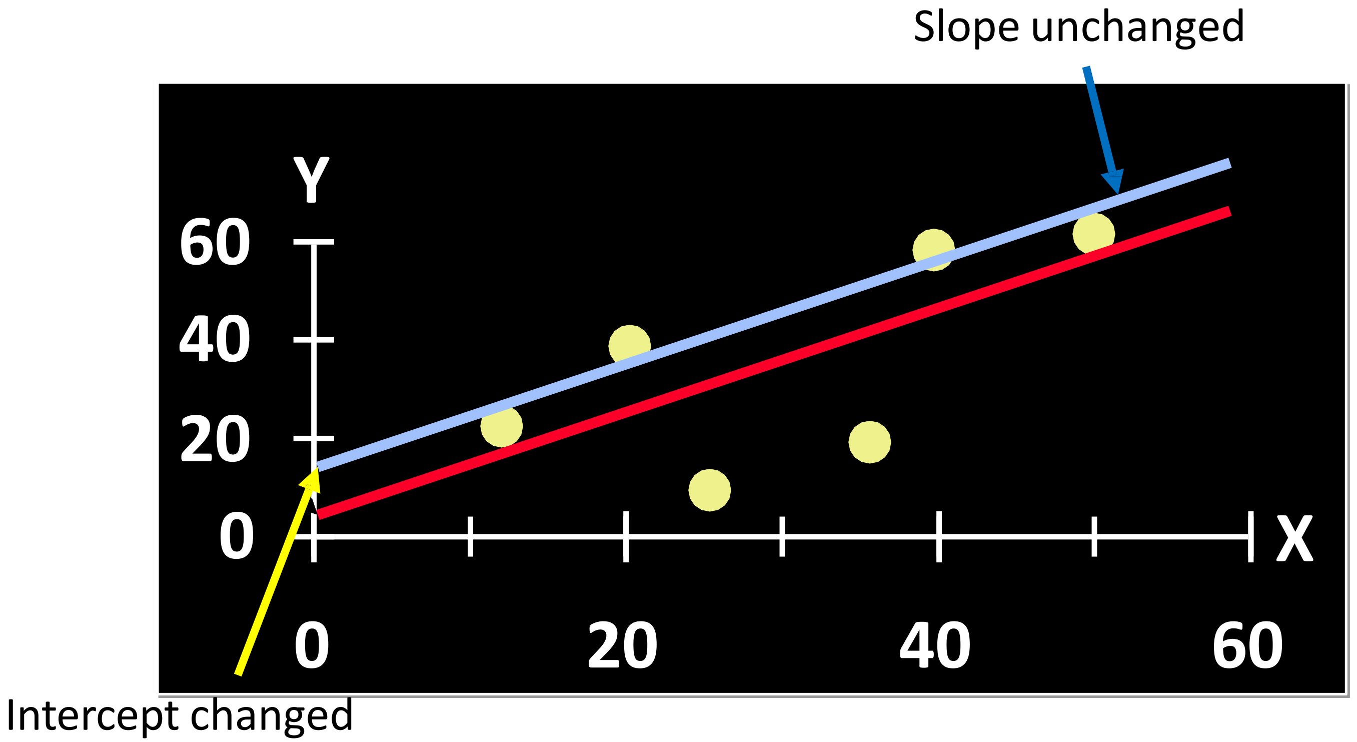
How would you draw a line through the points?
How do you determine which line 'fits best'?



Intercept unchanged

Thinking Challenge

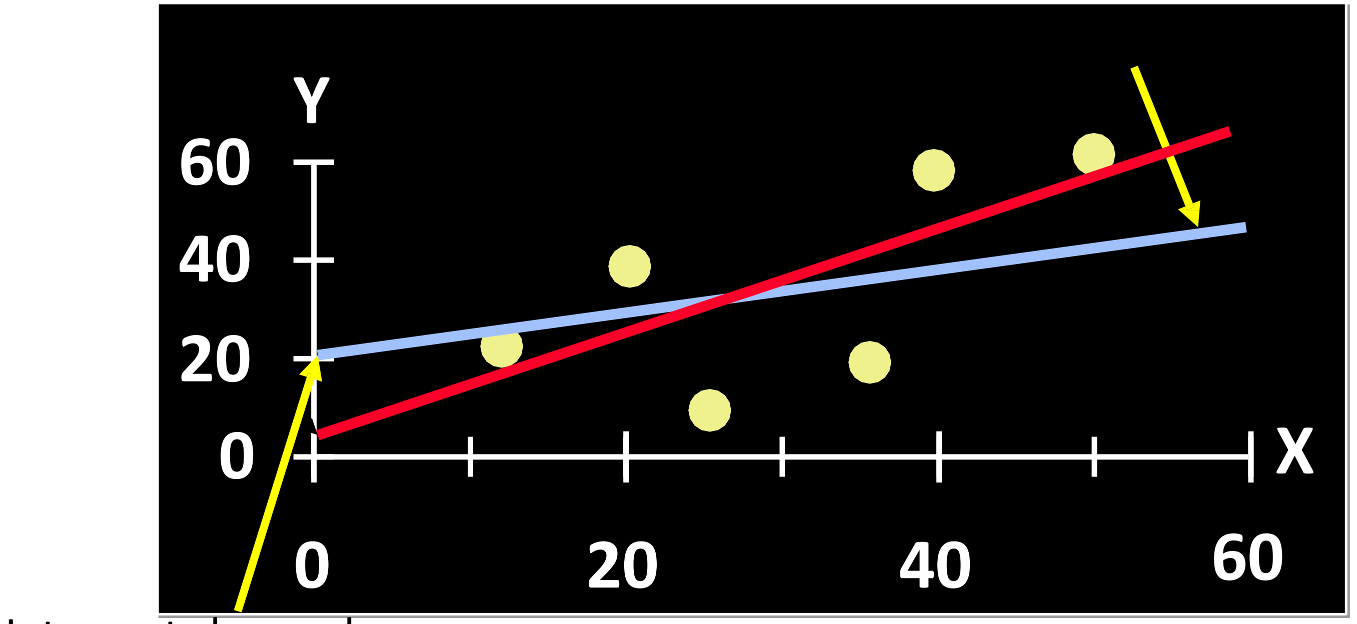
How would you draw a line through the points?
How do you determine which line 'fits best'?



Thinking Challenge

How would you draw a line through the points?
How do you determine which line 'fits best'?

Slope changed



Intercept changed

Least Squares

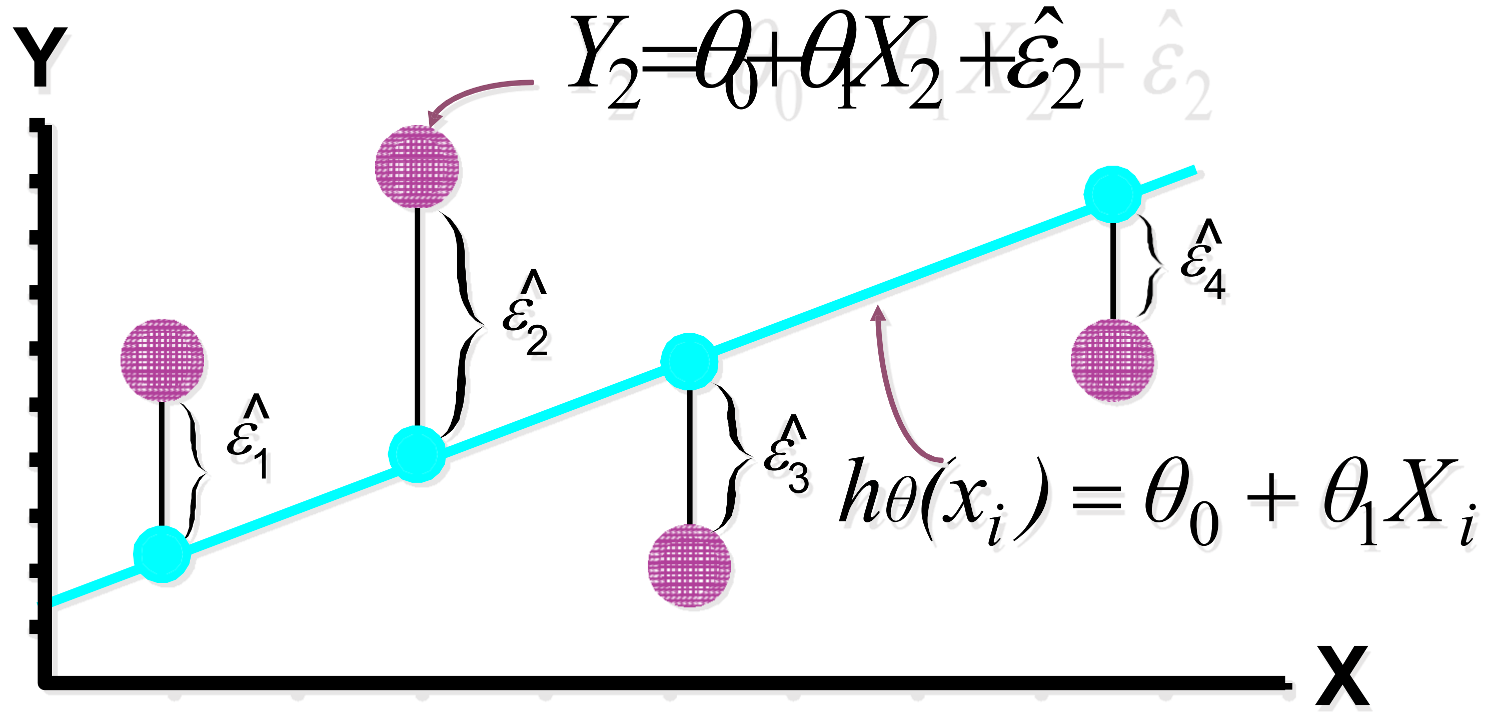
- 1. 'Best Fit' Means Difference Between Actual Y Values & Predicted Y Values Are a Minimum. So square errors!

$$\sum_{i=1}^{m}\left(Y_i - h\theta(x_i)\right)^2 = \sum_{i=1}^{m}\hat{\varepsilon}_i^{\,2}$$

- 2. LS Minimizes the Sum of the Squared Differences (errors) (SSE)

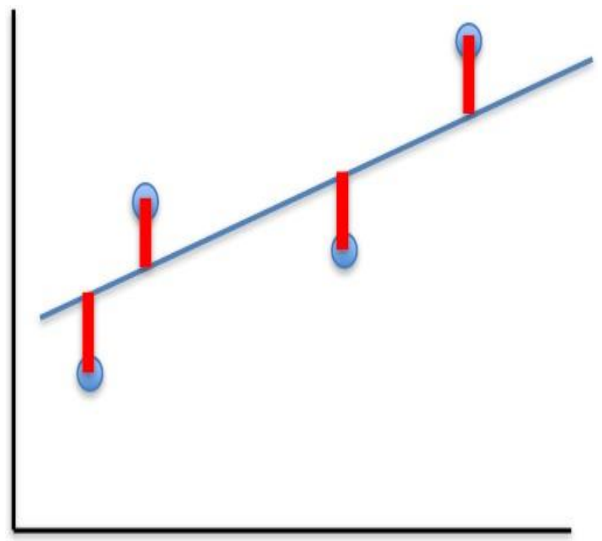
Least Squares Graphically

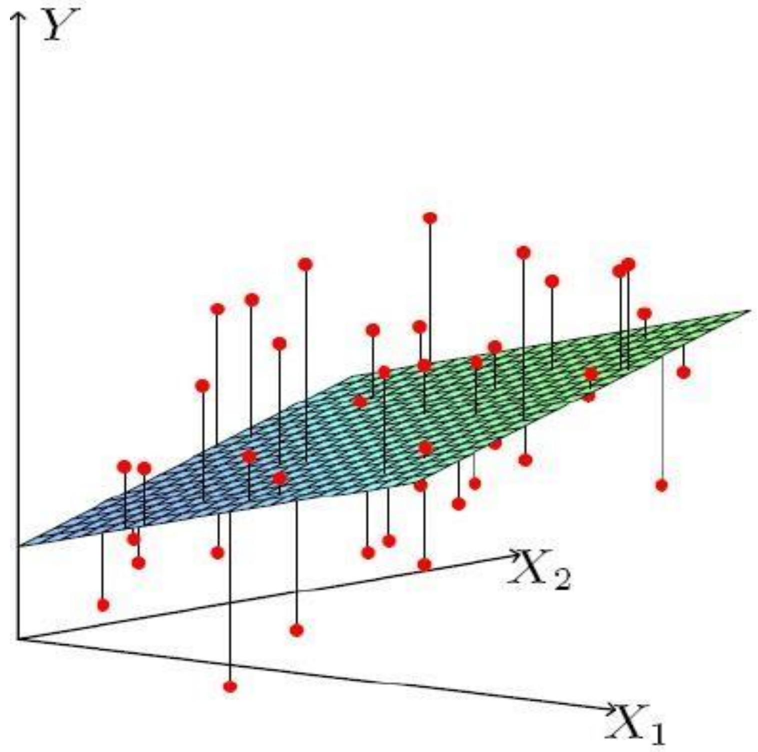
LS minimizes $\sum_{i=1}^{n} \hat{\varepsilon}_i^2 = \hat{\varepsilon}_1^2 + \hat{\varepsilon}_2^2 + \hat{\varepsilon}_3^2 + \hat{\varepsilon}_4^2$



$Y_2 = \theta_0 + \theta_1 X_2 + \hat{\varepsilon}_2$
Y
$\hat{\varepsilon}_1$
$\hat{\varepsilon}_2$
$\hat{\varepsilon}_3$
$\hat{\varepsilon}_4$
$h\theta(x_i) = \theta_0 + \theta_1 X_i$
X

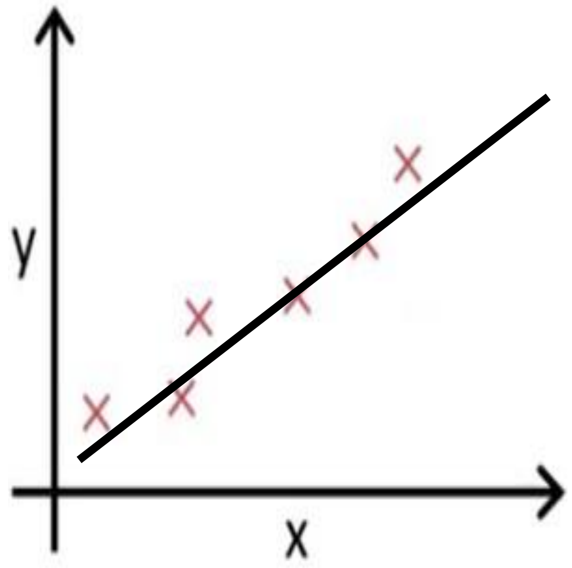
Least Squared errors Linear Regression





$$minimize_{\theta_0,\theta_1} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

COST FUNCTION



Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ is close to y for our training examples (x, y)

Minimize $\theta_0 \quad \theta_1$ $$\frac{1}{2m}\sum_{i}^{m}(h_\theta(x^i) - y^i)^2$$

$$\downarrow$$

$$h_\theta(x^i) = \theta_0 + \theta_1 x^i$$

$h_\theta(x^i)$ predictions on the training set

y^i the actual values

$$j(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i}^{m}(h_\theta(x^i) - y^i)^2$$

Minimize $\theta_0 \quad \theta_1$ $j(\theta_0, \theta_1)$

Cost function visualization

Consider a simple case of hypothesis by setting θ_0=0, then ***h*** becomes : $\mathbf{h_\theta(x)=\theta_1 x}$

Each value of θ_1 corresponds to a different hypothesis as it is the **slope** of the line

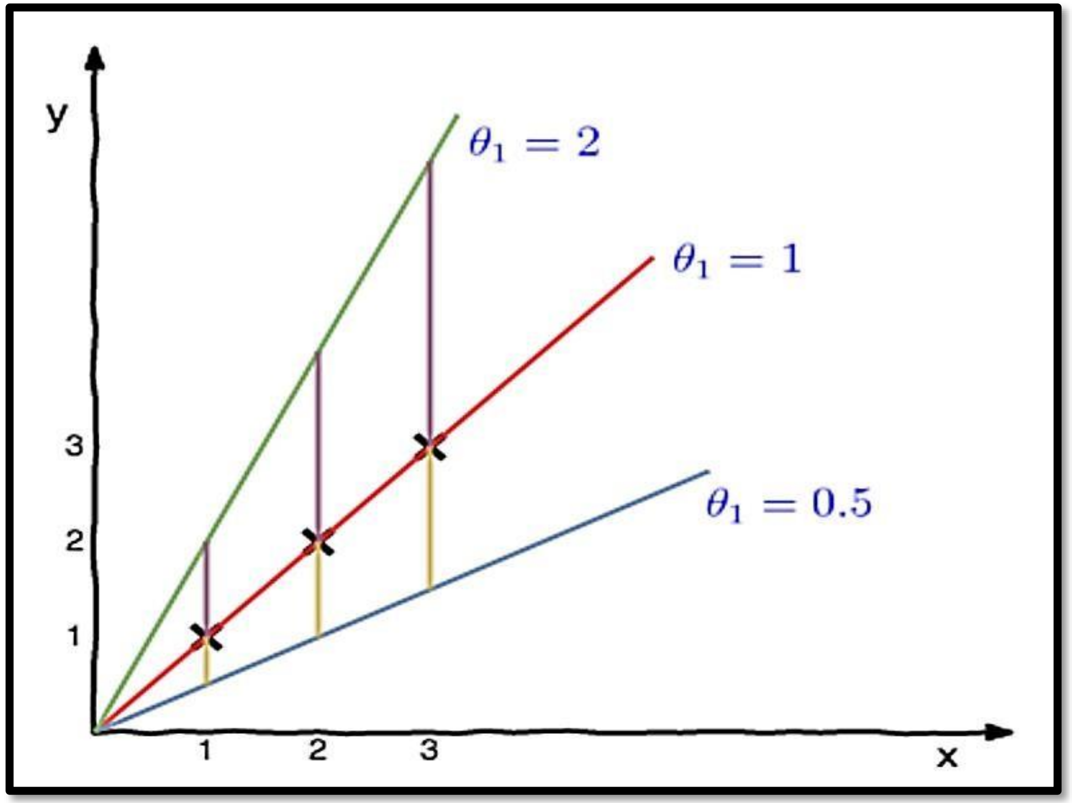
which corresponds to different lines passing through the **origin** as shown in plots below as **y-intercept** i.e. θ_0 is nulled out.

