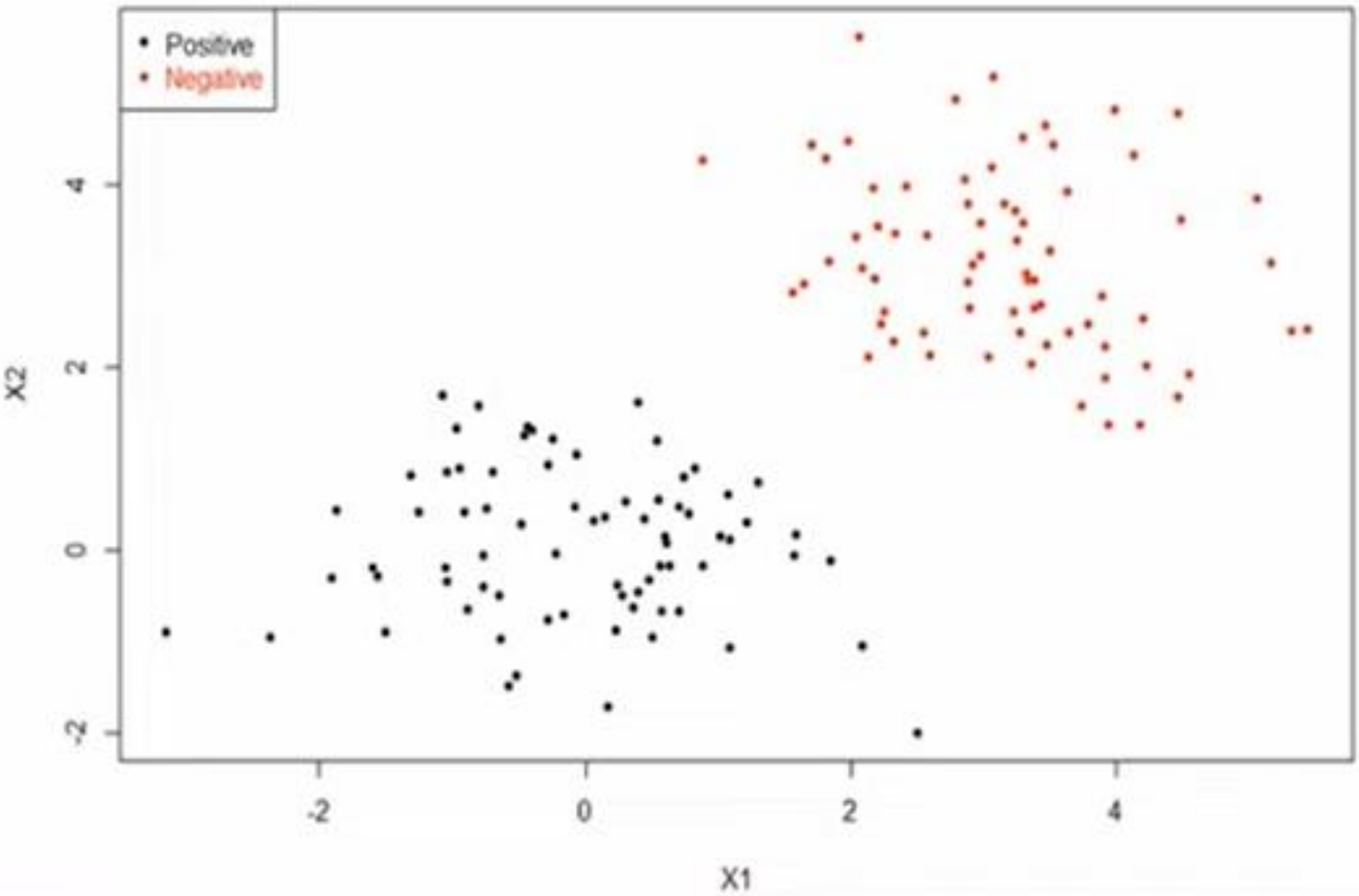
Support Vector Machine (SVM Classifier)

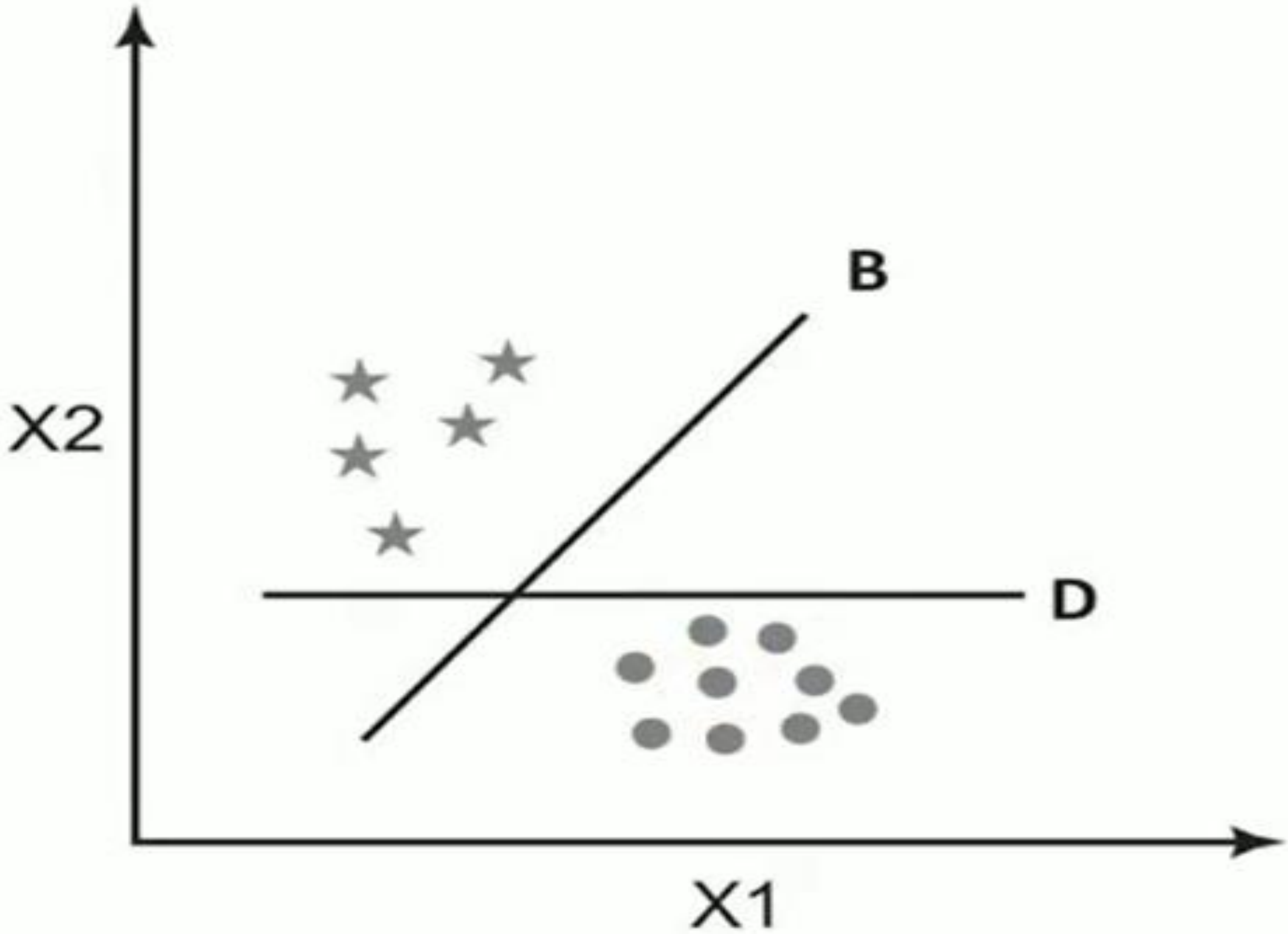
BCSE0105 MACHINE LEARNING

Support Vector Machine is a discriminative classifier that is formally designed by a separative hyperplane. It is a representation of examples as points in space that are mapped so that the points of different categories are separated by a gap as wide as possible.


Positive
Negative
X2
4
2
0
-2
-2
0
2
4
X1

SVM

- "Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges.
- However, it is mostly used in classification problems.
- In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.
- Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot).