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

At θ_1=2, $J(2) = \frac{1}{2*3}(1^2 + 2^2 + 3^2) = \frac{14}{6} = 2.33$

At θ_1=1, $J(1) = \frac{1}{2*3}(0^2 + 0^2 + 0^2) = 0$

At θ_1=0.5, $J(0 = \frac{1}{2*3}(0.5^2 + 1^2 + 1.5^2) = 0.58$



Simple Hypothesis

Cost function visualization

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

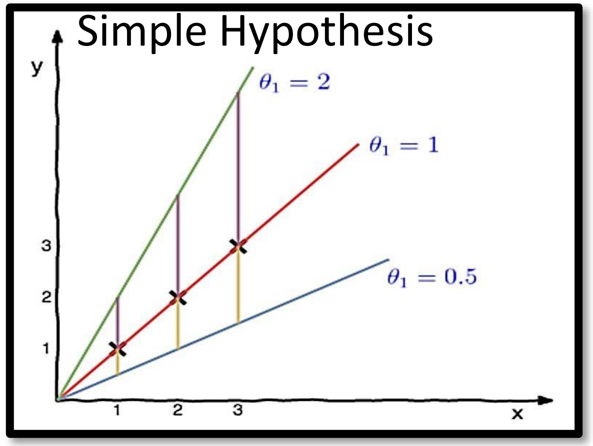
At θ_1=2, $J(2) = \frac{1}{2*3}(1^2 + 2^2 + 3^2) = \frac{14}{6} = 2.33$

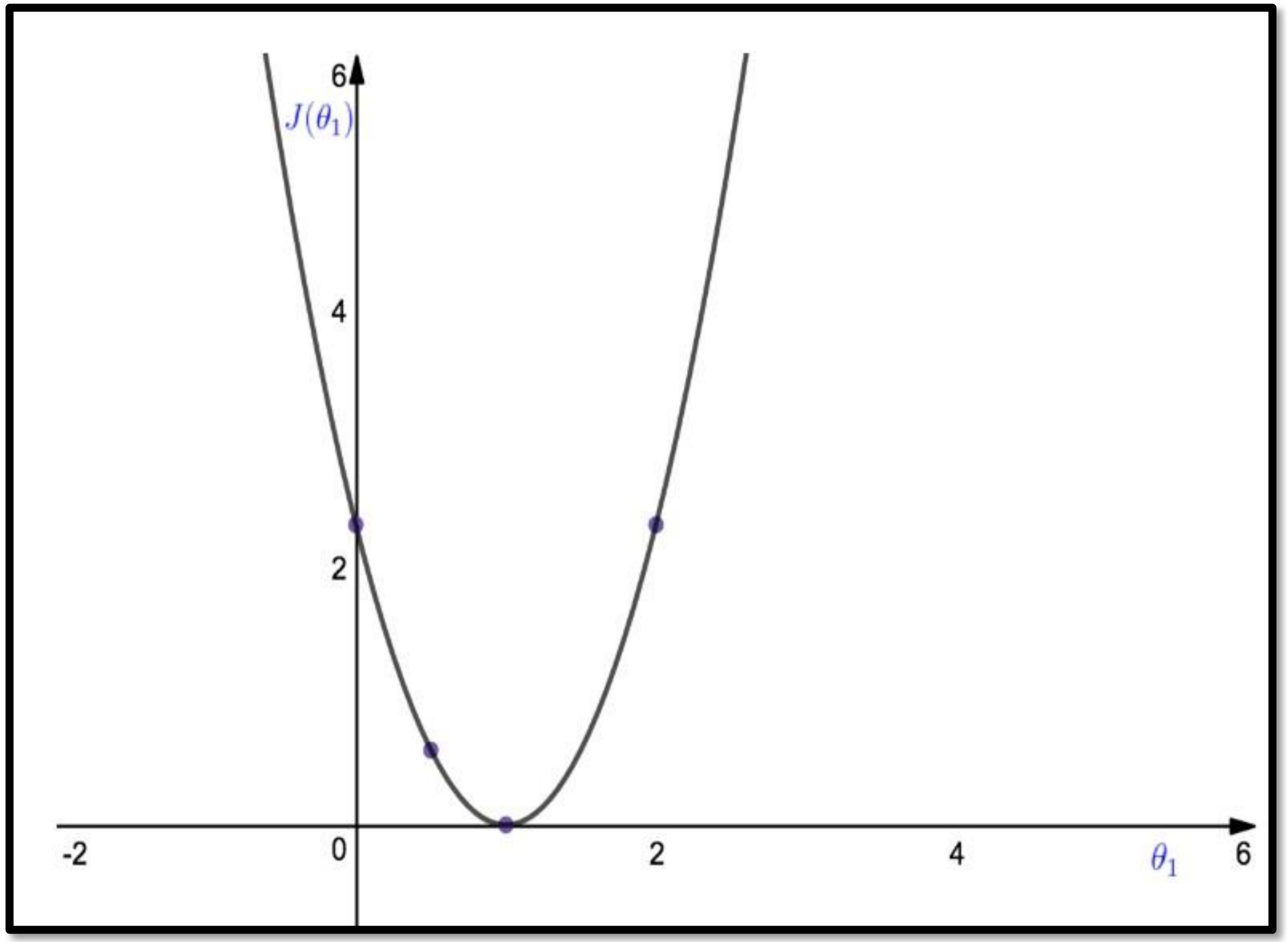
At θ_1=1, $J(1) = \frac{1}{2*3}(0^2 + 0^2 + 0^2) = 0$

At θ_1=0.5, $J(0.= \frac{1}{2*3}(0.5^2 + 1^2 + 1.5^2) = 0.58$

On **plotting points** like this further, one gets the following graph for the cost function which is dependent on parameter θ_1.

plot each value of θ_1 corresponds to a different hypothesizes





Cost function visualization

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)} \right)^2$$

What is the optimal value of θ_1 that minimizes $J(\theta_1)$?

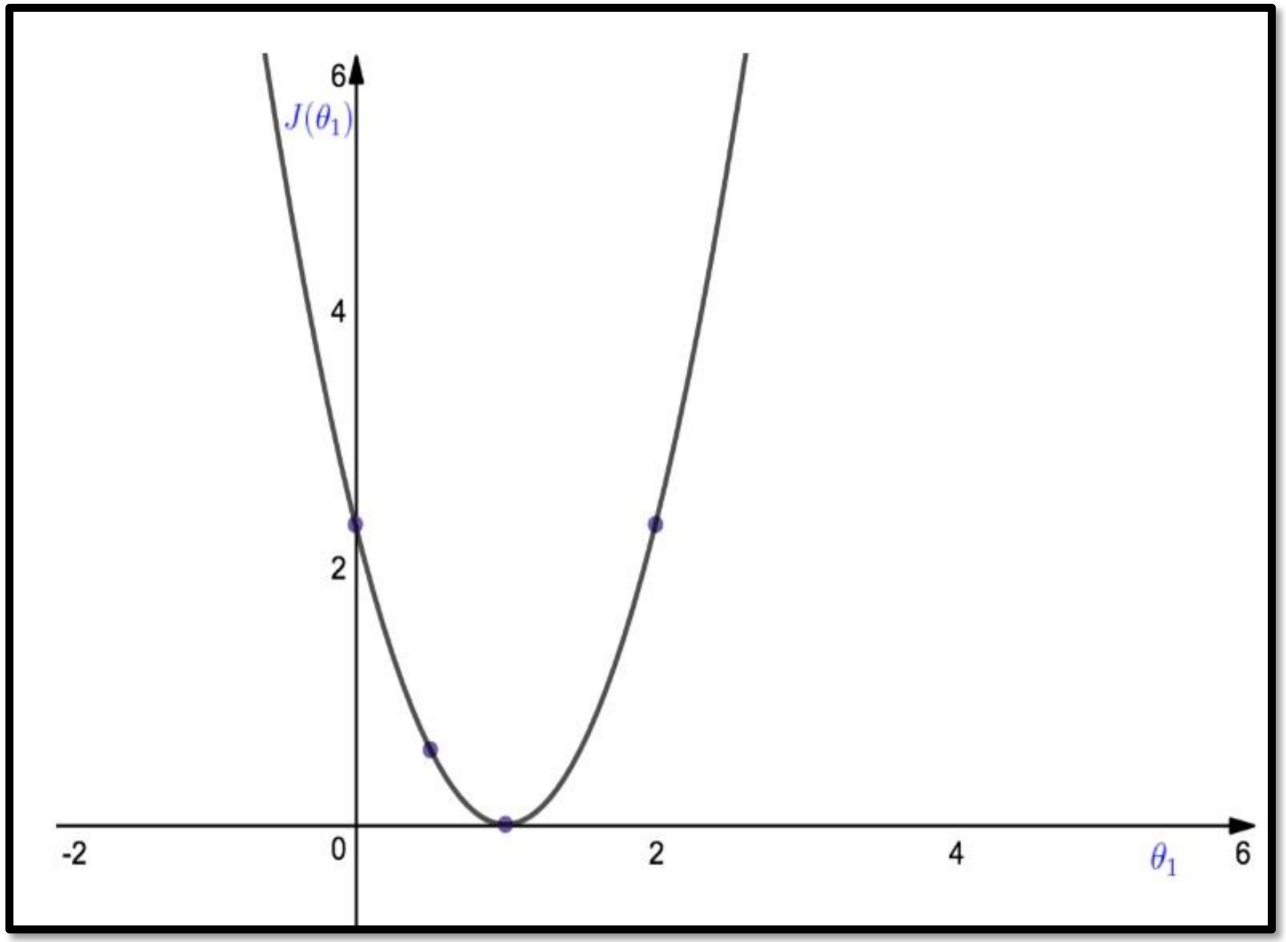
It is clear that best value for $\theta_1 = 1$ as $J(\theta_1) = 0$, which is the minimum.

How to find the best value for θ_1 ?

Plotting ?? Not practical specially in high dimensions?

The solution :

1. **Analytical solution: not applicable for large datasets**
2. **Numerical solution: ex: Gradient descent .**



COST FUNCTION (RECAP)

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost Function:

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$

Gradient Descent

GRADIENT DESCENT

- Iterative solution not only in linear regression. It's actually used all over the place in machine learning.

- Objective: minimize any function (Cost Function J)

PROBLEM SETUP

Have some function $J(\theta_0, \theta_1)$

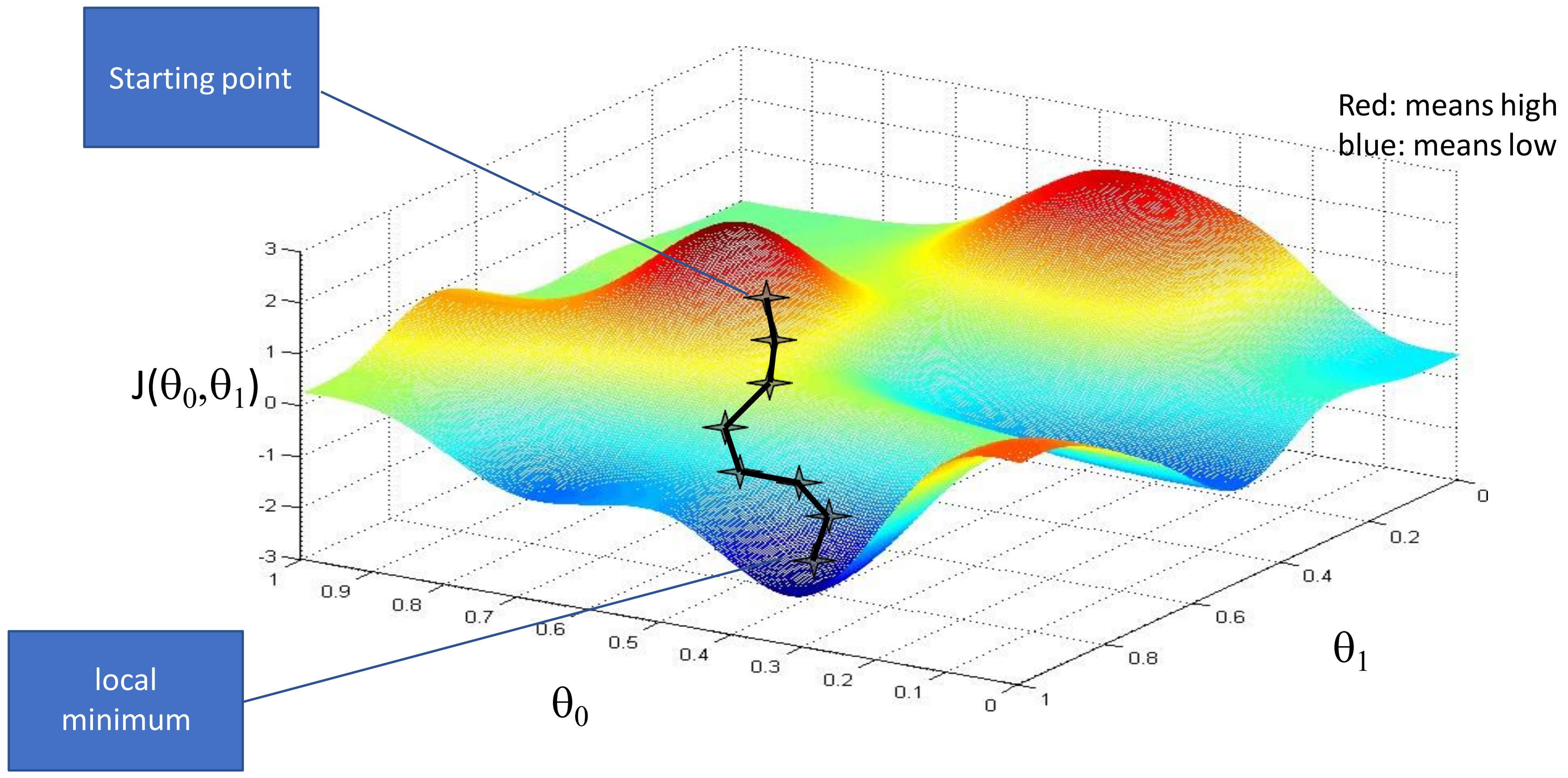
Want $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$

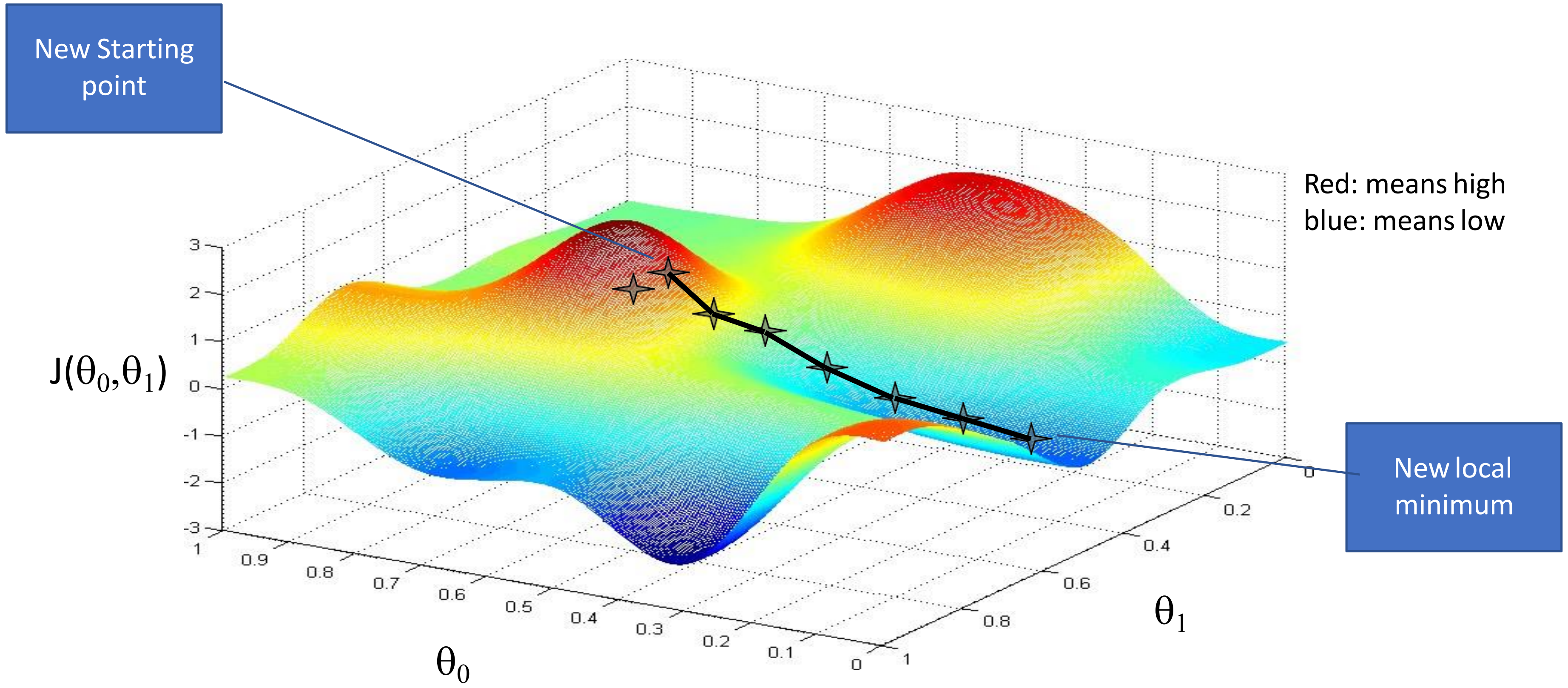
Outline:

- Start with some θ_0, θ_1
- Keep changing θ_0, θ_1 to reduce $J(\theta_0, \theta_1)$

  until we hopefully end up at a minimum

Imagine that this is a landscape of grassy park, and you want to go to the lowest point in the park as rapidly as possible





With different starting point

Gradient descent Algorithm

$$\text{repeat until convergence}\{\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \, \forall j \in \{0,1\}\}$$

- Where
    - := is the assignment operator
    - α is the **learning rate** which basically defines how big the steps are during the descent
    - $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ is the **partial derivative** term
    - j = 0, 1 represents the **feature index number**

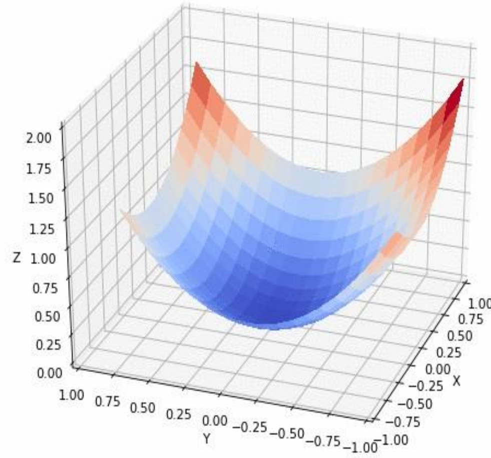
Also the parameters should be **updated simulatenously**, i.e. ,

$$temp_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$temp_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$\theta_0 := temp_0$$

$$\theta_1 := temp_1$$



GRADIENT DESCENT FOR A LINEAR REGRESSION

Gradient descent algorithm

$$\text{repeat until convergence } \{$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

$$(\text{for } j = 1 \text{ and } j = 0)$$

$$\}$$

Linear Regression Model

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$\boxed{\frac{d}{d\theta_j} j(\theta_0,\theta_1) = \frac{d}{d\theta_j}\frac{1}{2m}\sum_{i=1}^{m}\left(h\theta(x_i) - Y_i\right)^2}$$

$$\boxed{\frac{d}{d\theta_j} j(\theta_0,\theta_1) = \frac{d}{d\theta_j}\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0 + \theta_1 (x_i) - Y_i\right)^2}$$

$$j = 0 : \frac{d}{d\theta_0} j(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h\theta(x_i) - Y_i\right)$$

$$j = 1 : \frac{d}{d\theta_1} j(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h\theta(x_i) - Y_i\right) \bullet x_i$$

G.D. FOR LINEAR REGRESSION

Gradient descent algorithm

```
repeat until convergence {
    $\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$
    $\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$
}
```

“Batch” Gradient Descent

“Batch”: Each step of gradient descent uses all the training examples.

```
repeat until convergence {
    $\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$
    $\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) \cdot x^{(i)}$
}
```

Example after implement some iterations using gradient descent

Performance of Regression Models (Mean Squared Error)

Mean Squared Error (MSE) is a popular metric used to evaluate the performance of a regression model. It measures the average of the squared differences between the predicted and actual values.

💡 Formula for MSE:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Where:

- n = number of data points (samples).
- y_i = actual value.
- $\hat{y}_i$ = predicted value.

Performance of Regression Models (Mean Absolute Error)

Another option is the **Mean Absolute Error (MAE)**, which can be seen as a measure of "how accurate" the model is based on absolute deviations from the actual values (without squaring the differences, like MSE).

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Performance of Regression Models (R-Squared)

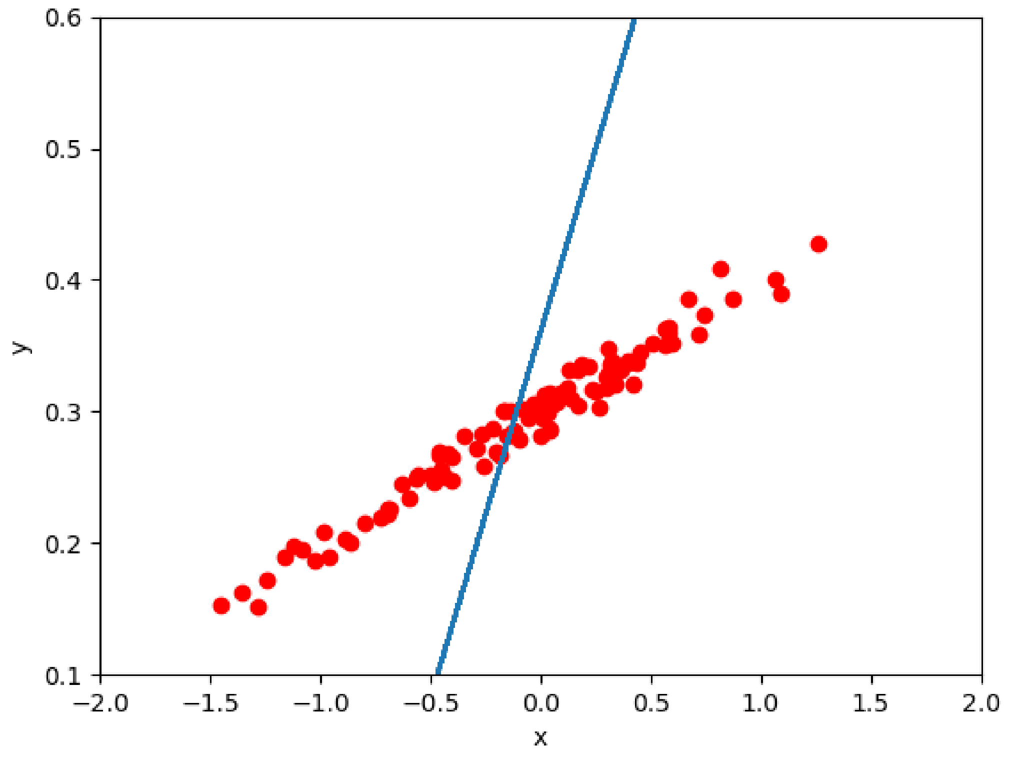
The **R^2** value is a measure of how well the regression model explains the variability of the target variable. It gives you an idea of **how much of the variance in the data is explained by the model**.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

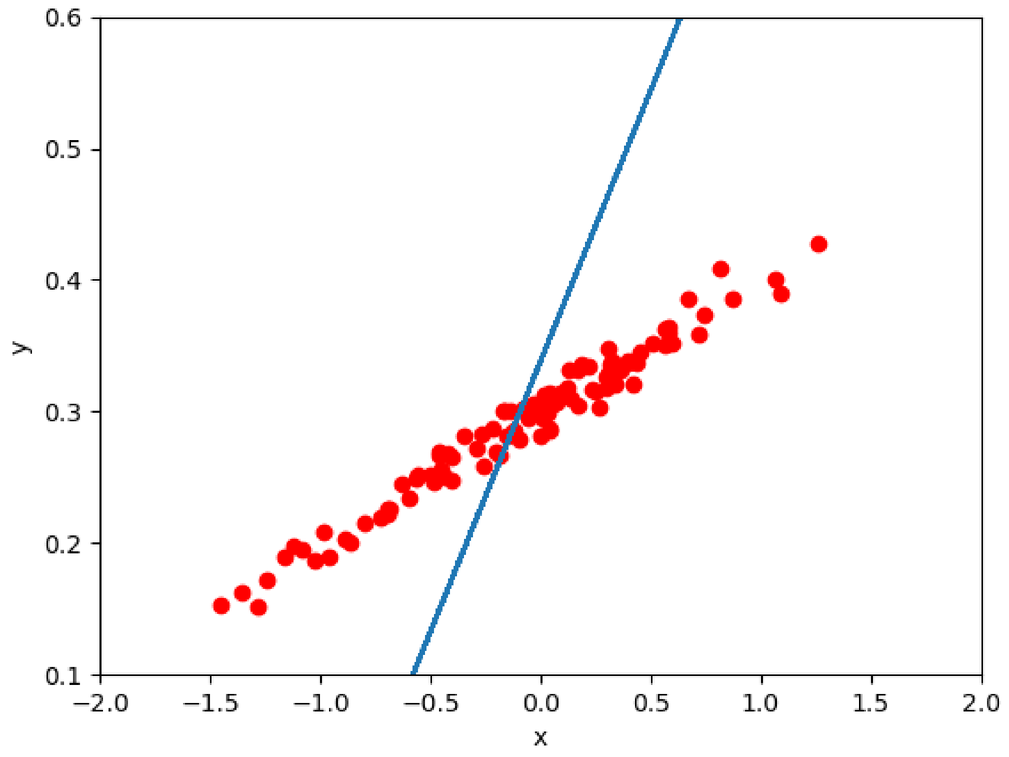
Where:

- y_i = Actual values.
- $\hat{y}_i$ = Predicted values.
- $\bar{y}$ = Mean of the actual values.
- **R^2 = 1**: Perfect fit — all points lie on the regression line.
- **R^2 = 0**: The model does not explain any of the variance (equivalent to using the mean as a predictor).
- **Negative R^2**: The model is worse than simply predicting the mean value of the target.