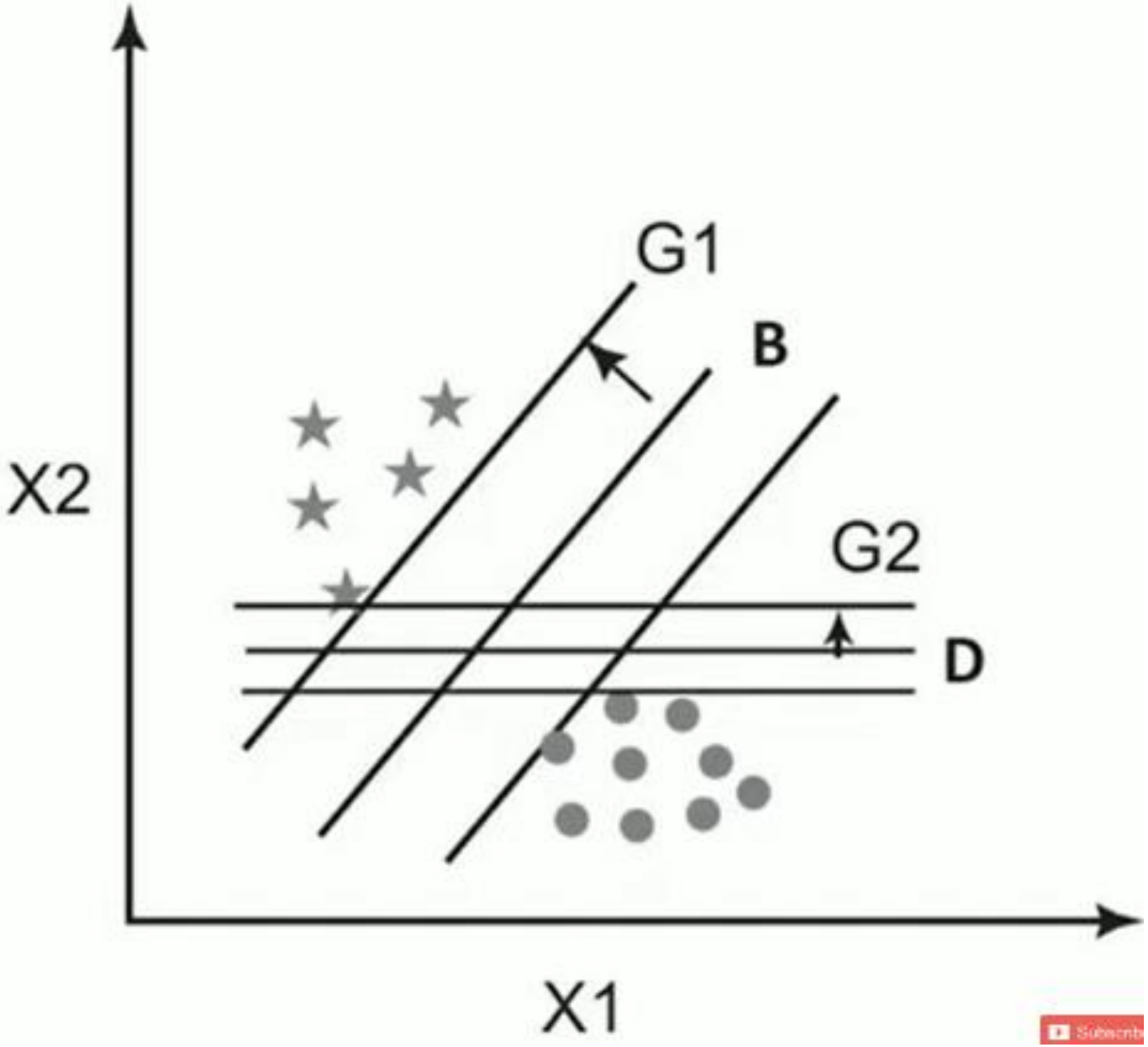


X2
B
D
X1

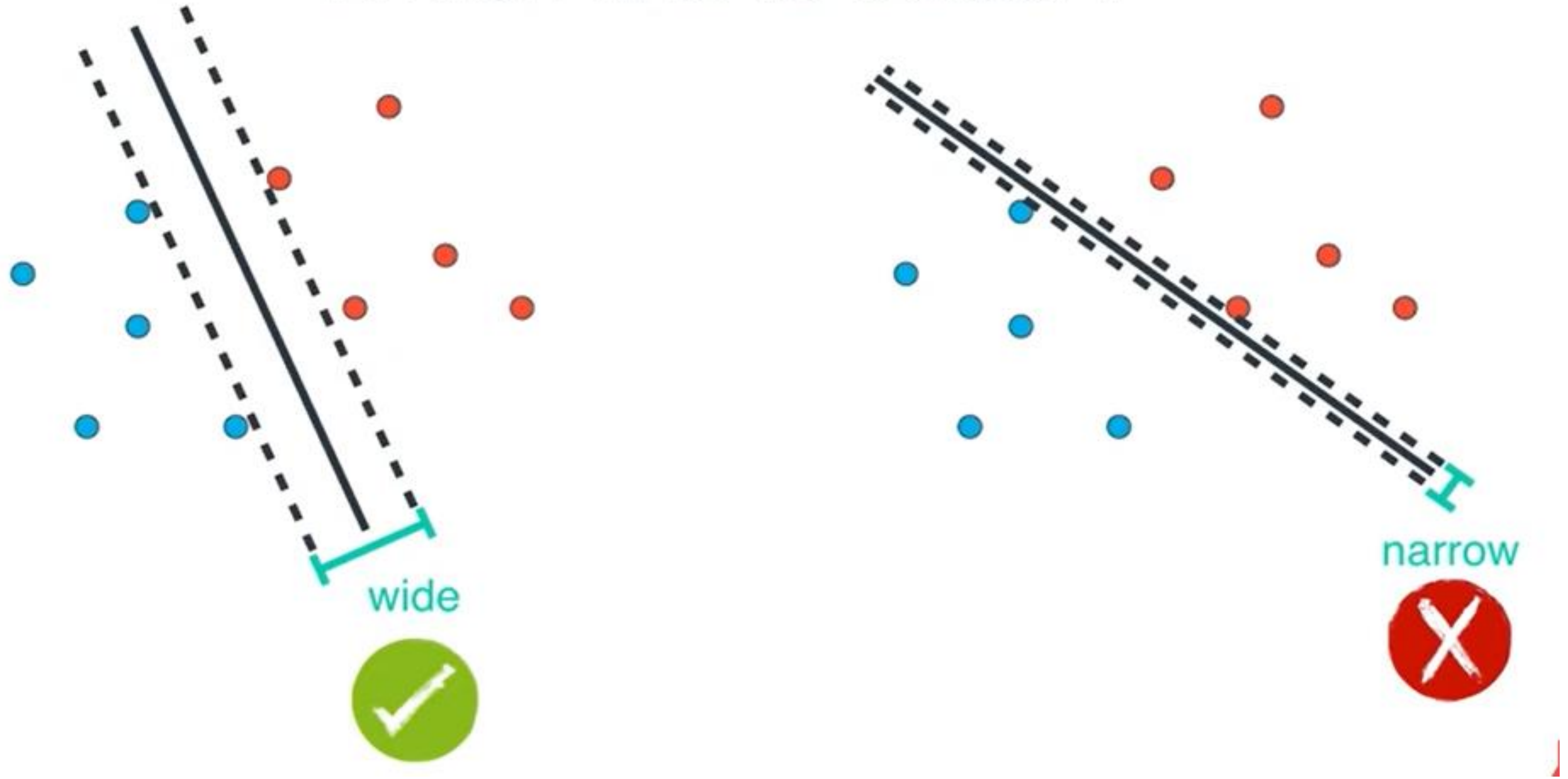
To identify the right hyperplane

Thumb rule to identify the right hyper-plane

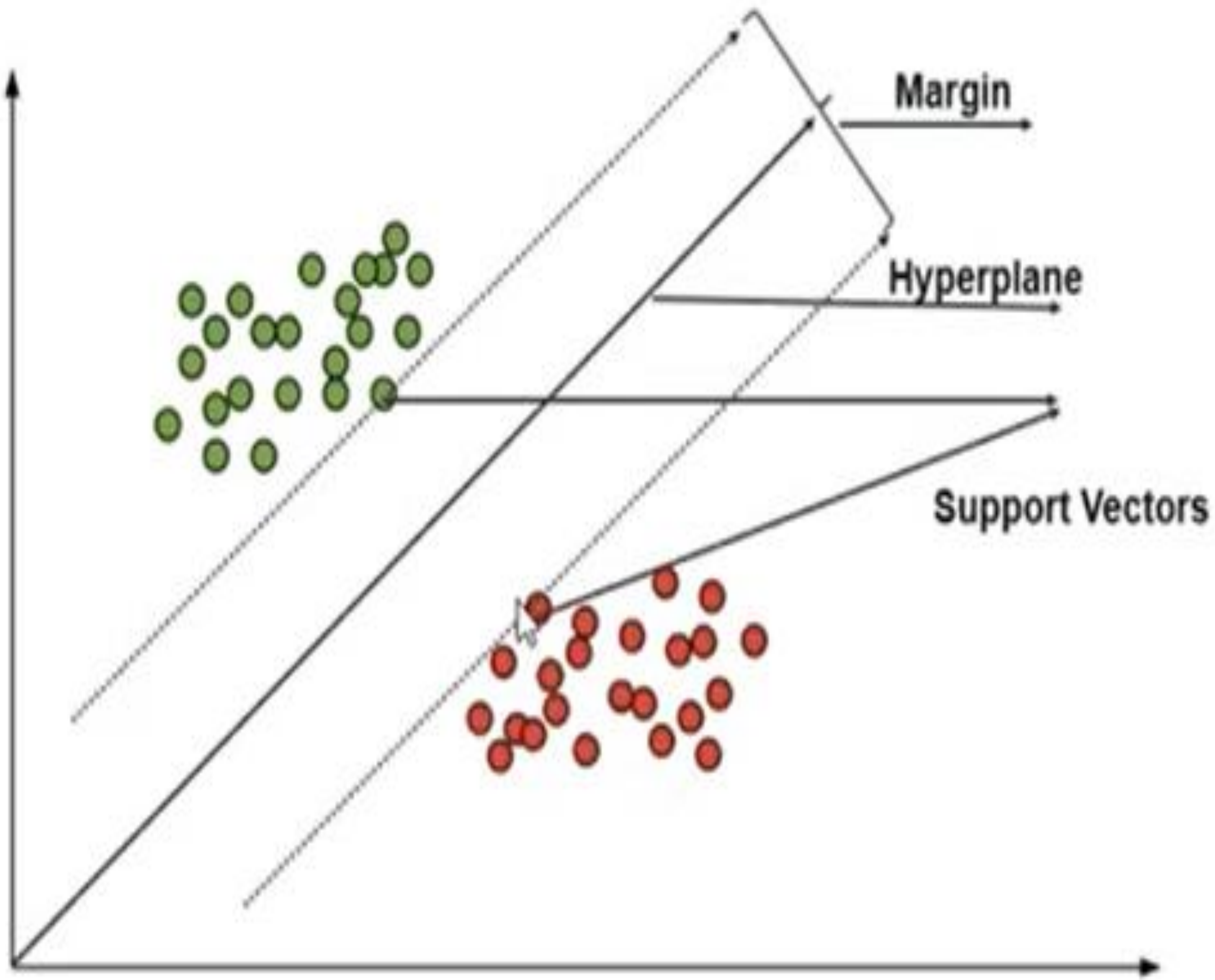
- Select the hyper-plane which segregates the two classes better.
- Maximizing the distances between nearest data point (either class) and hyper-plane. This distance is called as **Margin**.


G1
B
X2
G2
D
X1

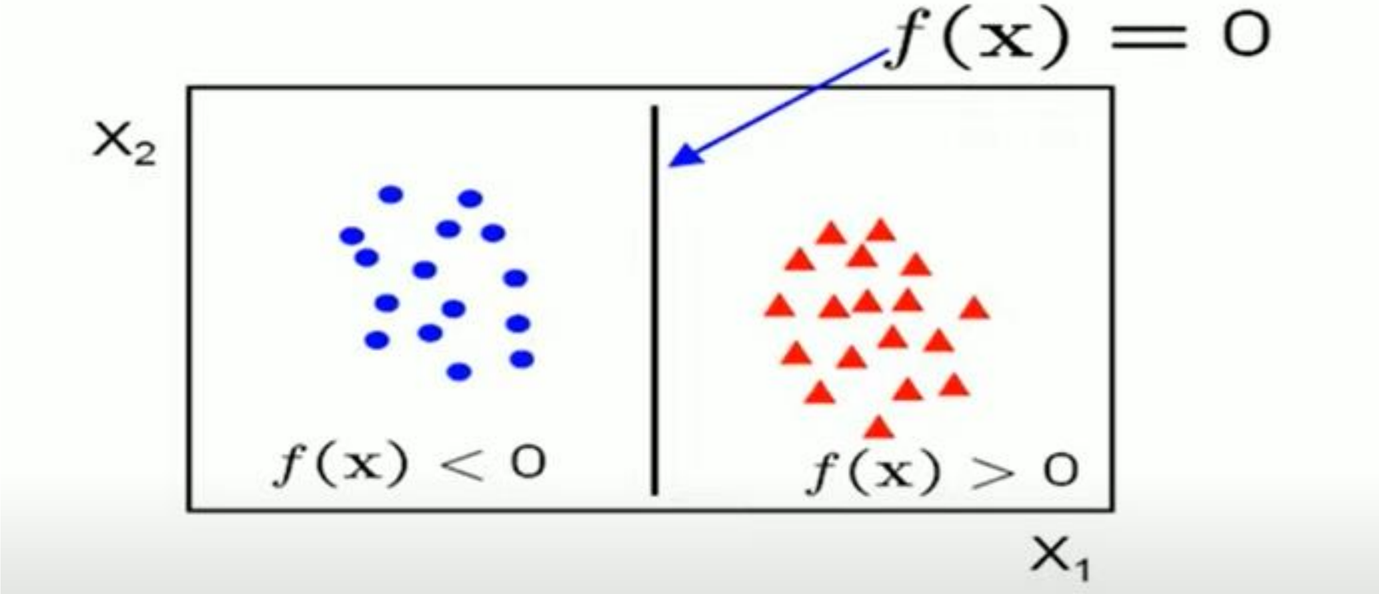
Which line is better?