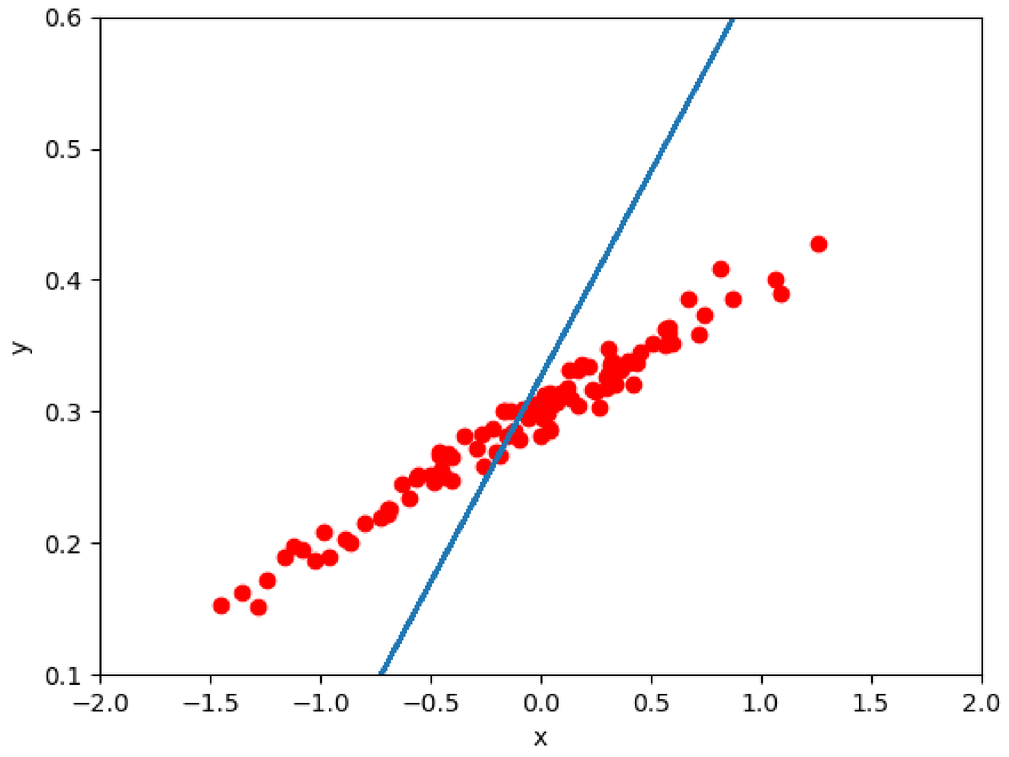
Iteration 1



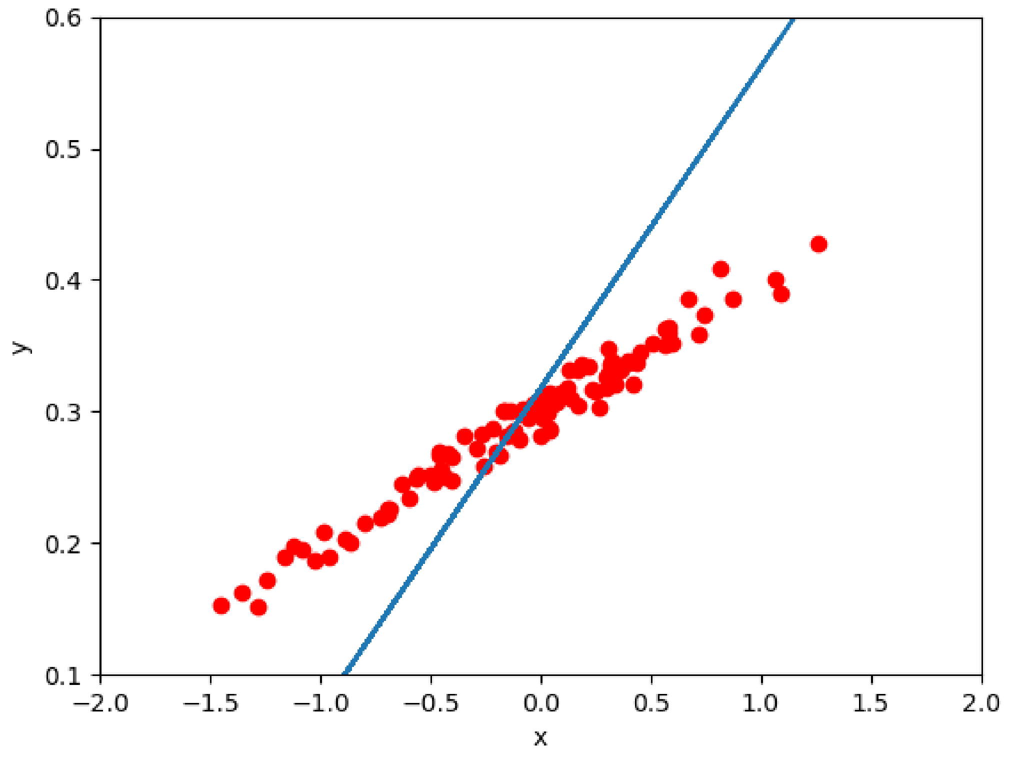
Iteration 2



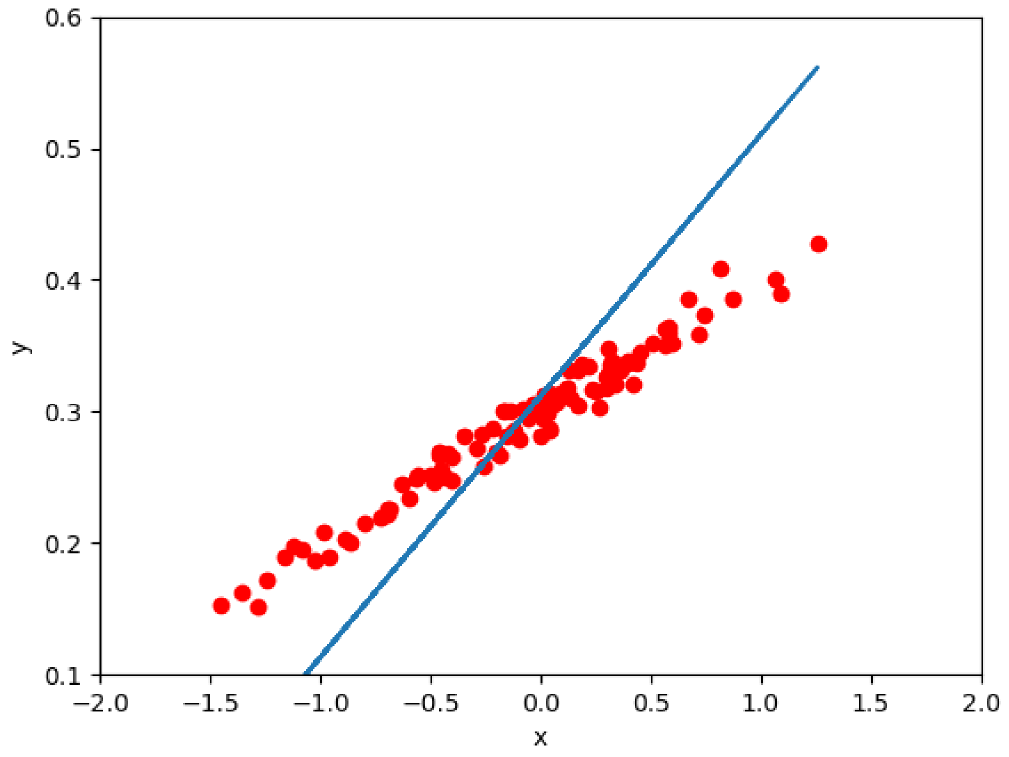
Iteration 3



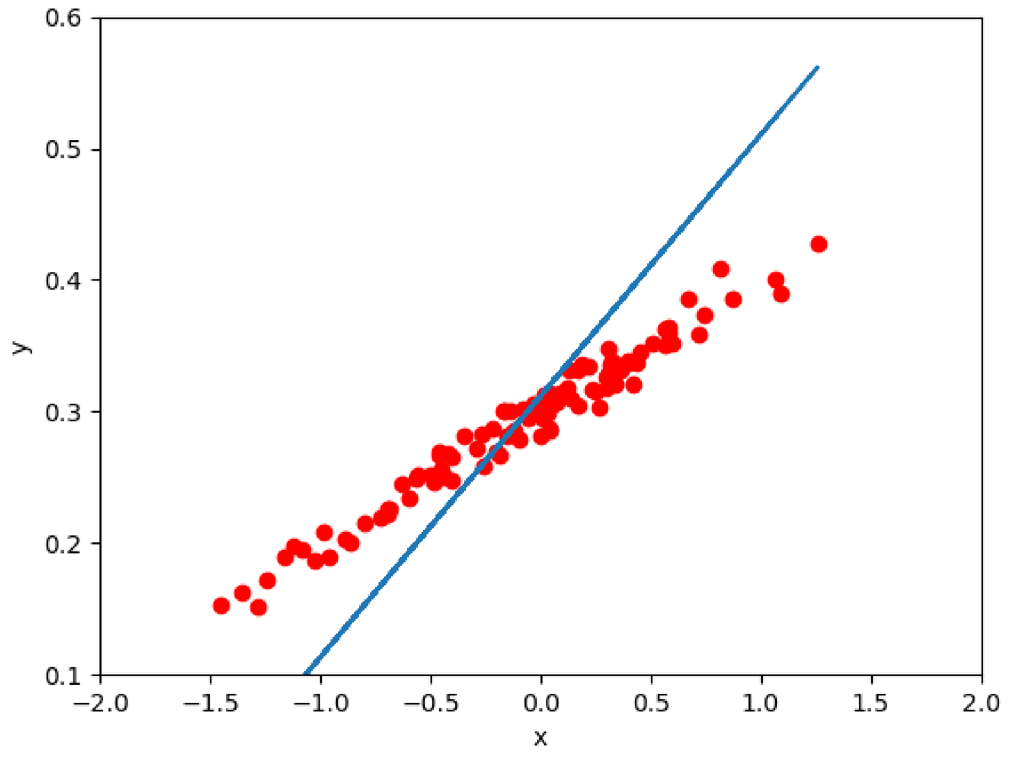
Iteration 4



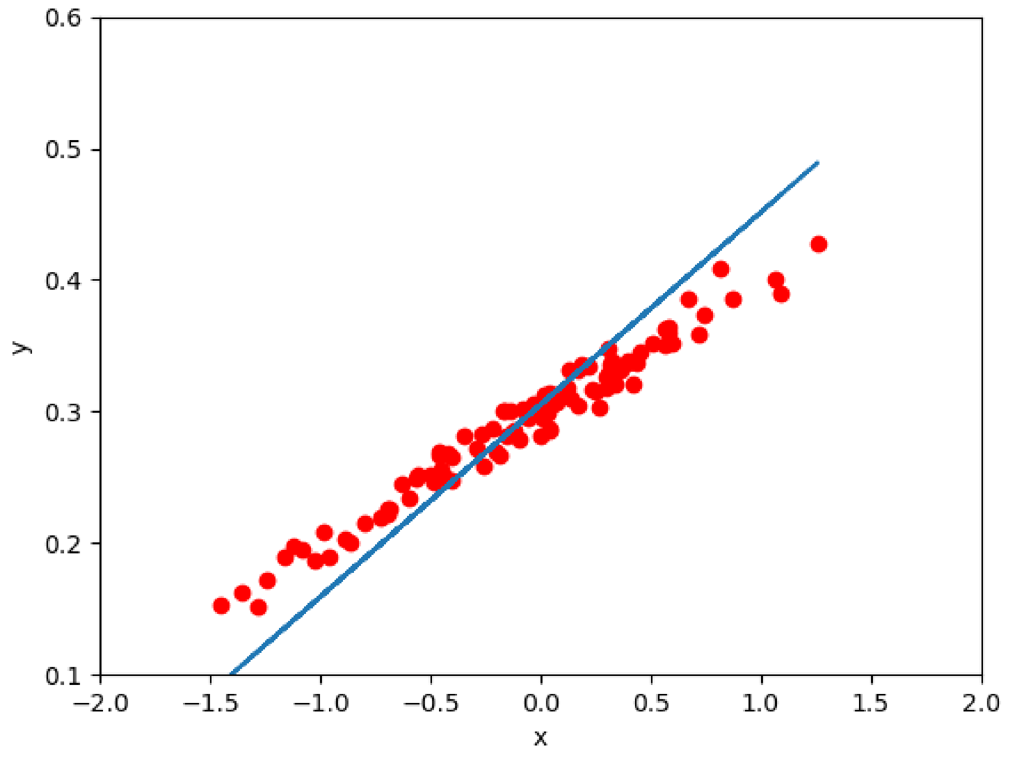
Iteration 5



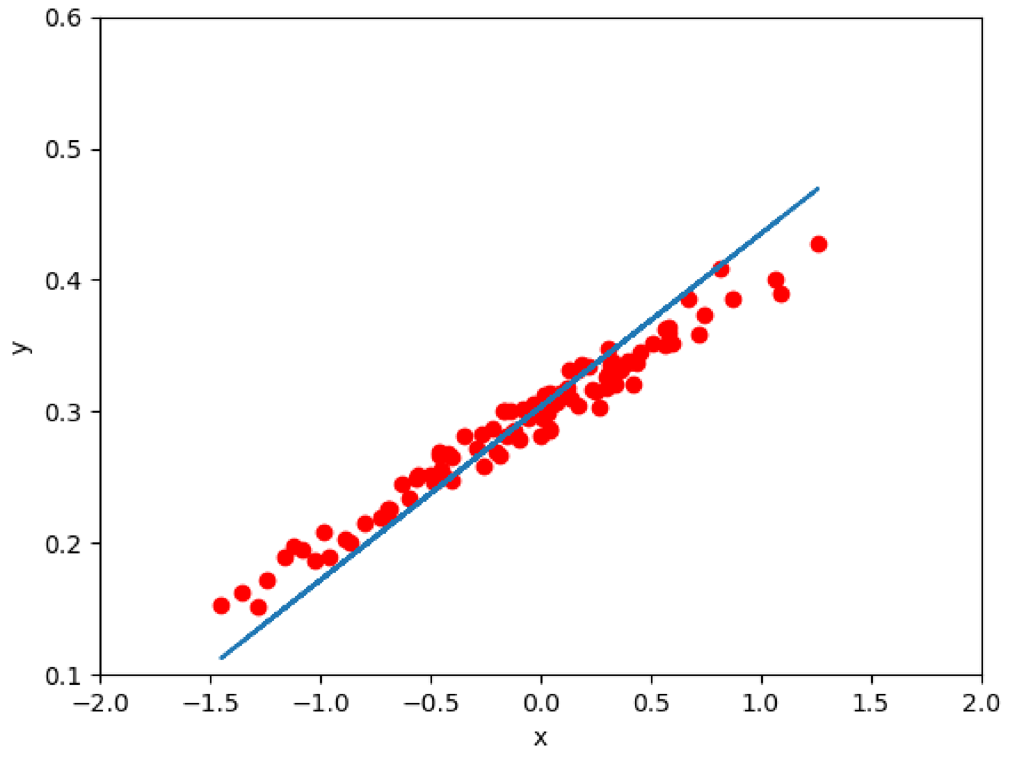
Iteration 6



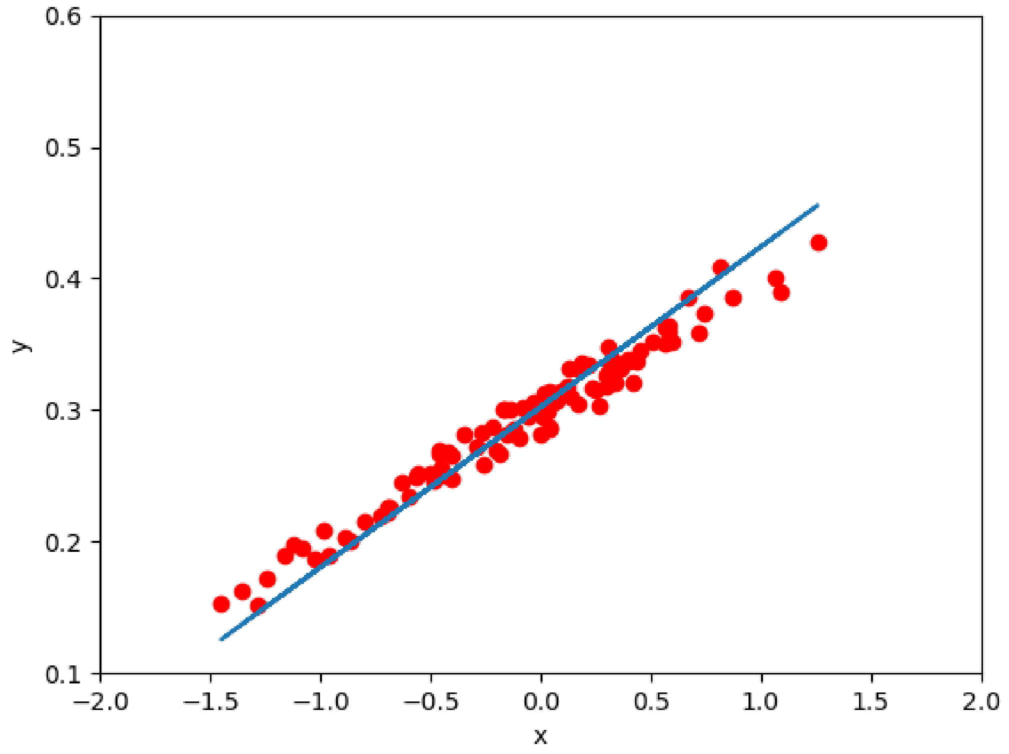
Iteration 7



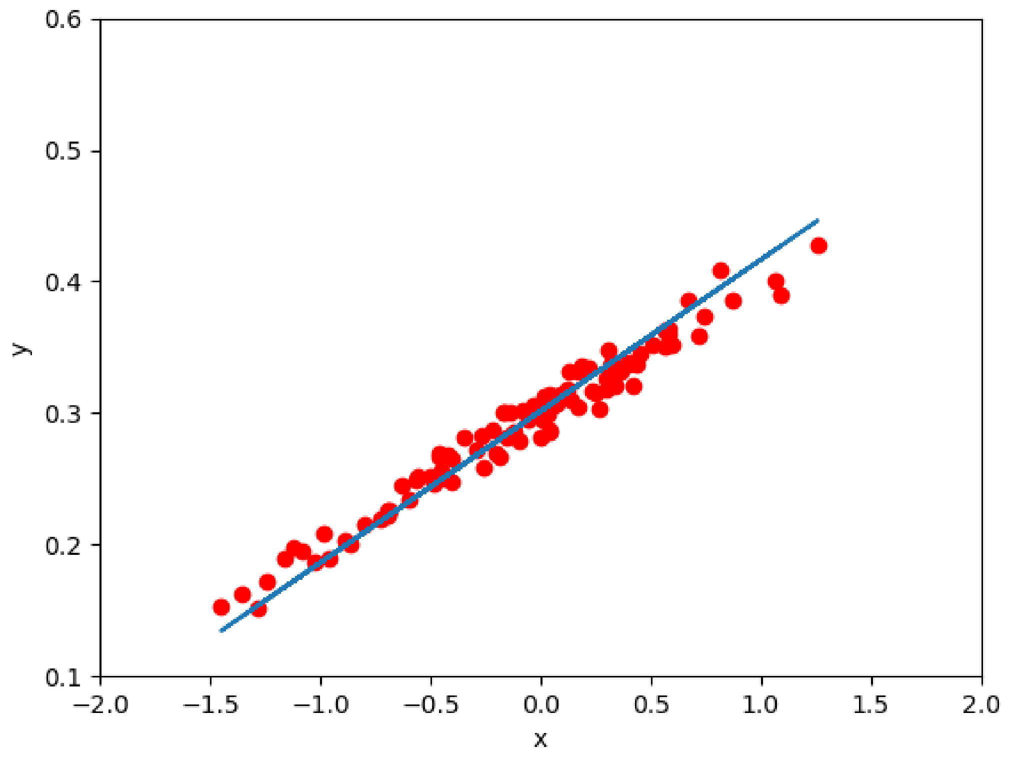
Iteration 8



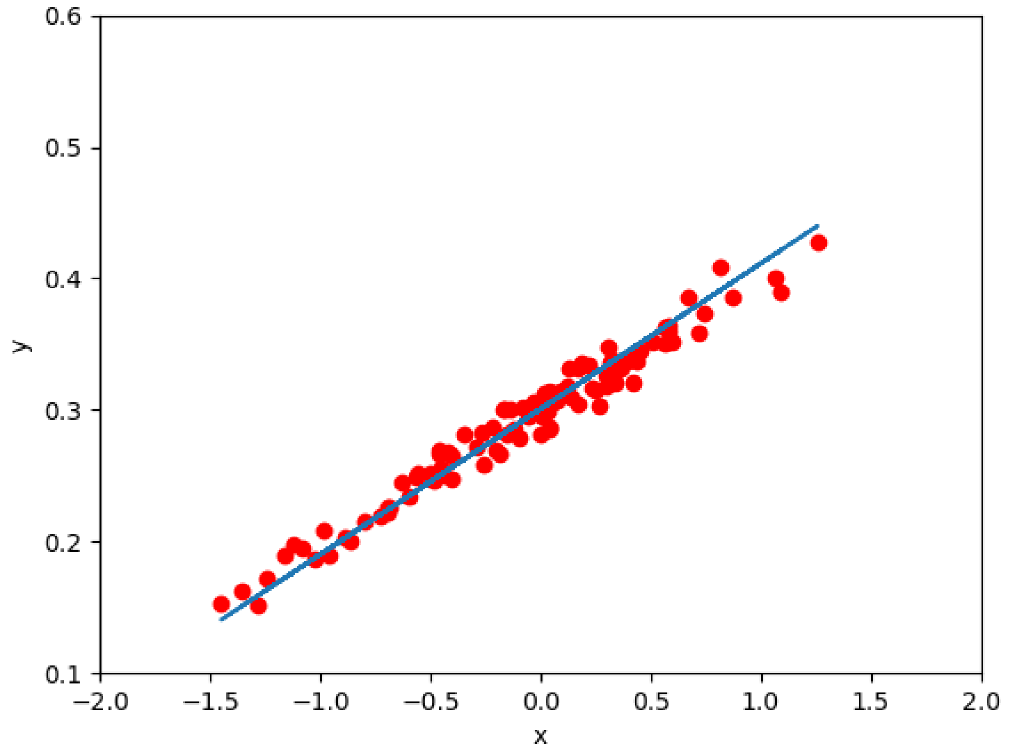
Iteration 9



Iteration 10



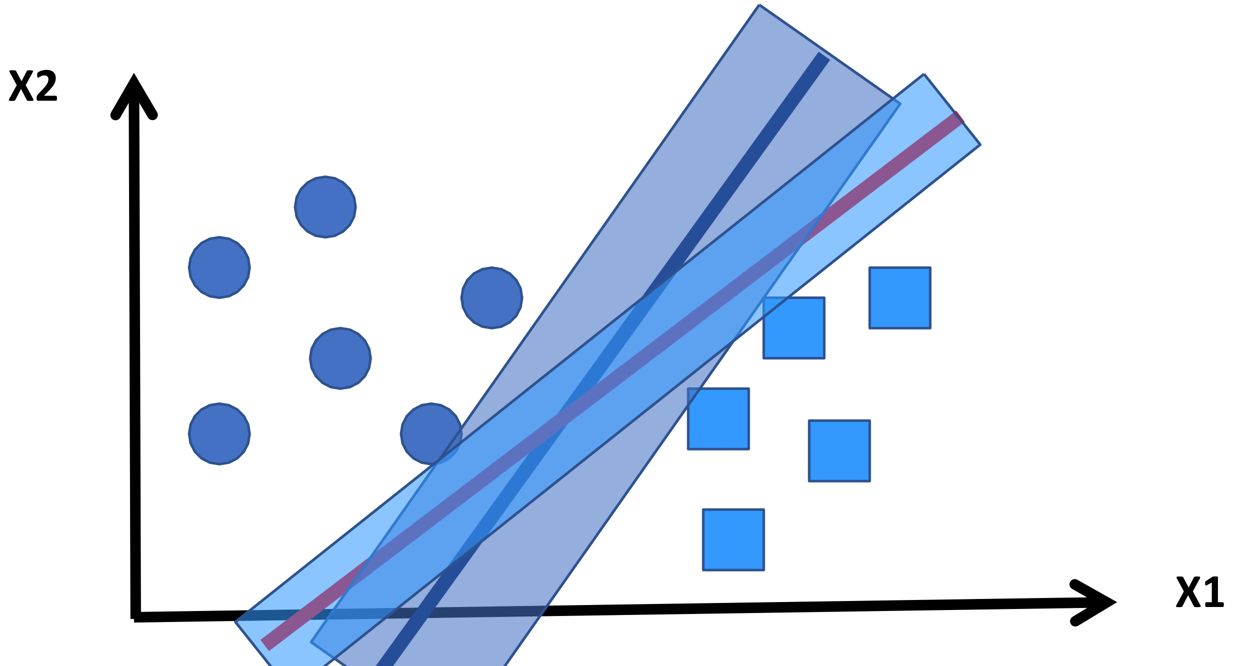
Iteration 11



Support Vector Machine (SVM)

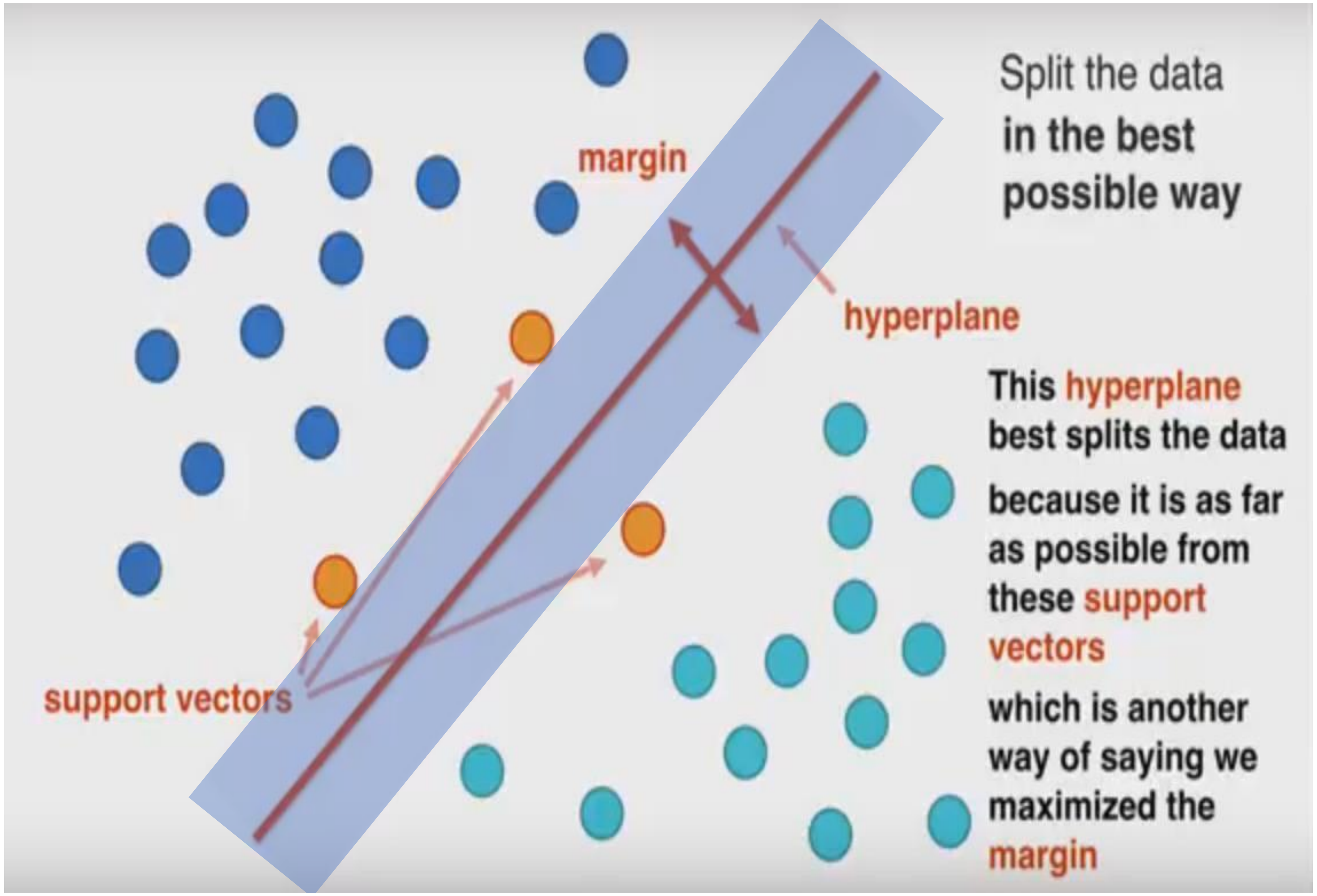
overview

- SVM for **linearly separable** binary set
- **Main Goal** to design a hyper plane that classify all training vectors into two classes
- *The best model* **that leaves the maximum margin** from both classes
- the two classes labels **+1** (positive examples and **-1** (negative examples)



overview

This is a constrained optimization problem

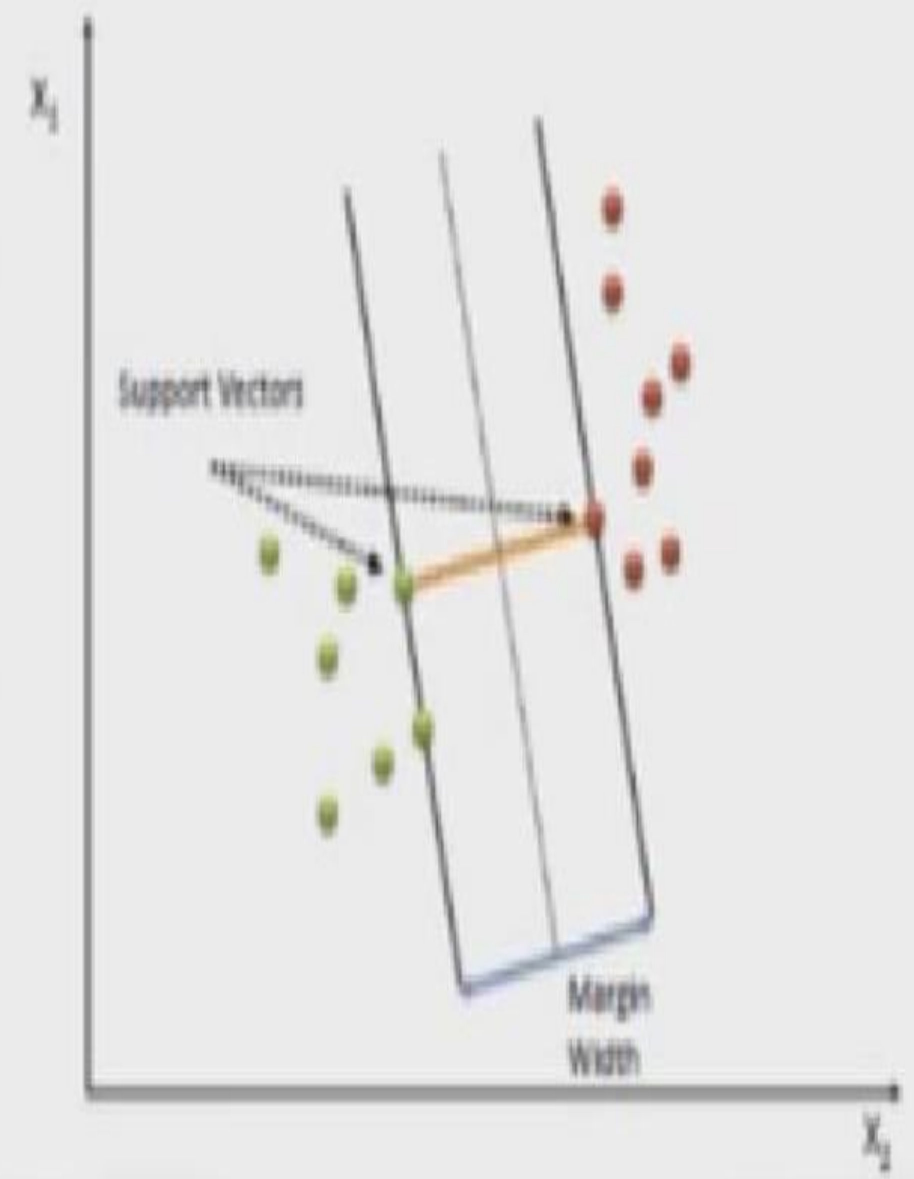


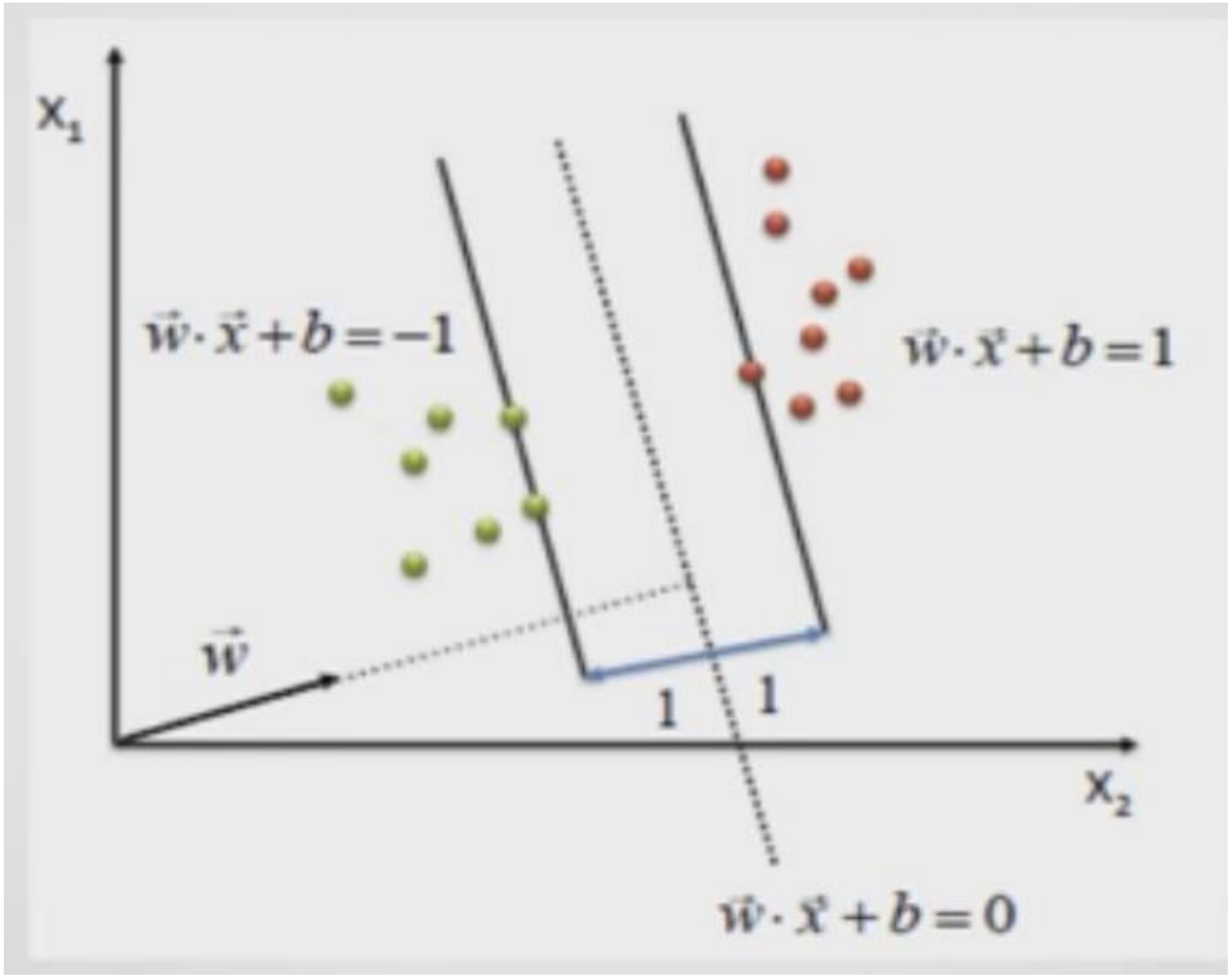
Split the data in the best possible way

This hyperplane best splits the data because it is as far as possible from these support vectors which is another way of saying we maximized the margin

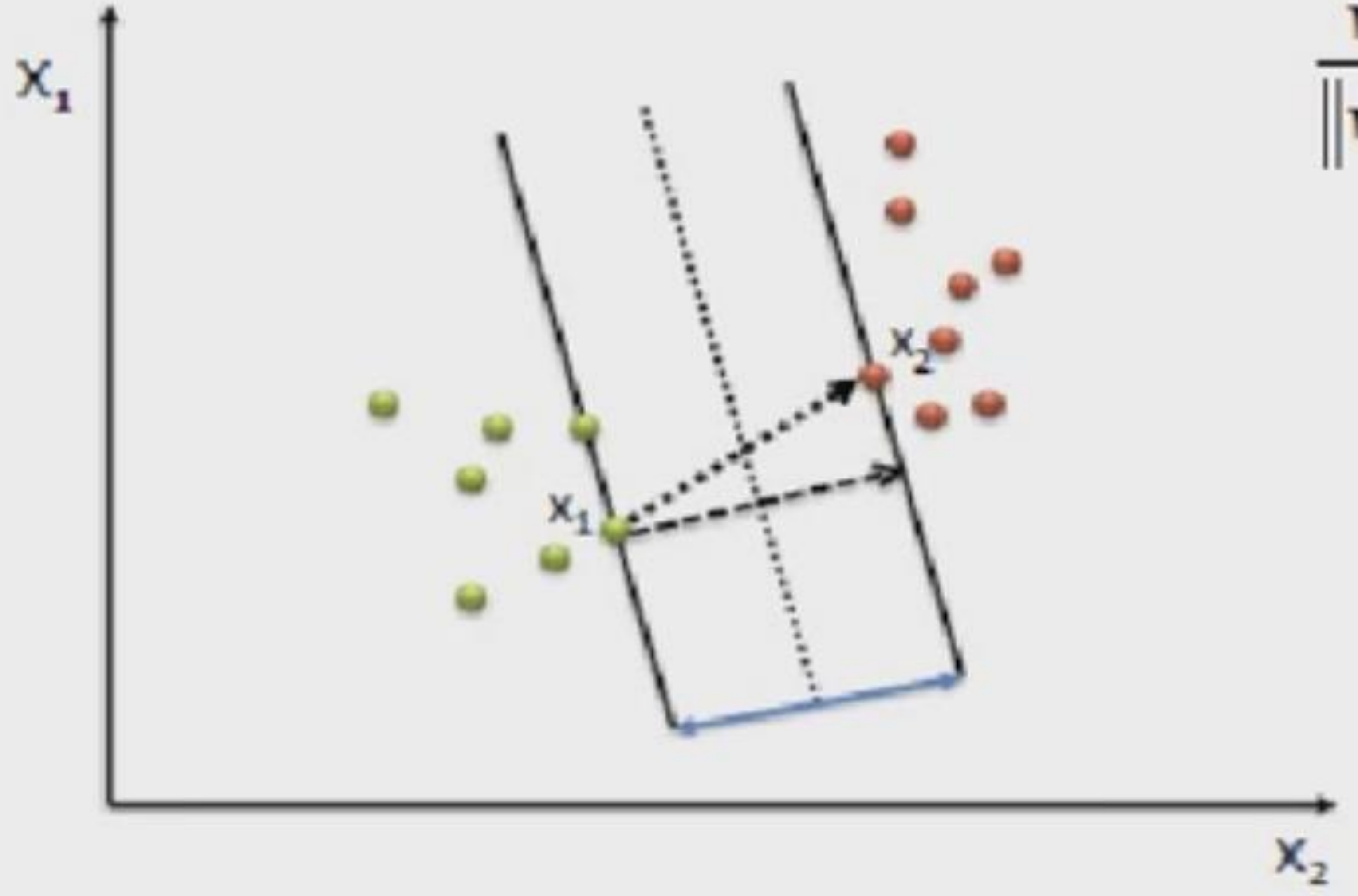
Intuition behind SVM

- Points (instances) are like vectors p = (x1,x2,...,xn)
- SVM finds the closest two points from the two classes (see figure), that support (define) the best separating line|plane
- Then SVM draws a line connecting them (the orange line in the figure)
- After that, SVM decides that the best separating line is the line that bisects, and is perpendicular to, the connecting line




x_1
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
x_2
$\vec{w}\cdot\vec{x}+b=0$