Linearly Separable classes (use Linear SVM)

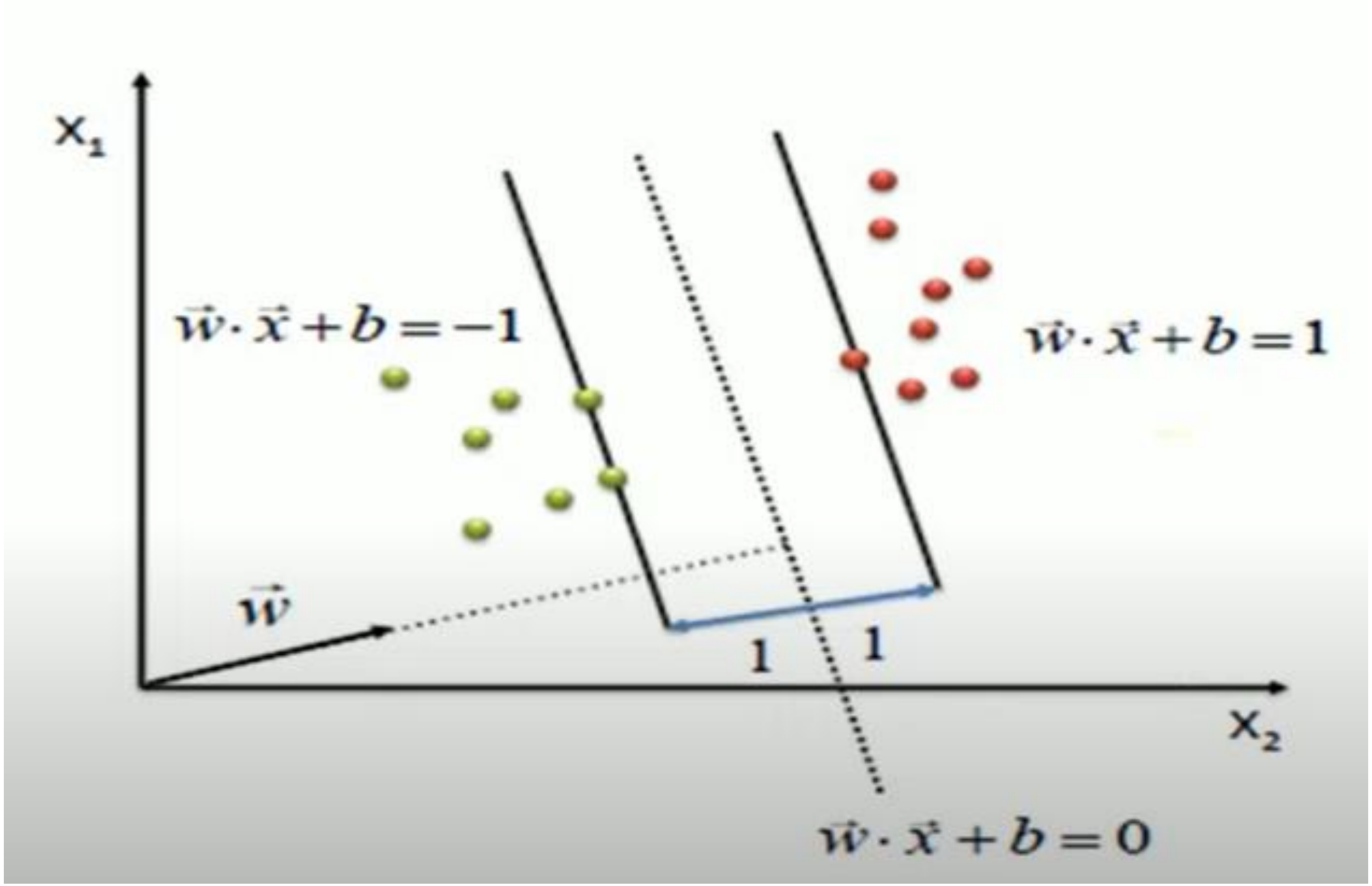


- The widest 'road' that can be laid between the two classes
- This width is bounded by the presence of observations (aka vectors) on either sides
- As the Maximum Margin Hyperplane is **support**ed by these vectors, they are called *support vectors*



- $f(x) = W.X + b$
- W is the normal to the line, X is input vector and b the bias
- W is known as the weight vector

SVM Model


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

$$\boxed{\max \frac{2}{\|w\|}}$$

$s.t.$

$(w \cdot x + b) \geq 1, \forall x$ of class 1

$(w \cdot x + b) \leq -1, \forall x$ of class 2

Advantages

- The main strength of SVM is that they work well even when the number of SVM features is much larger than the number of instances.
- It can work on datasets with huge feature space, such is the case in spam filtering, where a large number of words are the potential signifiers of a message being spam.
- Even when the optimal decision boundary is a nonlinear curve, the SVM transforms the variables to create new dimensions such that the representation of the classifier is a linear function of those transformed dimensions of the data.
- SVMs are conceptually easy to understand. They create an easy-to-understand linear classifier.
- SVMs are now available with almost all data analytics toolsets.

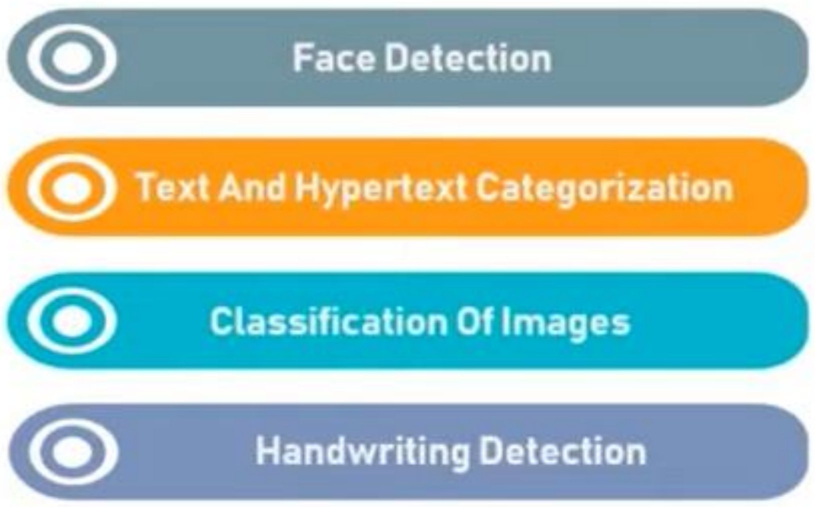
Disadvantages

- The SVM technique has two major constraints
  - It works well only with real numbers, i.e., all the data points in all the dimensions must be defined by numeric values only,
  - It works only with binary classification problems. One can make a series of cascaded SVMs to get around this constraint.
- Training the SVMs is an inefficient and time consuming process, when the data is large.
- It does not work well when there is much noise in the data, and thus has to compute soft margins.
- The SVMs will also not provide a probability estimate of classification, i.e., the confidence level for classifying an instance.