Margin in terms ofW



$$\frac{w}{\|w\|}\cdot(x_2 - x_1) = \text{width} = \frac{2}{\|w\|}$$

$$w \cdot x_2 + b = 1$$

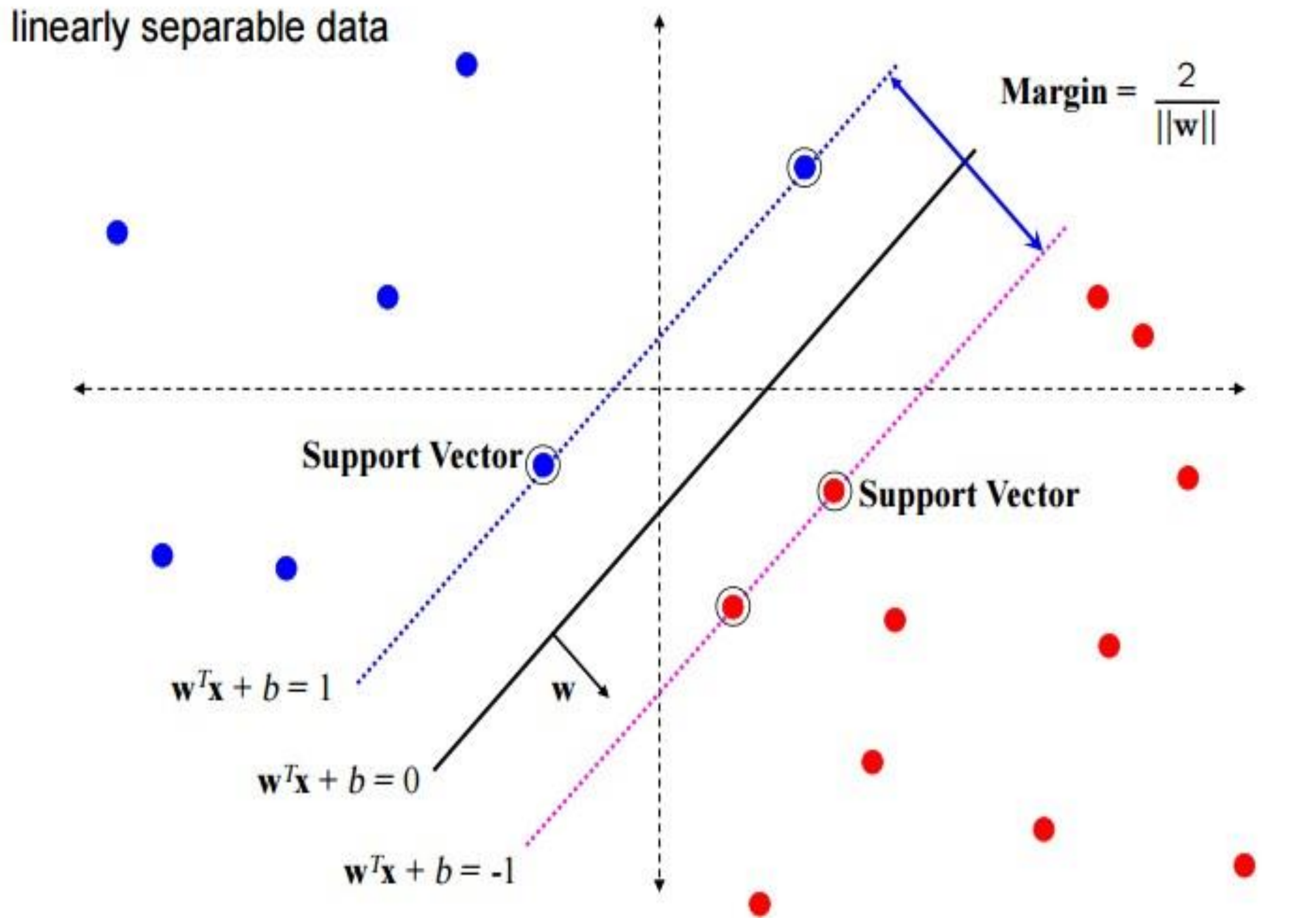
$$w \cdot x_1 + b = -1$$

$$w \cdot x_2 + b - w \cdot x_1 - b = 1 - (-1)$$

$$w \cdot x_2 - w \cdot x_1 = 2$$

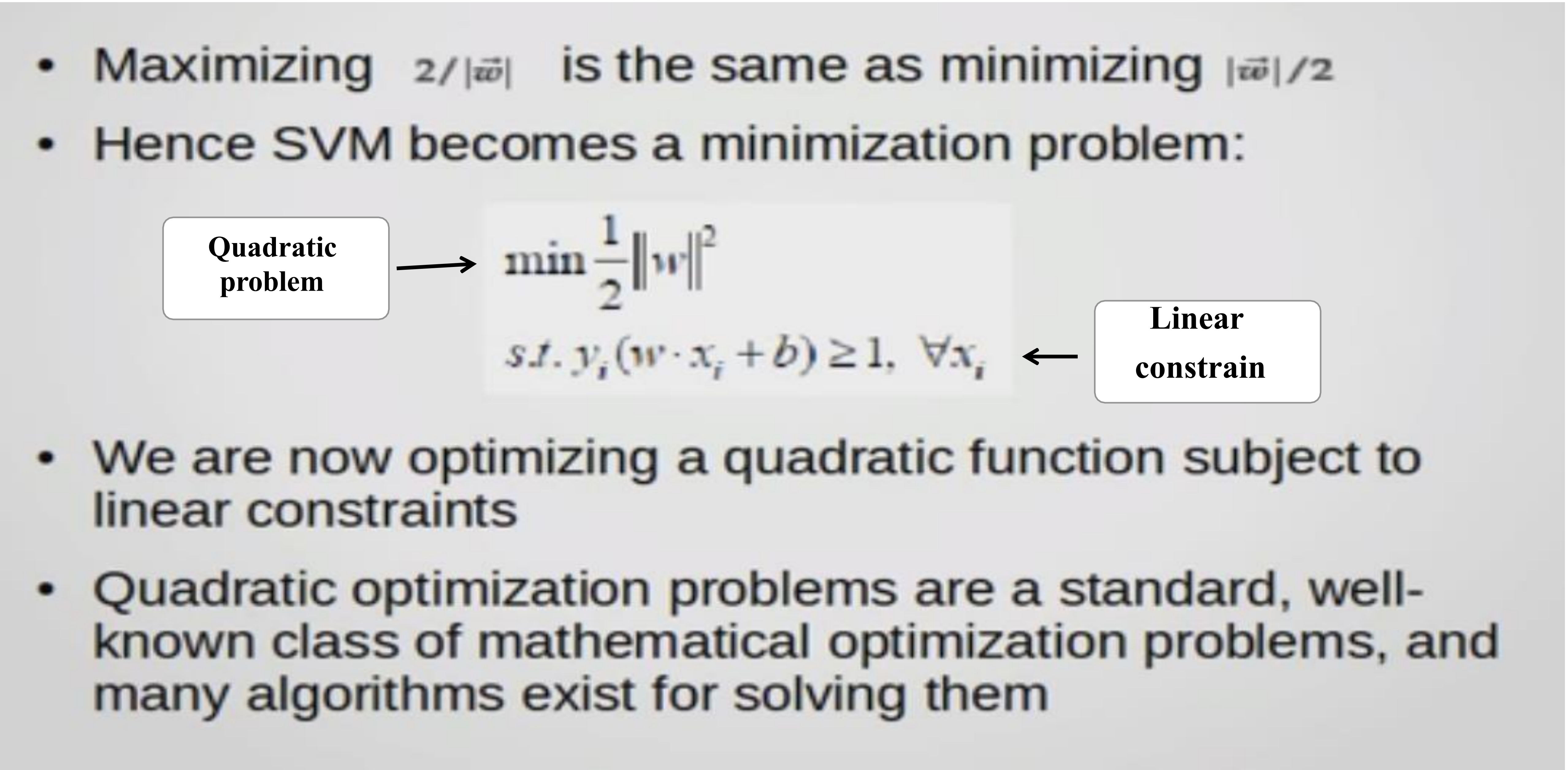
$$\frac{w}{\|w\|}(x_2 - x_1) = \frac{2}{\|w\|}$$

Support Vector Machine



linearly separable data
Margin = $\frac{2}{||\mathbf{w}||}$
Support Vector
Support Vector
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Svm as a minimization problem



- Maximizing $2/|\vec{w}|$ is the same as minimizing $|\vec{w}|/2$
- Hence SVM becomes a minimization problem:

Quadratic problem $\longrightarrow$

$$\min \frac{1}{2}\|w\|^2$$

$$s.t.\ y_i(w \cdot x_i + b) \geq 1,\ \forall x_i$$

$\longleftarrow$ Linear constrain

- We are now optimizing a quadratic function subject to linear constraints
- Quadratic optimization problems are a standard, well-known class of mathematical optimization problems, and many algorithms exist for solving them

$$\min \frac{1}{2}\|\mathbf{w}\|^2 \qquad \text{s.t.} \qquad y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0 \;\; \forall_i$$

In order to cater for the constraints in this minimization, we need to allocate them Lagrange multipliers $\boldsymbol{\alpha}$, where $\alpha_i \geq 0 \; \forall_i$:

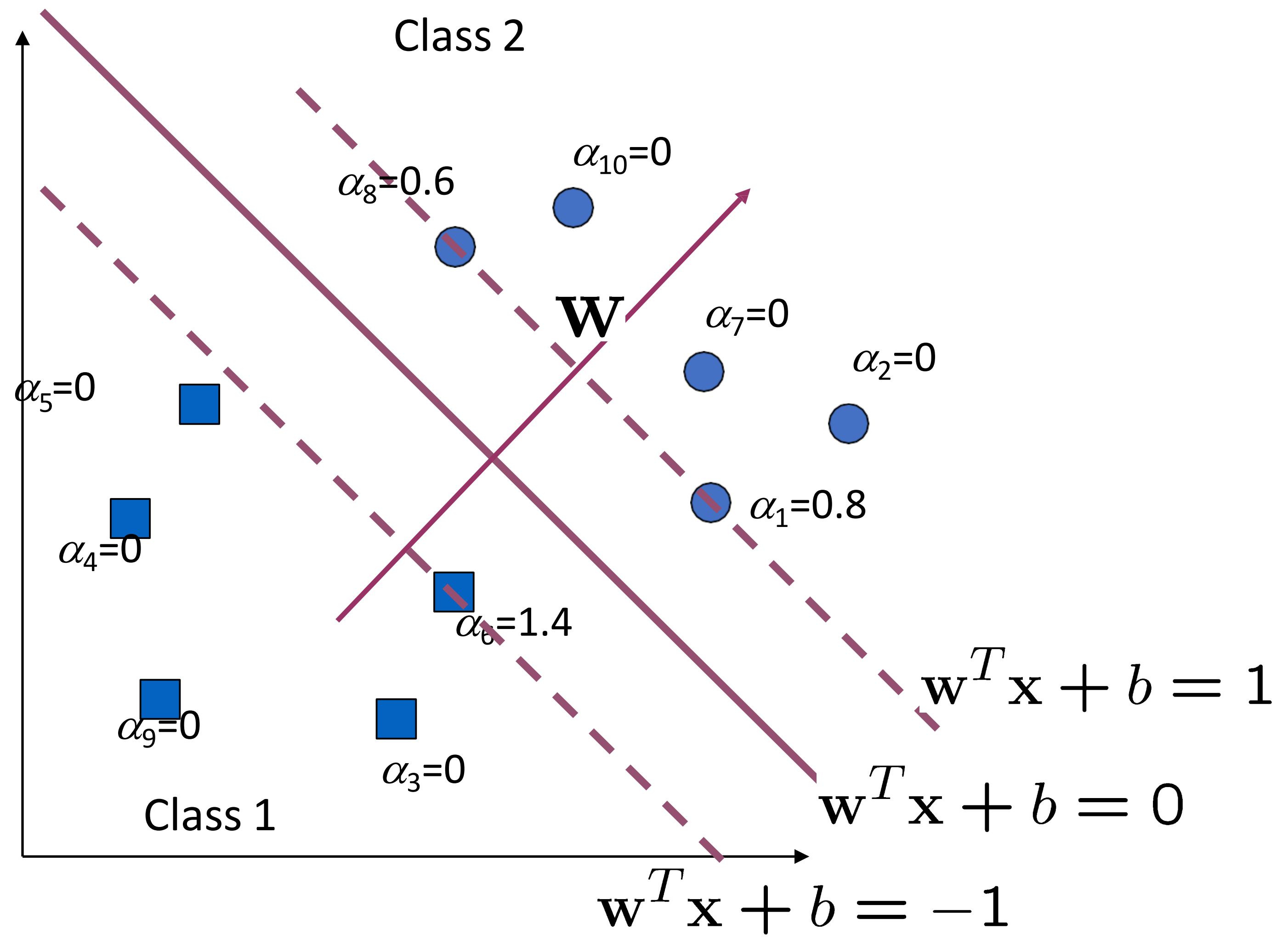
$$\begin{aligned} L_P &\equiv \frac{1}{2}\|\mathbf{w}\|^2 - \boldsymbol{\alpha}\left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \;\forall_i\right] \\ &\equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{L} \alpha_i \left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1\right] \\ &\equiv \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{L} \alpha_i y_i(\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{L} \alpha_i \end{aligned}$$

We wish to find the **w** and **b** which minimizes, and the **α** which maximizes LP(whilst keeping αi ≥ 0 ∀i). We can do this by differentiating LP with respect to w and b and setting the derivatives to zero:

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i$$

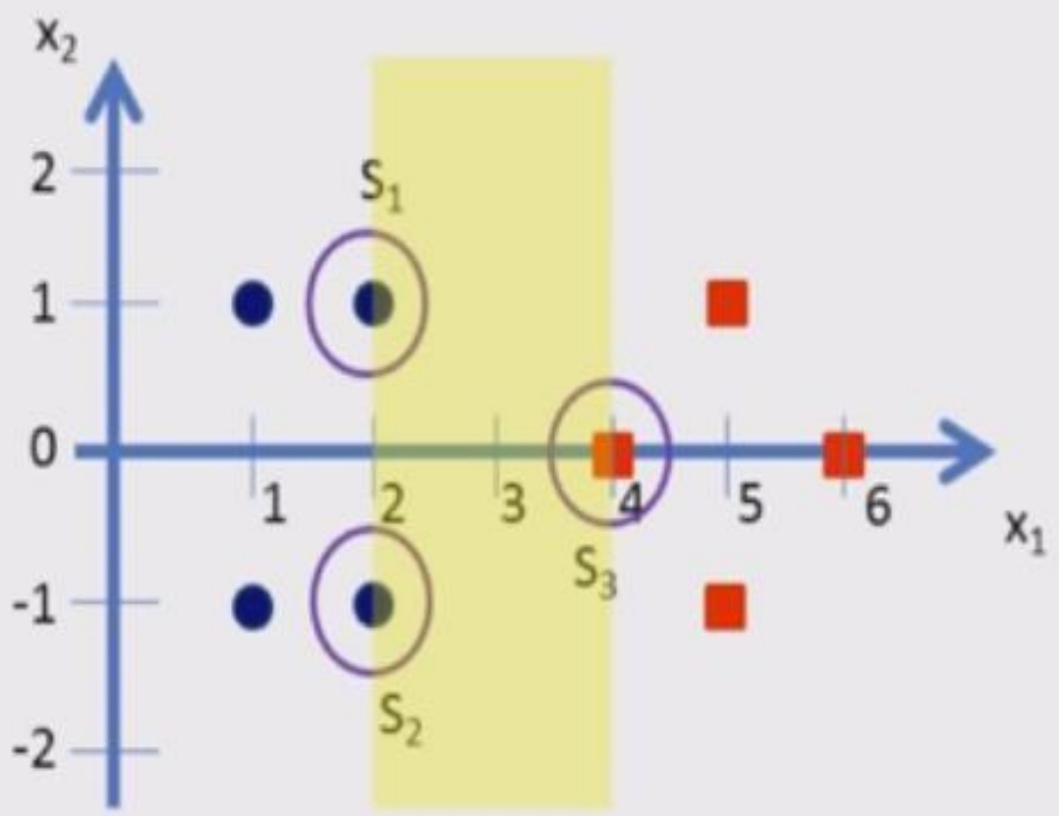
$$\frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0$$

A Geometrical Interpretation



Example

- Here we select 3 Support Vectors to start with.
- They are S_1, S_2 and S_3.

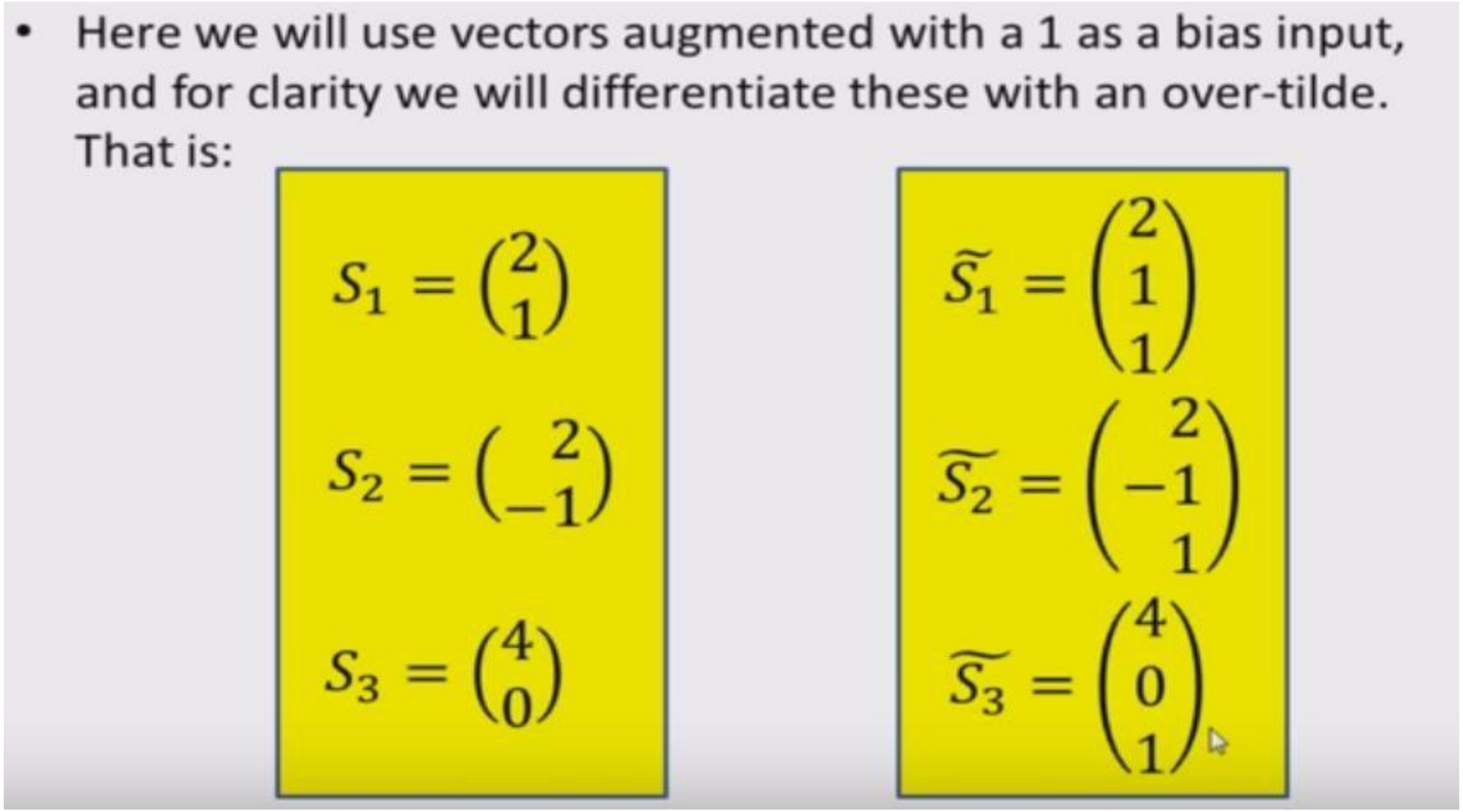


$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

Example

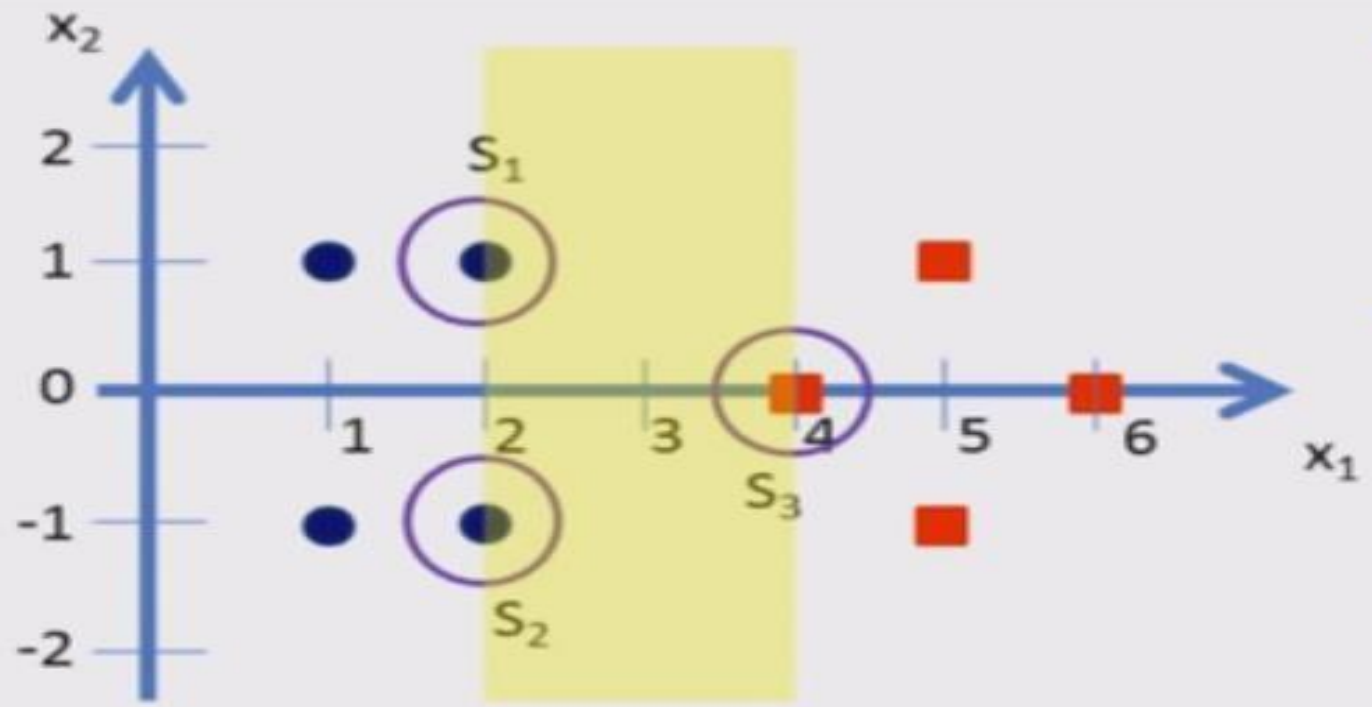


- Here we will use vectors augmented with a 1 as a bias input, and for clarity we will differentiate these with an over-tilde. That is:

$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad \widetilde{S_1} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix} \qquad \widetilde{S_2} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \qquad \widetilde{S_3} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$



- Now we need to find 3 parameters α_1, α_2, and α_3 based on the following 3 linear equations:

$$\alpha_1 \widetilde{S_1}.\widetilde{S_1} + \alpha_2 \widetilde{S_2}.\widetilde{S_1} + \alpha_3 \widetilde{S_3}.\widetilde{S_1} = -1 \ \ (-ve\ class)$$

$$\alpha_1 \widetilde{S_1}.\widetilde{S_2} + \alpha_2 \widetilde{S_2}.\widetilde{S_2} + \alpha_3 \widetilde{S_3}.\widetilde{S_2} = -1 \ (-ve\ class)$$

$$\alpha_1 \widetilde{S_1}.\widetilde{S_3} + \alpha_2 \widetilde{S_2}.\widetilde{S_3} + \alpha_3 \widetilde{S_3}.\widetilde{S_3} = +1 \ \ (+ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_1} + \alpha_2\widetilde{S_2}.\widetilde{S_1} + \alpha_3\widetilde{S_3}.\widetilde{S_1} = -1 \;\; (-ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_2} + \alpha_2\widetilde{S_2}.\widetilde{S_2} + \alpha_3\widetilde{S_3}.\widetilde{S_2} = -1 \;(-ve\ class)$$

$$\alpha_1\widetilde{S_1}.\widetilde{S_3} + \alpha_2\widetilde{S_2}.\widetilde{S_3} + \alpha_3\widetilde{S_3}.\widetilde{S_3} = +1 \;\; (+ve\ class)$$

- Let's substitute the values for $\widetilde{S}_1$, $\widetilde{S}_2$ and $\widetilde{S}_3$ in the above equations.

$$\widetilde{S_1} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \qquad \widetilde{S_2} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \qquad \widetilde{S_3} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} . \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = +1$$

$$\alpha_1 \begin{pmatrix}2\\1\\1\end{pmatrix} \cdot \begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\-1\\1\end{pmatrix} \cdot \begin{pmatrix}2\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\0\\1\end{pmatrix} \cdot \begin{pmatrix}2\\1\\1\end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix}2\\1\\1\end{pmatrix} \cdot \begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\-1\\1\end{pmatrix} \cdot \begin{pmatrix}2\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\0\\1\end{pmatrix} \cdot \begin{pmatrix}2\\-1\\1\end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix}2\\1\\1\end{pmatrix} \cdot \begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}2\\-1\\1\end{pmatrix} \cdot \begin{pmatrix}4\\0\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}4\\0\\1\end{pmatrix} \cdot \begin{pmatrix}4\\0\\1\end{pmatrix} = +1$$

- After simplification we get:

$$6\alpha_1 + 4\alpha_2 + 9\alpha_3 = -1$$

$$4\alpha_1 + 6\alpha_2 + 9\alpha_3 = -1$$

$$9\alpha_1 + 9\alpha_2 + 17\alpha_3 = +1$$

- Simplifying the above 3 simultaneous equations we get: $\alpha_1 = \alpha_2 = -3.25$ and $\alpha_3 = 3.5$.

$\alpha_1 = \alpha_2 = -3.25$ and $\alpha_3 = 3.5$

$$\widetilde{S_1} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad \widetilde{S_2} = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \quad \widetilde{S_3} = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

- The hyper plane that discriminates the positive class from the negative class is give by:

$$\widetilde{w} = \sum_i \alpha_i \widetilde{S_i}$$

- Substituting the values we get:

$$\widetilde{w} = \alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

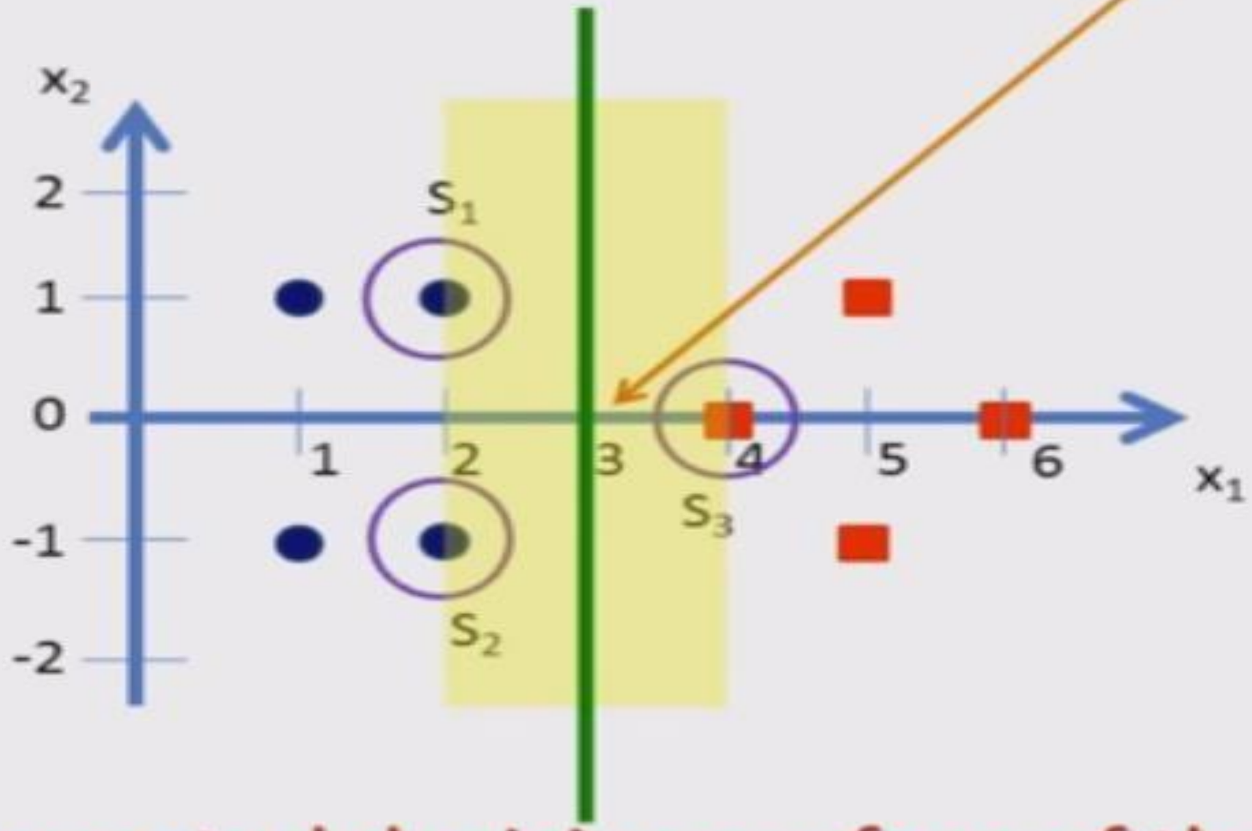
$$\widetilde{w} = (-3.25).\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25).\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5).\begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

$$\tilde{w} = (-3.25).\begin{pmatrix}2\\1\\1\end{pmatrix} + (-3.25).\begin{pmatrix}2\\-1\\1\end{pmatrix} + (3.5).\begin{pmatrix}4\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\-3\end{pmatrix}$$

- Our vectors are augmented with a bias.
- Hence we can equate the entry in $\tilde{w}$ as the hyper plane with an offset b.
- Therefore the separating hyper plane equation $y = wx + b$ with $w = \begin{pmatrix}1\\0\end{pmatrix}$ and offset $b = -3$.

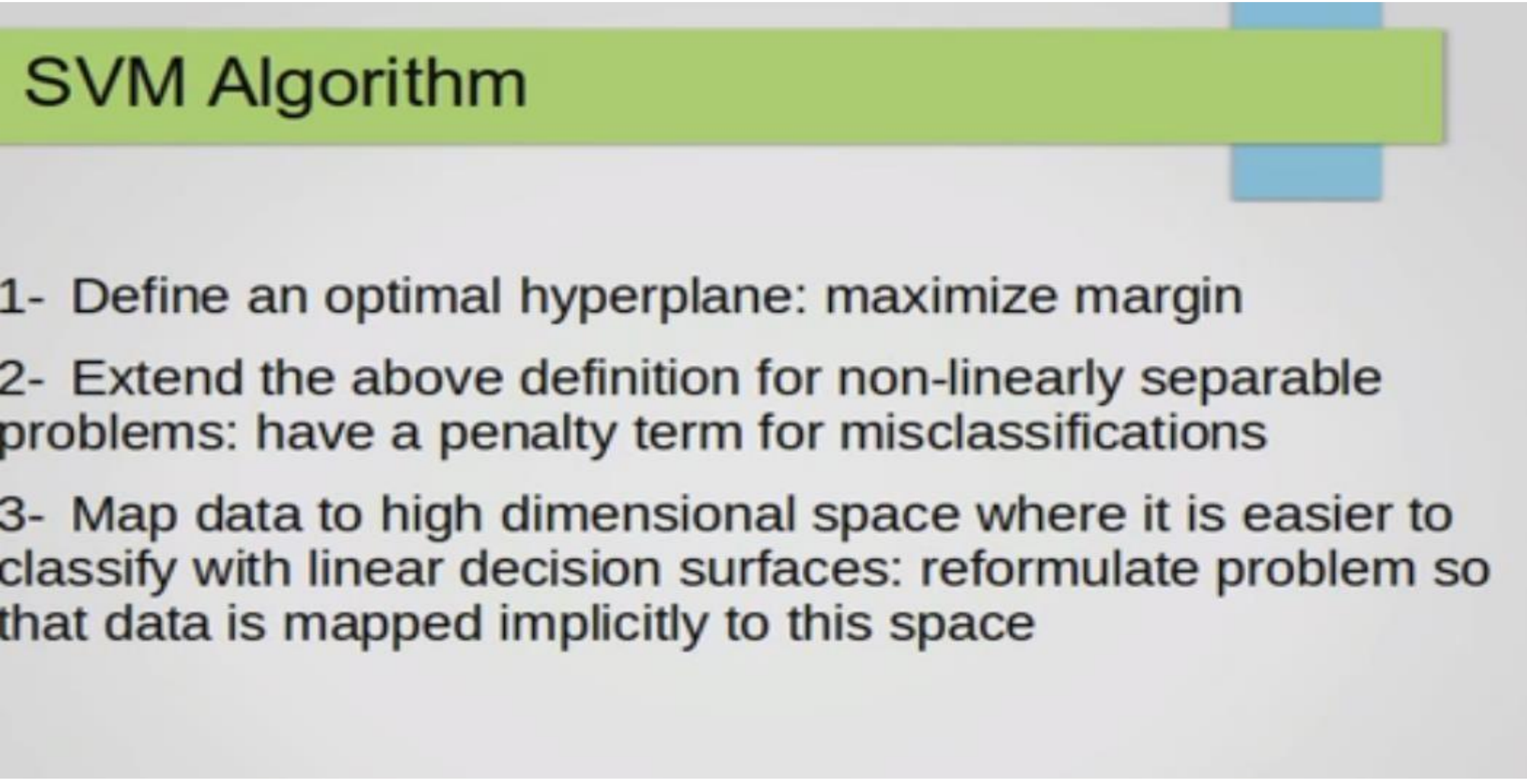
Support Vector Machines

- $y = wx + b$ with $w = \binom{1}{0}$ and offset $b = -3$.



- This is the expected decision surface of the LSVM.