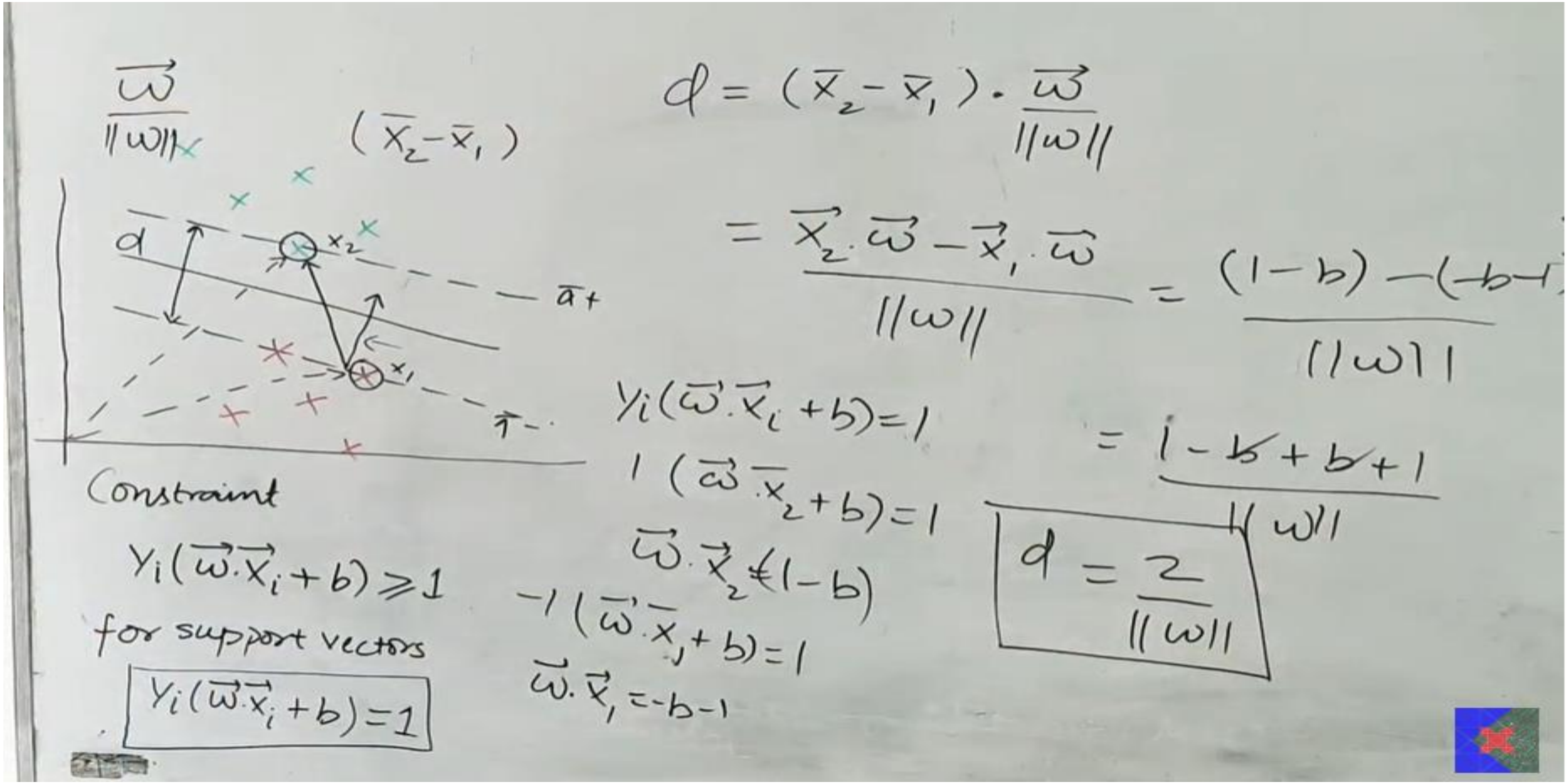
Applications of SVMs

1. Classification
2. Regression analysis
3. Pattern recognition
4. Outliers detection.
5. Relevance based applications

Applications



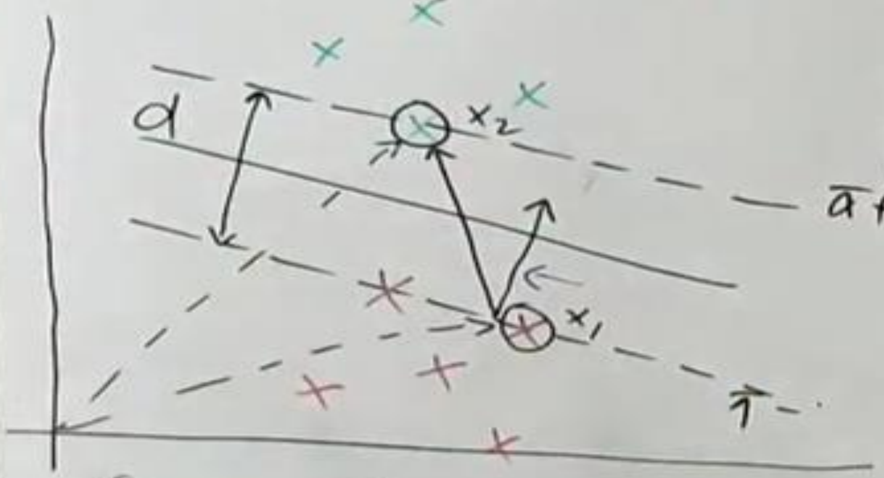
Geometric explanation



Optimization function /cost function

$w^T x + b = 10$

$\frac{\vec{w}}{\|w\|}$ $(\bar{x}_2 - \bar{x}_1)$



Constraint

$y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1$

for support vectors

$\boxed{y_i(\vec{w}\cdot\vec{x}_i + b) = 1}$

$y_i(\vec{w}\cdot\vec{x}_i + b) = 1$

$1(\vec{w}\cdot\bar{x}_2 + b) = 1$

$\vec{w}\cdot\vec{x}_2 = (1 - b)$

$-1(\vec{w}\cdot\bar{x}_1 + b) = 1$

$\vec{w}\cdot\vec{x}_1 = -b - 1$

$$\boxed{\underset{(w^*, b^*)}{\operatorname{argmax}} \frac{2}{\|w\|} \text{ such that } y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1}$$

$$= \frac{1 - \cancel{b} + \cancel{b} + 1}{\|w\|}$$

$$\boxed{d = \frac{2}{\|w\|}}$$

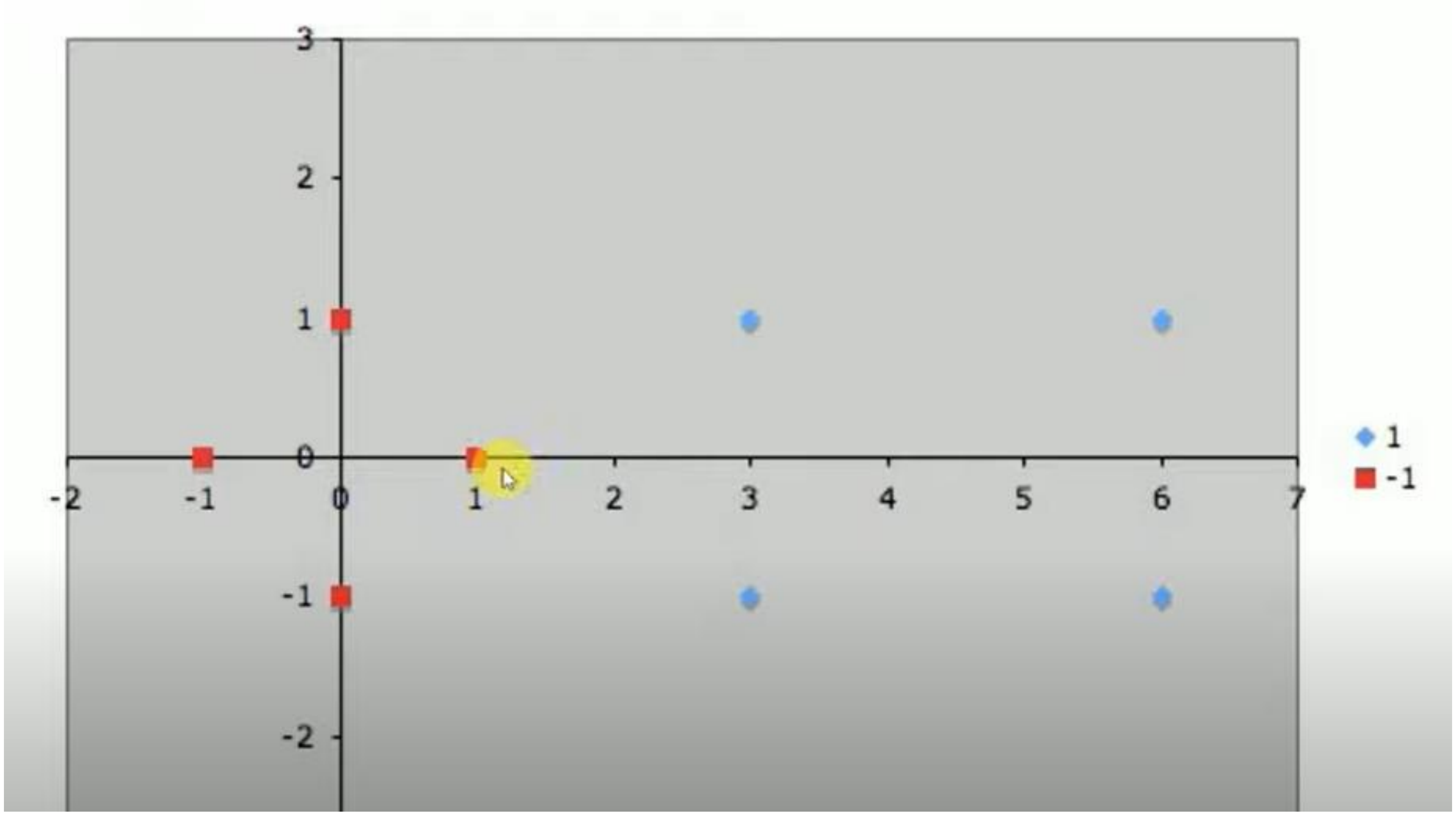
Support Vector Machine - Linear Example Solved

Suppose we are given the following positively labeled data points,

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

and the following negatively labeled data points,

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$

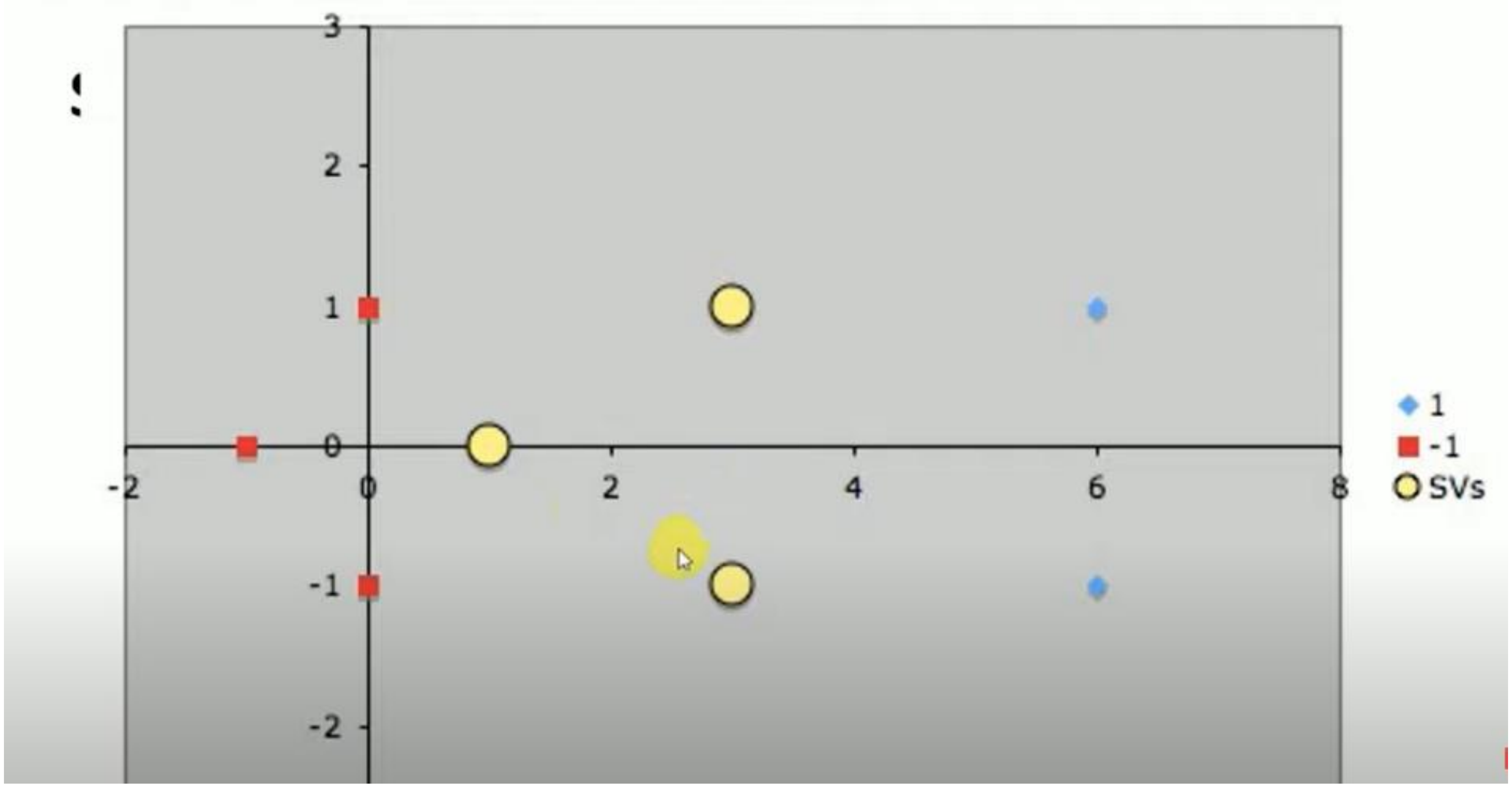


3
2
1
0
-1
-2
-2
-1
0
1
2
3
4
5
6
7
1
-1

Support Vector Machine - Linear Example Solved

- By inspection, it should be obvious that there are **three** support vectors,

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$



3
2
1
0
-1
-2
-2
0
2
4
6
8
1
-1
SVs

Support Vector Machine - Linear Example Solved

- Each vector is augmented with a 1 as a bias input

- So, $s_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then $\widetilde{s_1} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$

- Similarly,

- $s_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$, then $\widetilde{s_2} = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}$ and $s_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix}$, then $\widetilde{s_3} = \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$

Support Vector Machine - Linear Example Solved

$$
\begin{aligned}
\alpha_1 \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_1 &= -1 \\
\alpha_1 \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_2 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_2 &= +1 \\
\alpha_1 \tilde{s}_1 \cdot \tilde{s}_3 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_3 + \alpha_3 \tilde{s}_3 \cdot \tilde{s}_3 &= +1
\end{aligned}
$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} = 1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix} = 1$$

$$\alpha_1(1+0+1)+\alpha_2(3+0+1)+\alpha_3(3+0+1)=-1$$

$$\alpha_1(3+0+1)+\alpha_2(9+1+1)+\alpha_3(9-1+1)=1$$

$$\alpha_1(3+0+1)+\alpha_2(9-1+1)+\alpha_3(9+1+1)=1$$

$$2\alpha_1+4\alpha_2+4\alpha_3=-1$$

$$4\alpha_1+11\alpha_2+9\alpha_3=1$$

$$4\alpha_1+9\alpha_2+11\alpha_3=1$$

$$\alpha_1=-3.5$$

$$\alpha_2=0.75$$

$$\alpha_3=0.75$$

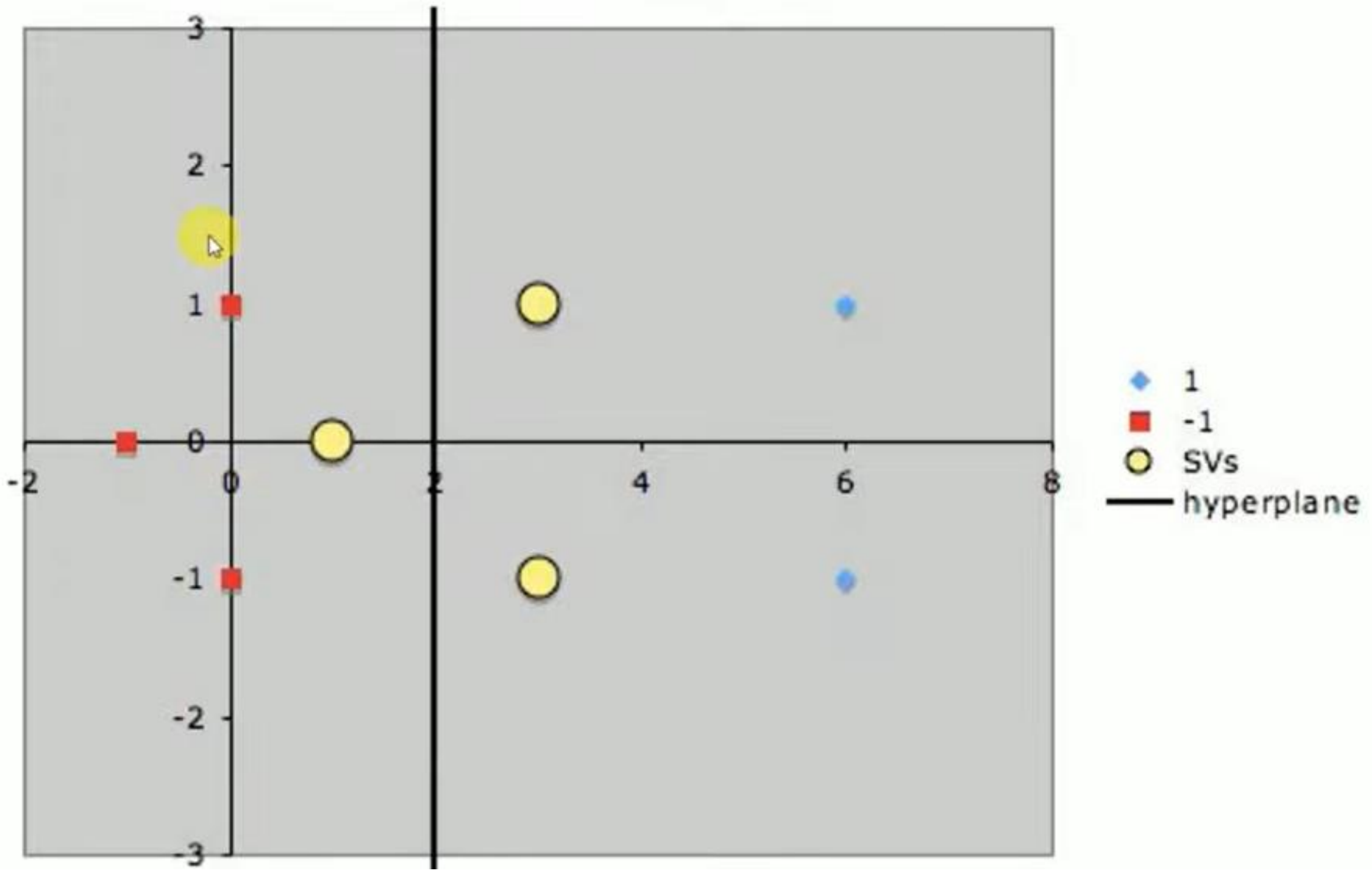
Support Vector Machine - Linear Example Solved

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i$$

$$= -3.5 \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix} + 0.75 \begin{pmatrix} 3 \\ -1 \\ 1 \end{pmatrix}$$

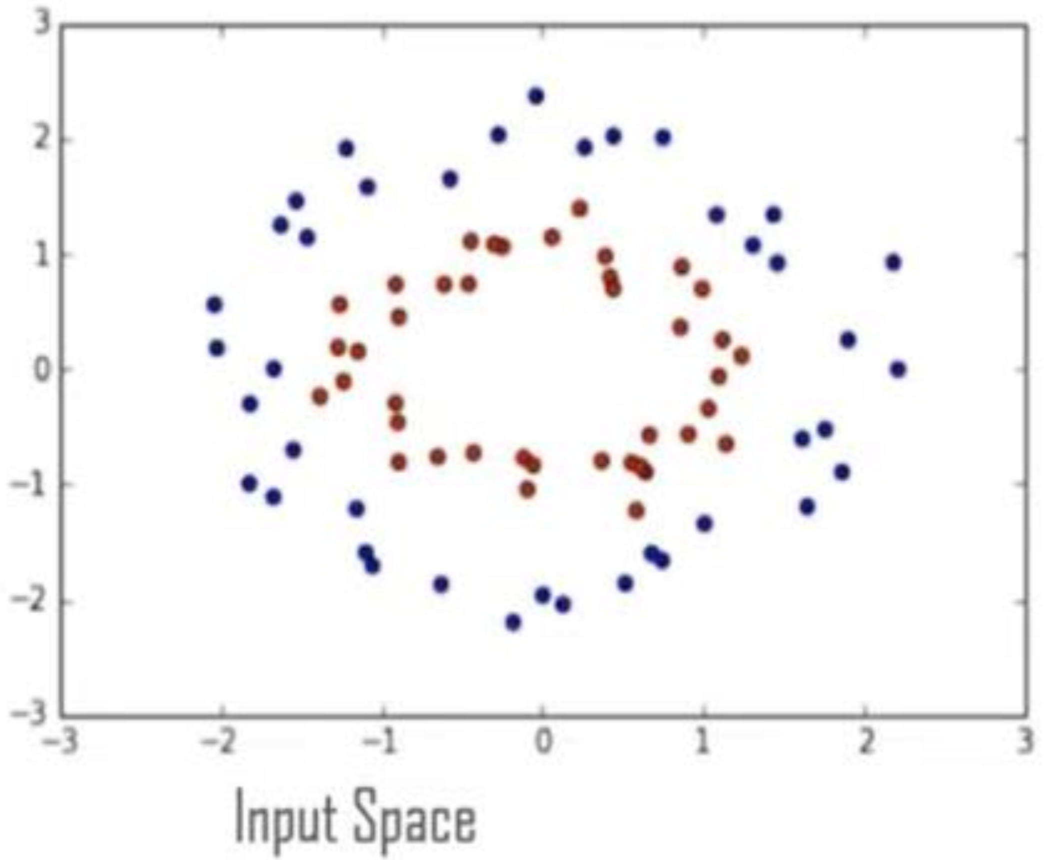
$$= \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix}$$