Kernel trick



SVM Algorithm

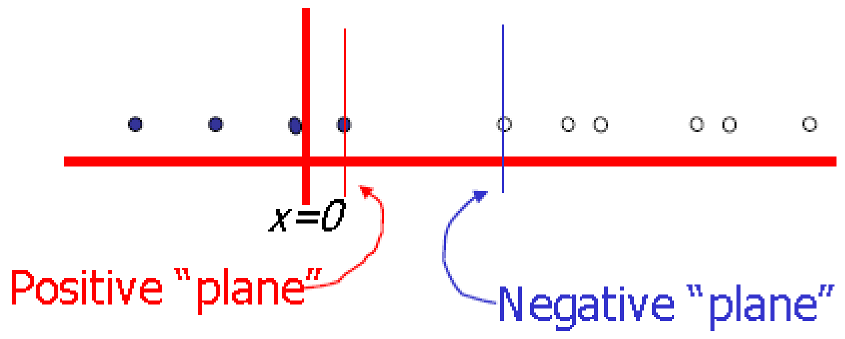
1- Define an optimal hyperplane: maximize margin

2- Extend the above definition for non-linearly separable problems: have a penalty term for misclassifications

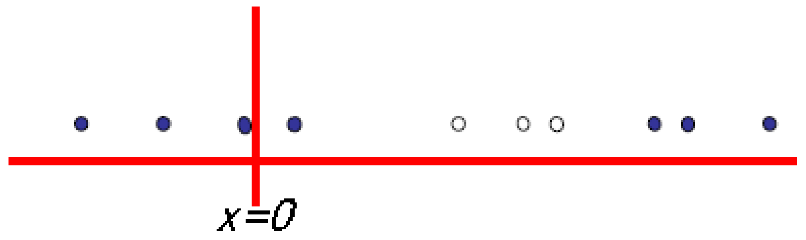
3- Map data to high dimensional space where it is easier to classify with linear decision surfaces: reformulate problem so that data is mapped implicitly to this space

Suppose we're in 1-dimension

Not a big surprise



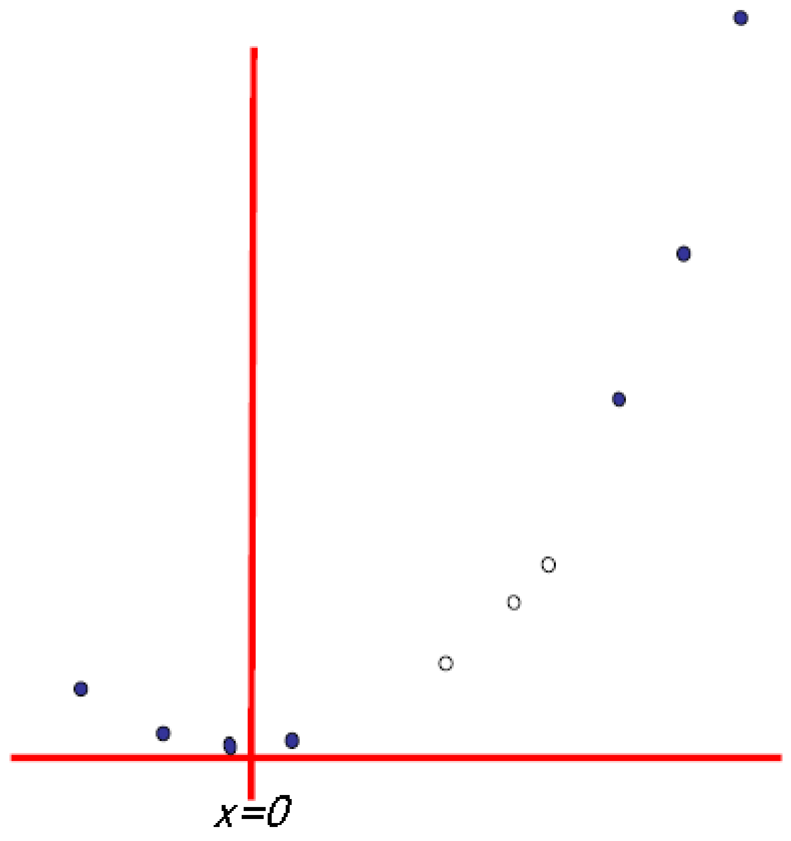
Harder 1-dimensional dataset



That's wiped the smirk off SVM's face.

What can be done about this?

Harder 1-dimensional dataset

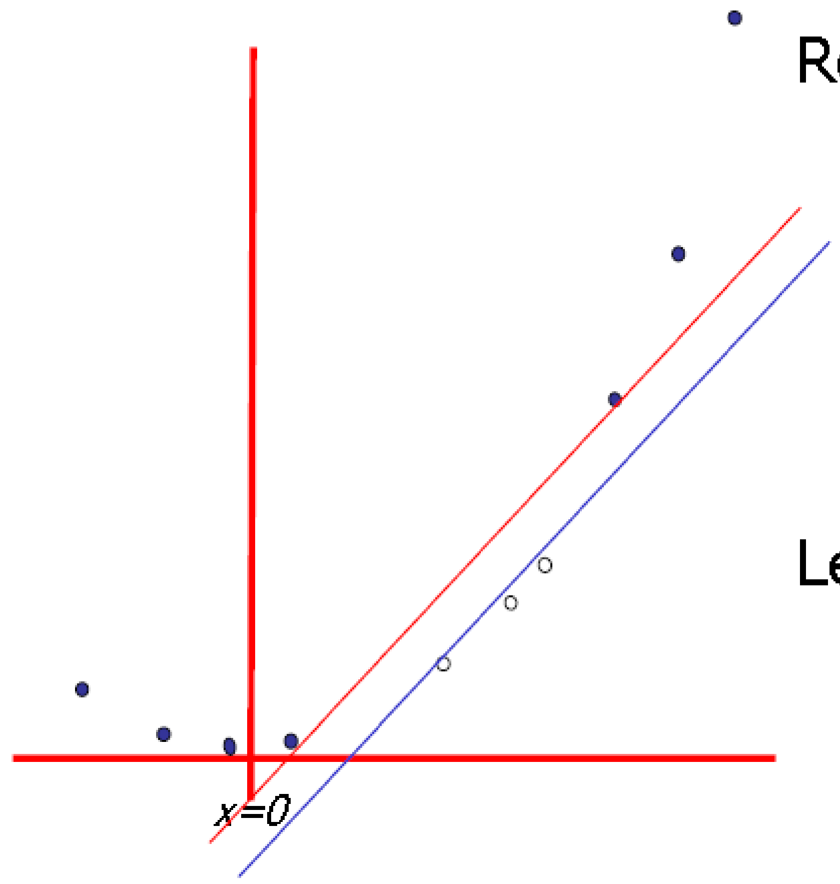


Remember how permitting non-linear basis functions made linear regression so much nicer?

Let's permit them here too

$$\mathbf{z}_k = (x_k, x_k^2)$$

Harder 1-dimensional dataset



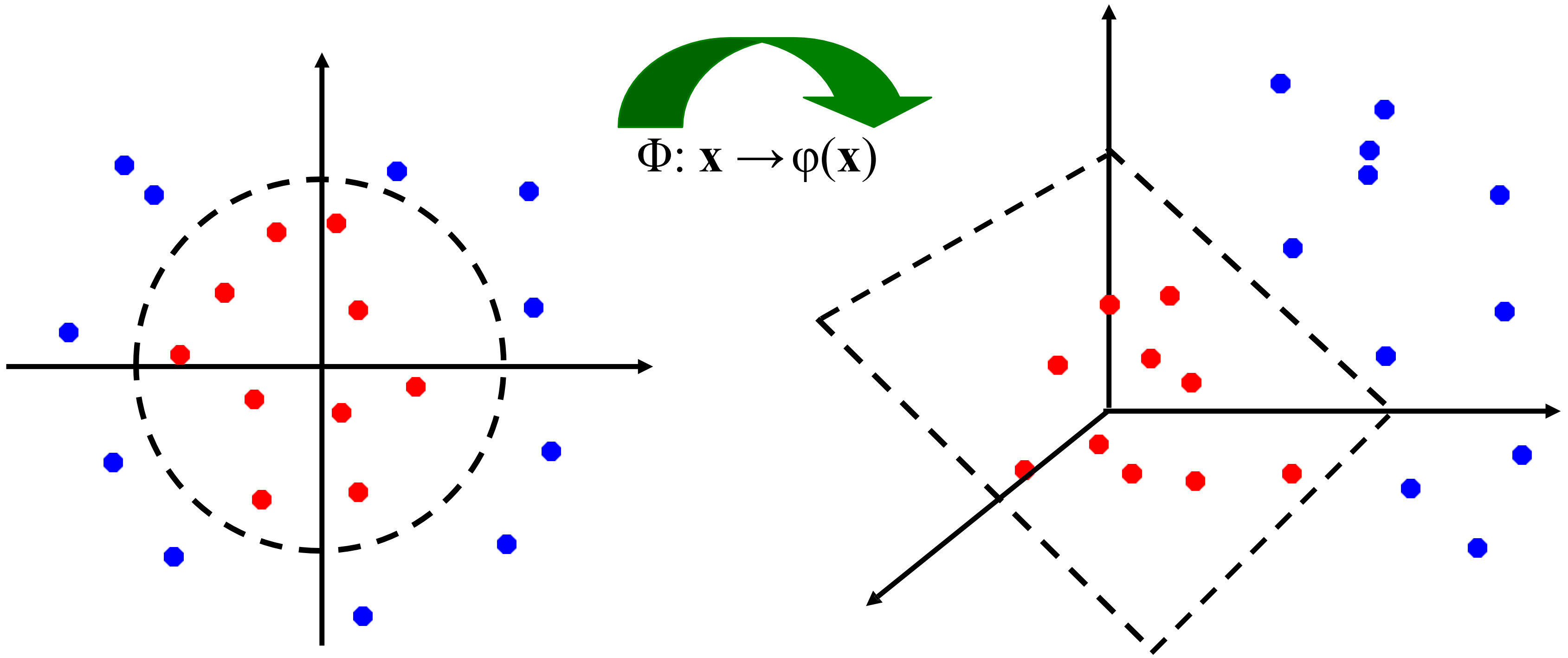
Remember how permitting non-linear basis functions made linear regression so much nicer?

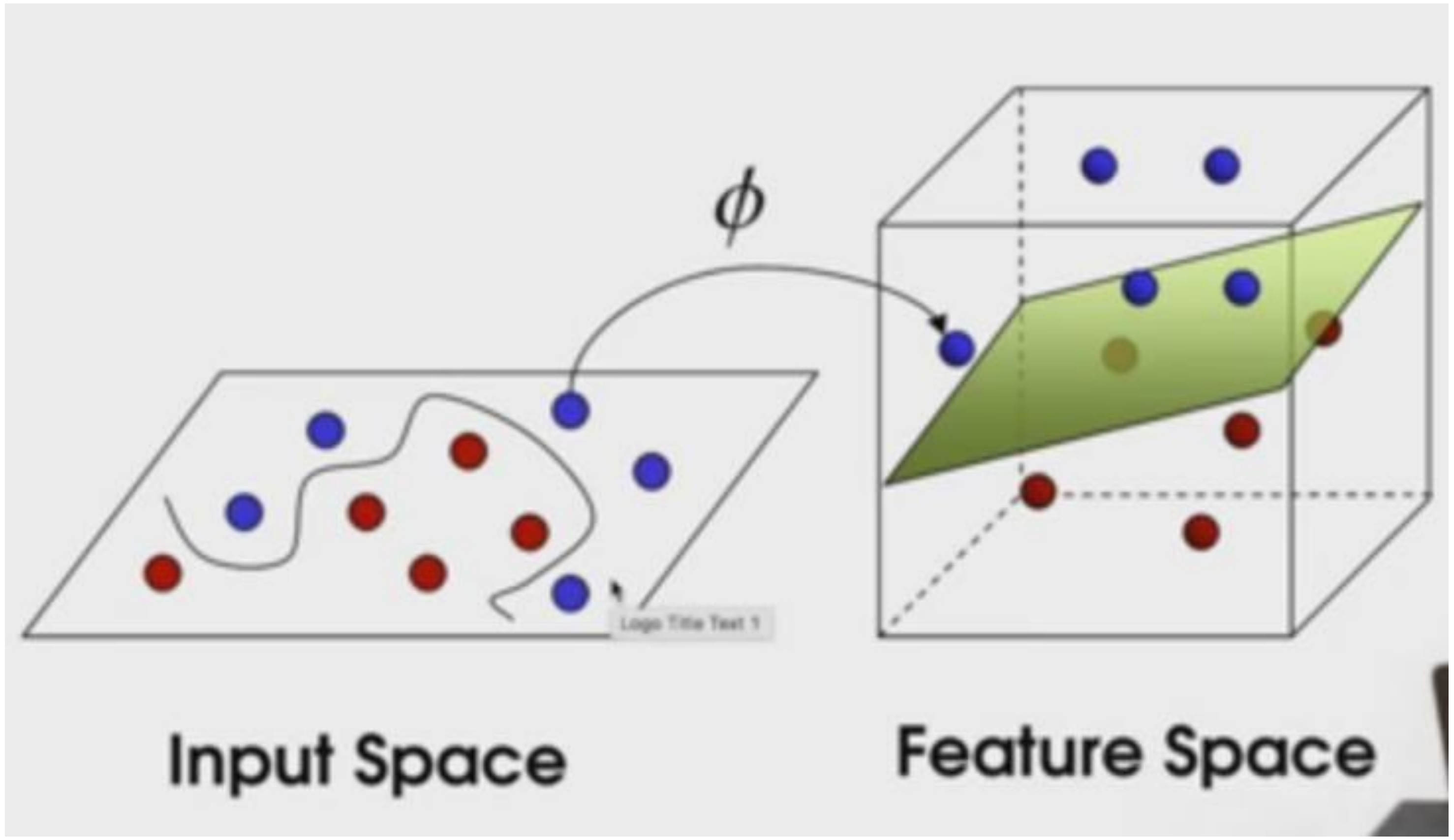
Let's permit them here too

$$\mathbf{z}_k = (x_k, x_k^2)$$

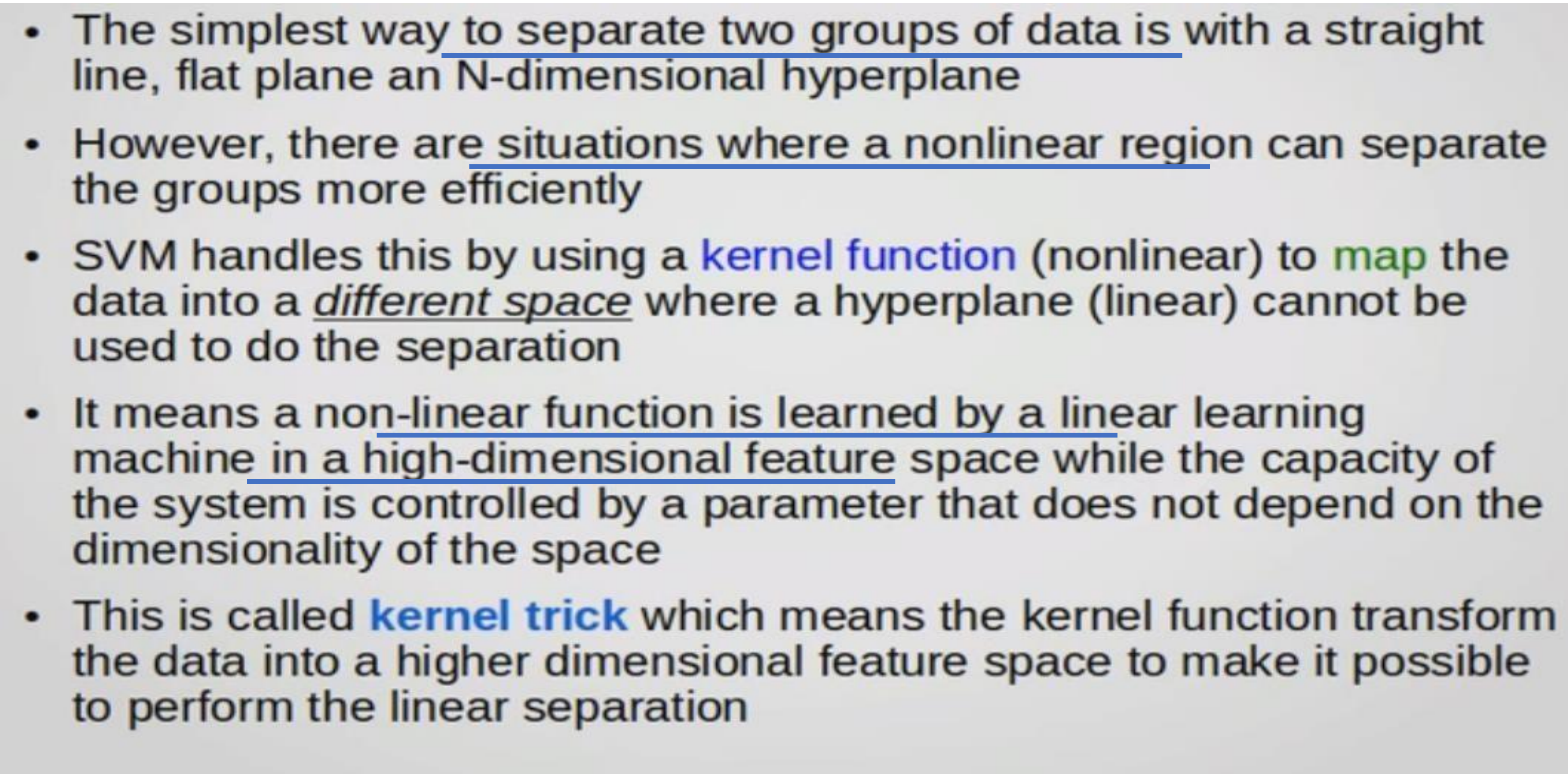
Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:




ϕ
Logo Title Text 1
Input Space
Feature Space

svm for nonlinear reparability



- The simplest way to separate two groups of data is with a straight line, flat plane an N-dimensional hyperplane
- However, there are situations where a nonlinear region can separate the groups more efficiently
- SVM handles this by using a kernel function (nonlinear) to map the data into a *different space* where a hyperplane (linear) cannot be used to do the separation
- It means a non-linear function is learned by a linear learning machine in a high-dimensional feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space
- This is called **kernel trick** which means the kernel function transform the data into a higher dimensional feature space to make it possible to perform the linear separation

Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial $\mathbf{K(x,z) = (1+x^Tz)^d}$
    - Gives feature conjunctions
  - Radial basis function (infinite dimensional space)

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2}$$

- Haven't been very useful in text classification

Next:

Ensemble learning

جامعة مصر للمعلوماتية
EGYPT UNIVERSITY OF INFORMATICS

FACULTY OF ENGINEERING

Thank You

جامعة مصر للمعلوماتية
EGYPT UNIVERSITY
OF INFORMATICS

FACULTY OF
ENGINEERING