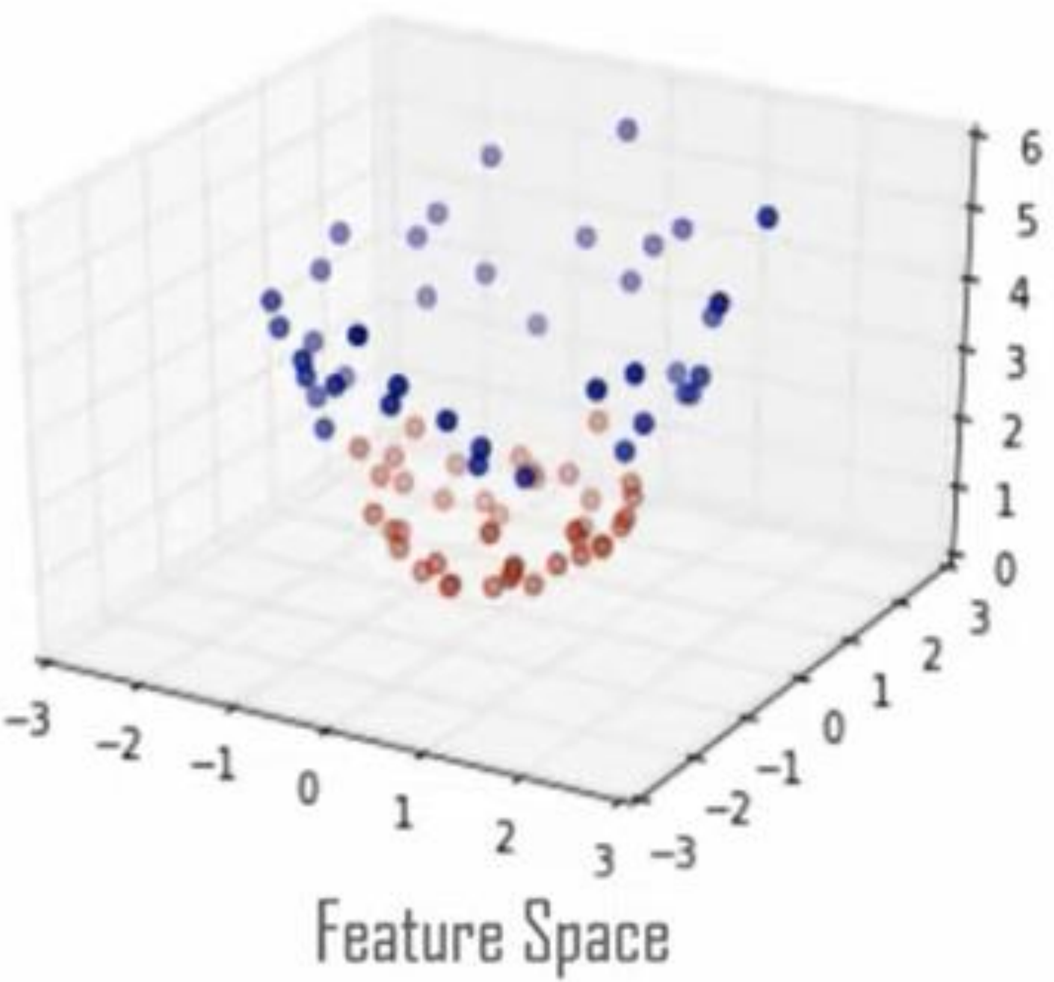
- Finally, remembering that our vectors are augmented with a bias.
- We can equate the last entry in $\tilde{w}$ as the hyperplane offset b and write the separating
- Hyperplane equation $\boldsymbol{y} = \boldsymbol{wx} + \boldsymbol{b}$
- with $\boldsymbol{w} = \binom{1}{0}$ and $\boldsymbol{b} = -2$.

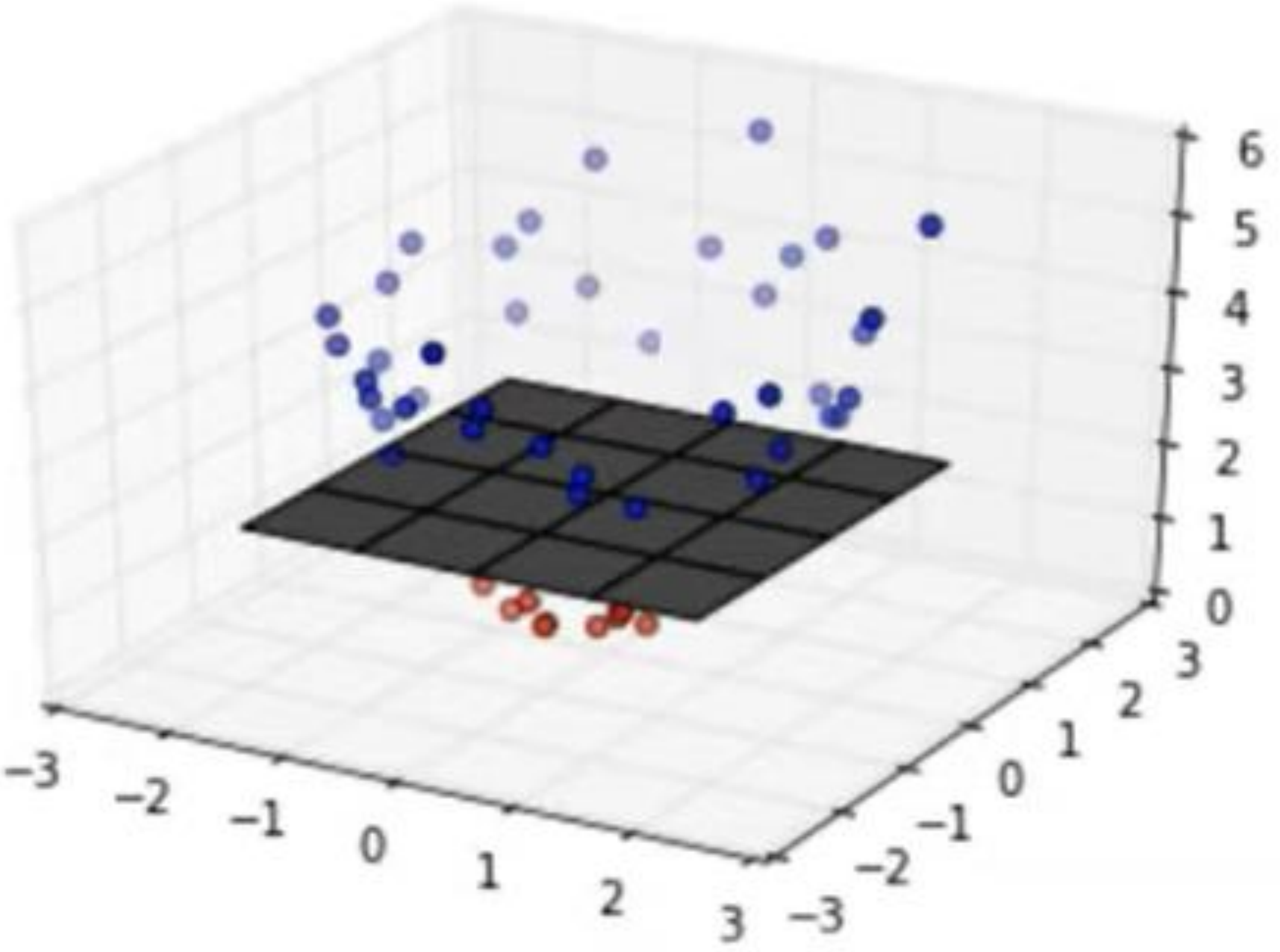


3
2
1
0
-1
-2
-3
-2
0
2
4
6
8
1
-1
SVs
hyperplane

Non-linearly Separable Classes (use Non-Linear SVM)



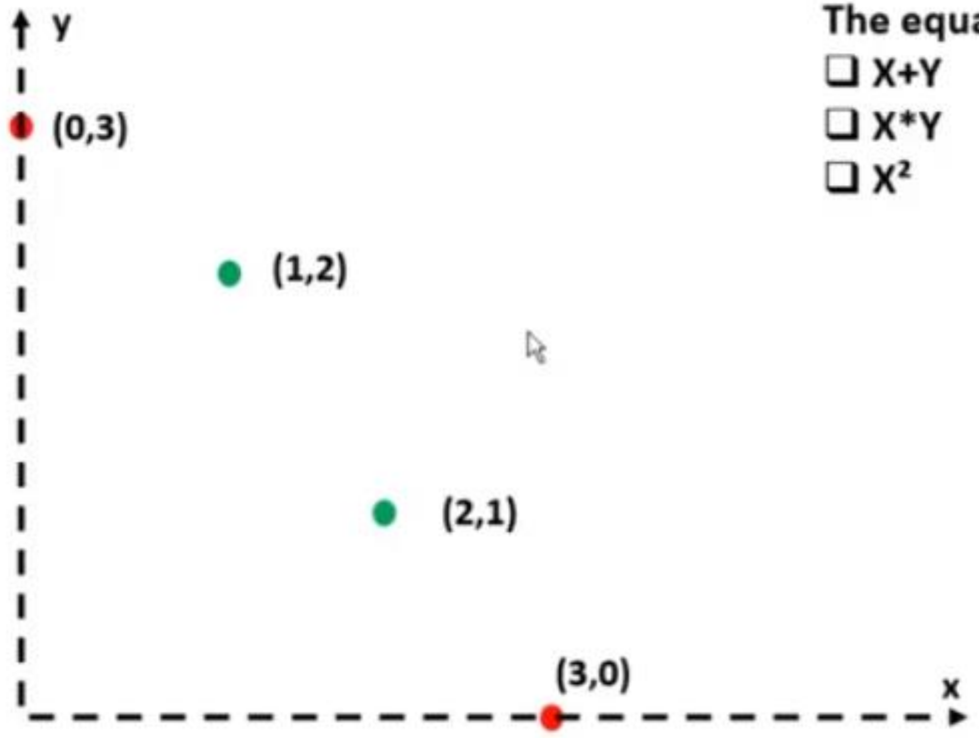




6
5
4
3
2
1
0
−3
−2
−1
0
1
2
3
3
2
1
0
−1
−2
−3

In order to make the mathematics possible, **Support Vector Machines** use something called **Kernel Functions** to *systematically* find **Support Vector Classifiers** in higher dimensions.

The kernel trick

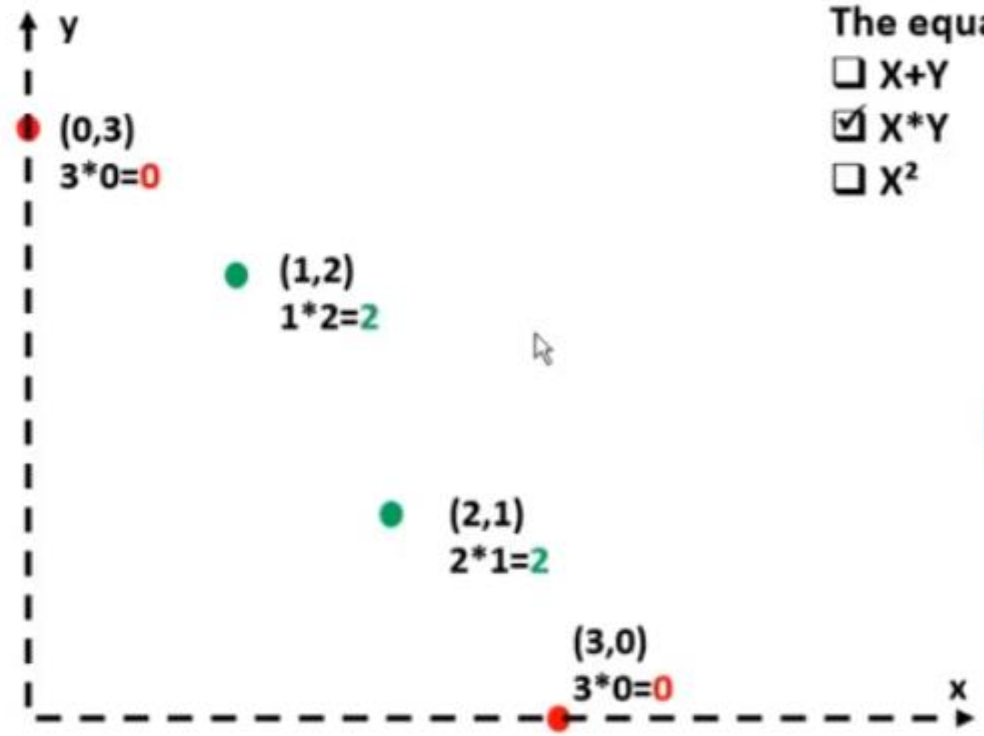


The equation that best represents the graph is:

- ❑ X+Y
- ❑ X*Y
- ❑ X^2

	(0,3)	(1,2)	(2,1)	(3,0)
X+Y	3	3	3	3
X*Y	0	2	2	0
X^2	0	1	4	9

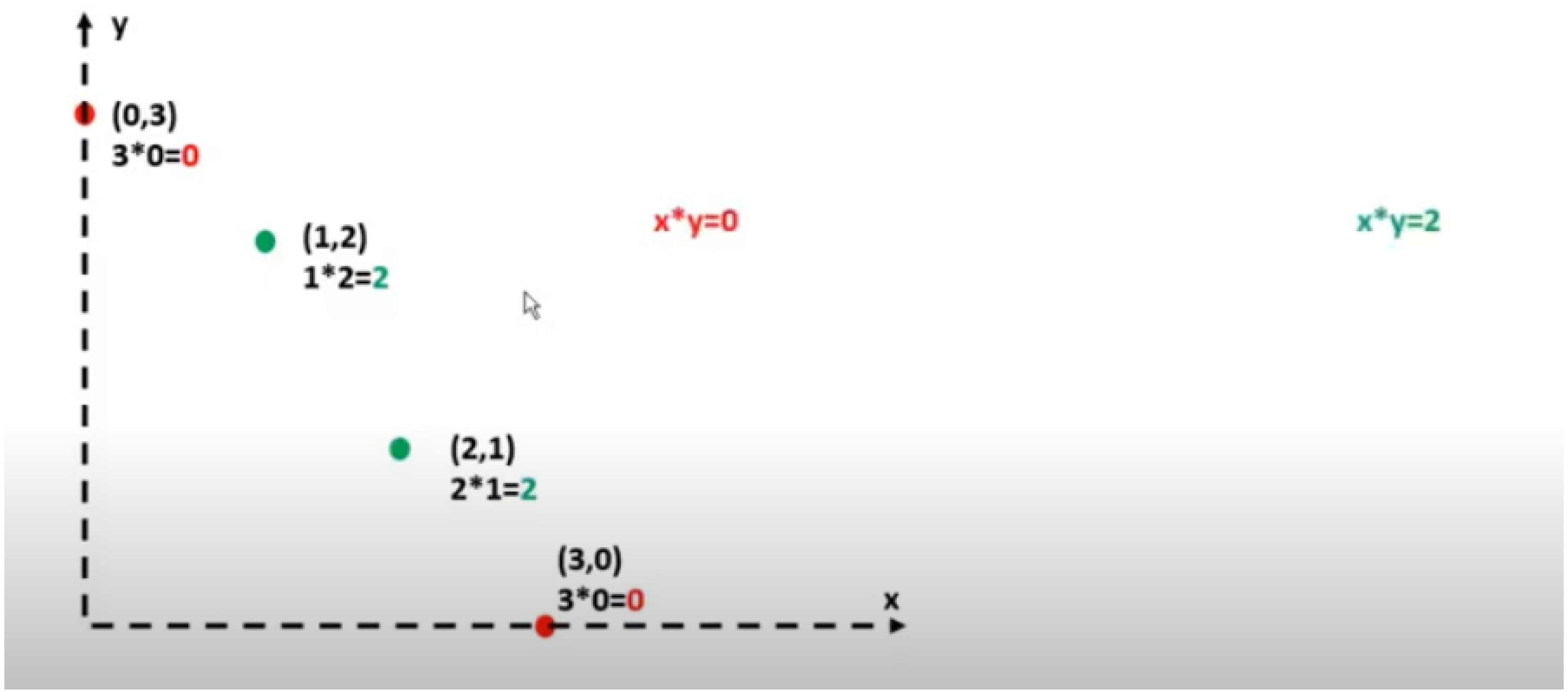
X*Y is best as it represents one class with low values(0) and other class with higher values(2)

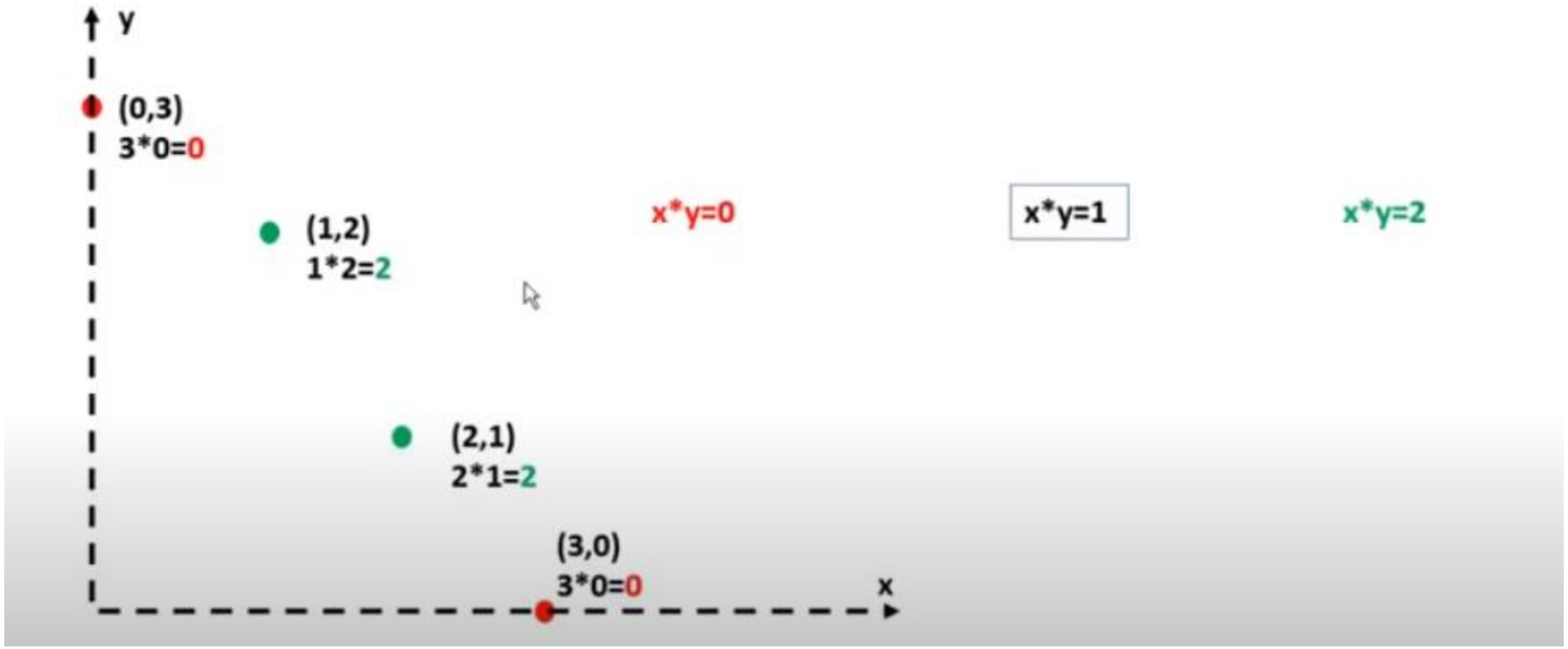


The equation that best represents the graph is:

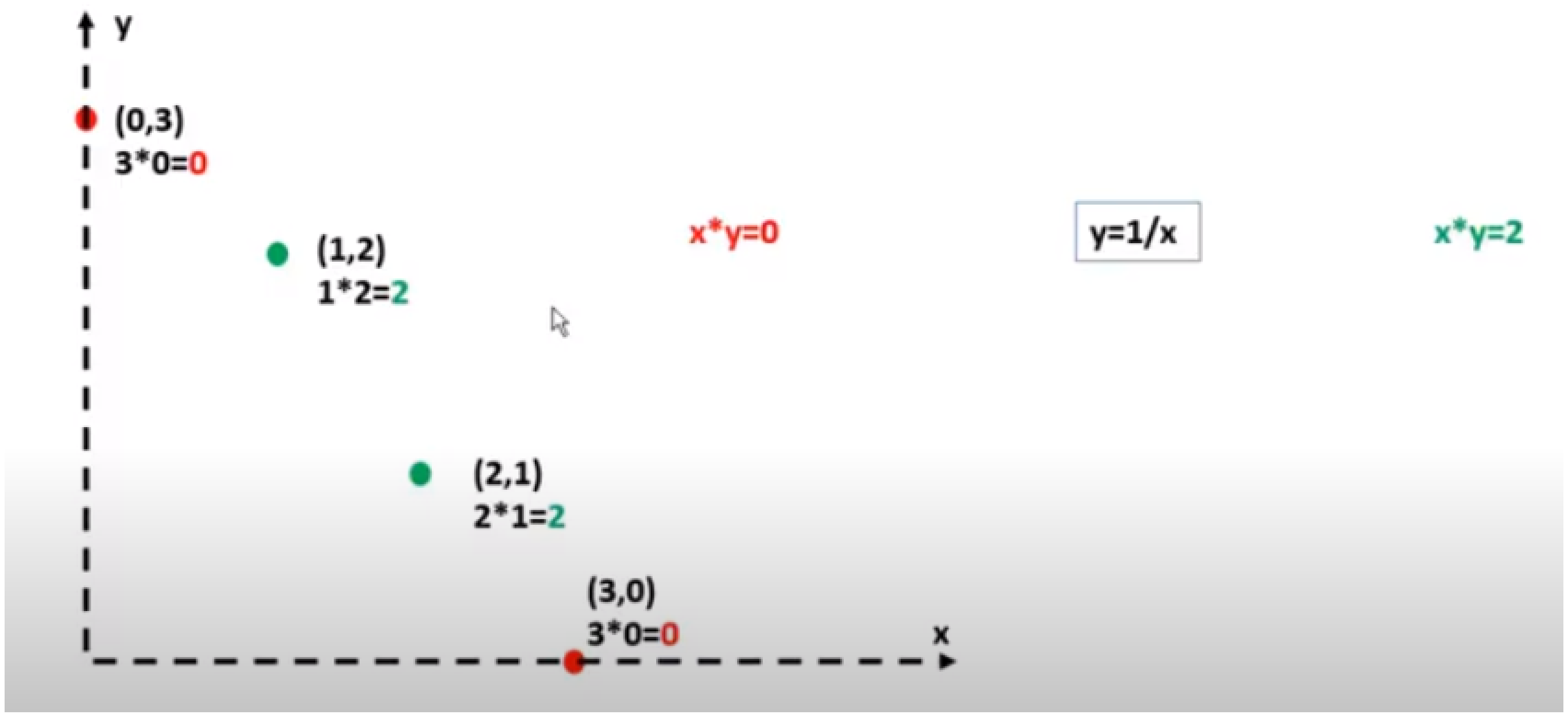
- ☐ X+Y
- ☑ X*Y
- ☐ X^2

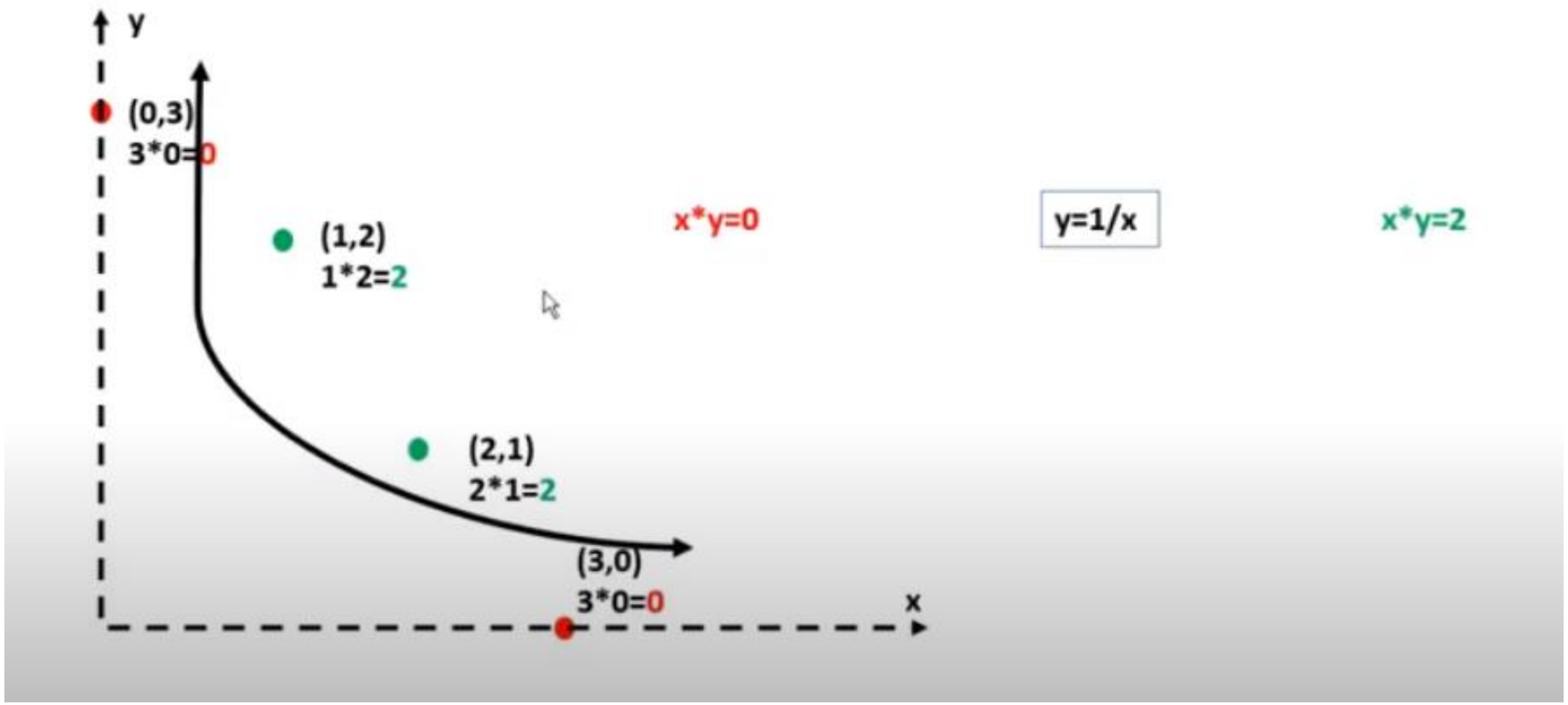
	(0,3)	(1,2)	(2,1)	(3,0)
X+Y	3	3	3	3
X*Y	0	2	2	0
X^2	0	1	4	9


y
(0,3)
3*0=0
(1,2)
1*2=2
x*y=0
x*y=2
(2,1)
2*1=2
(3,0)
3*0=0
x



y
(0,3)
3*0=0
(1,2)
1*2=2
x*y=0
x*y=1
x*y=2
(2,1)
2*1=2
(3,0)
3*0=0
x


y
(0,3)
3*0=0
(1,2)
1*2=2
x*y=0
y=1/x
x*y=2
(2,1)
2*1=2
(3,0)
3*0=0
x



y
(0,3)
3*0=0
(1,2)
1*2=2
(2,1)
2*1=2
(3,0)
3*0=0
x
x*y=0
y=1/x
x*y=2

2D representation

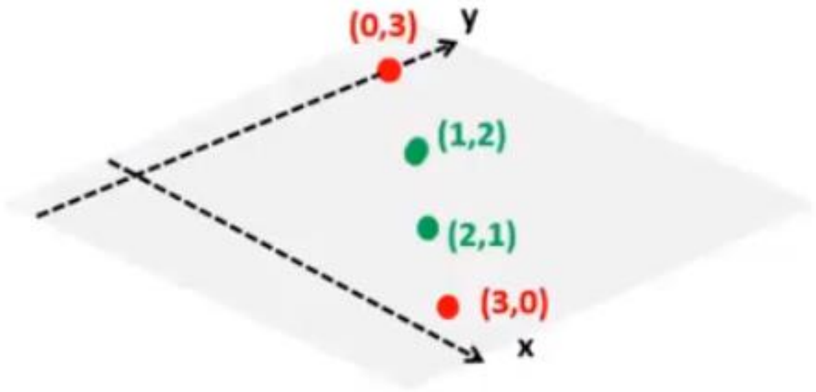
(x,y)

(0,3)

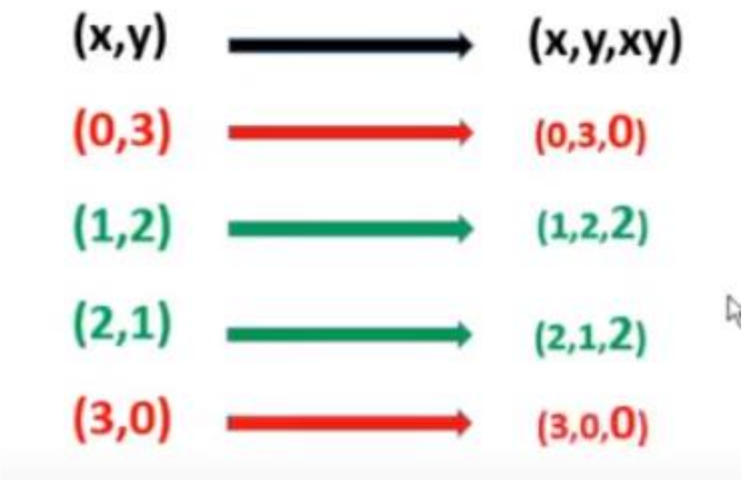
(1,2)

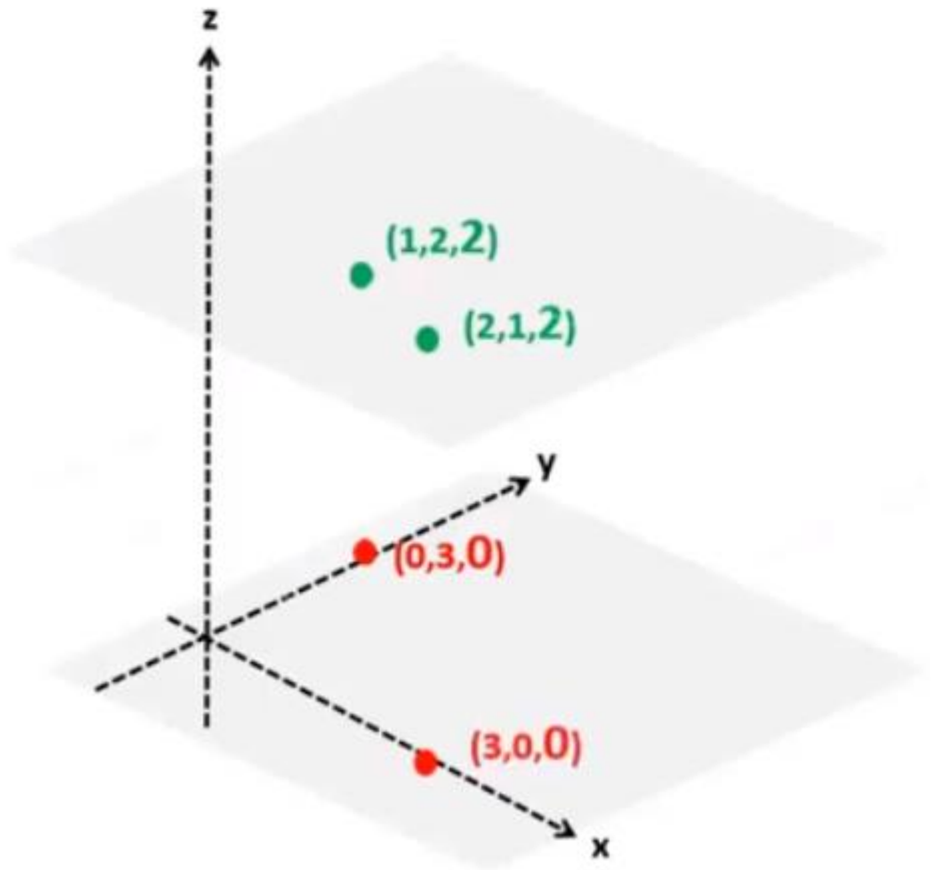
(2,1)

(3,0)



3D representation





Kernel Trick

This **trick**, calculating the high-dimensional relationships without actually transforming the data to the higher dimension, is called **The Kernel Trick**.

Kernel Trick

The Kernel Trick reduces the amount of computation required for **Support Vector Machines** by avoiding the math that transforms the data from low to high dimensions…

Common Kernels
(let **a** and **b** are the given vectors)

Linear: $K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b}$

Polynomial: $K(\mathbf{a}, \mathbf{b}) = \left(\gamma \mathbf{a}^T \mathbf{b} + r\right)^d$

Gaussian RBF: $K(\mathbf{a}, \mathbf{b}) = \exp\left(-\gamma \| \mathbf{a} - \mathbf{b} \|^2\right)$

Sigmoid: $K(\mathbf{a}, \mathbf{b}) = \tanh\left(\gamma \mathbf{a}^T \mathbf{b} + r\right)$