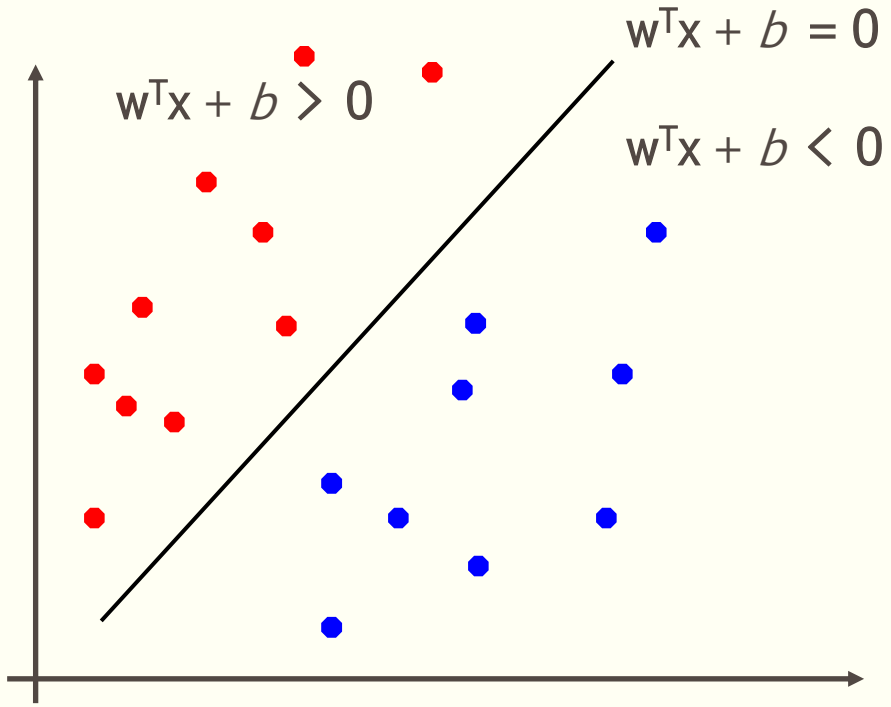
Machine learning

Prepared by : Dr. Hanaa Bayomi
Updated By: Prof Abeer ElKorany

Lecture 8: SVM

Perceptron Revisited: Linear Separators

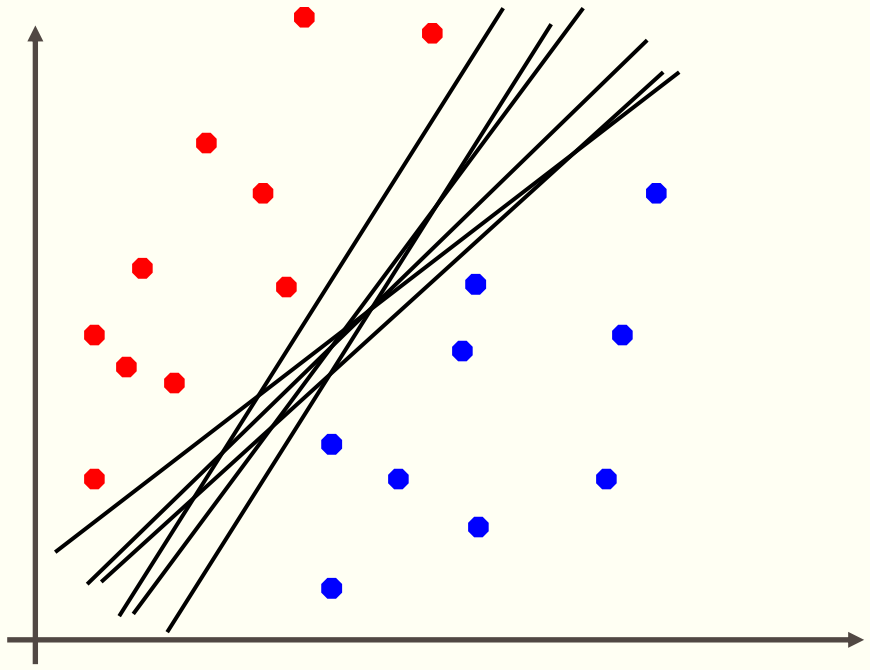
- Binary classification can be viewed as the task of separating classes in feature space:



$$f(x) = \text{sign}(w^T x + b)$$

Linear Separators

- Which of the linear separators is optimal?

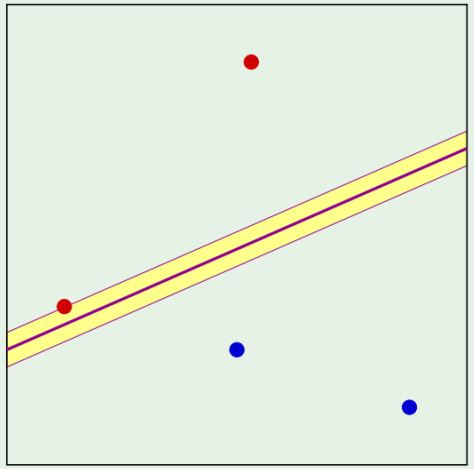


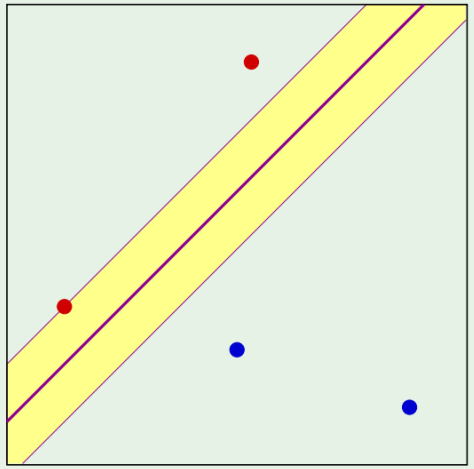
Linear separation of Data

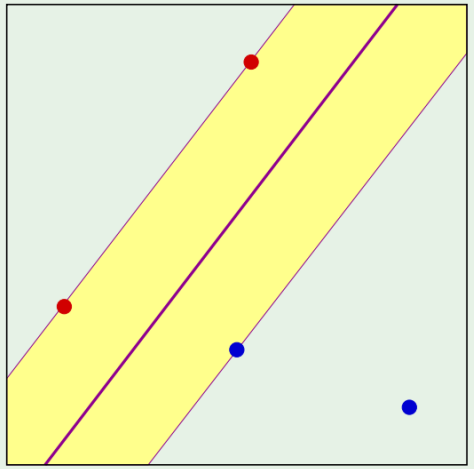
Linearly separable data

Different separating lines

Which is best?







Two questions:

1. Why is bigger margin better?
2. Which $\mathbf{w}$ maximizes the margin?

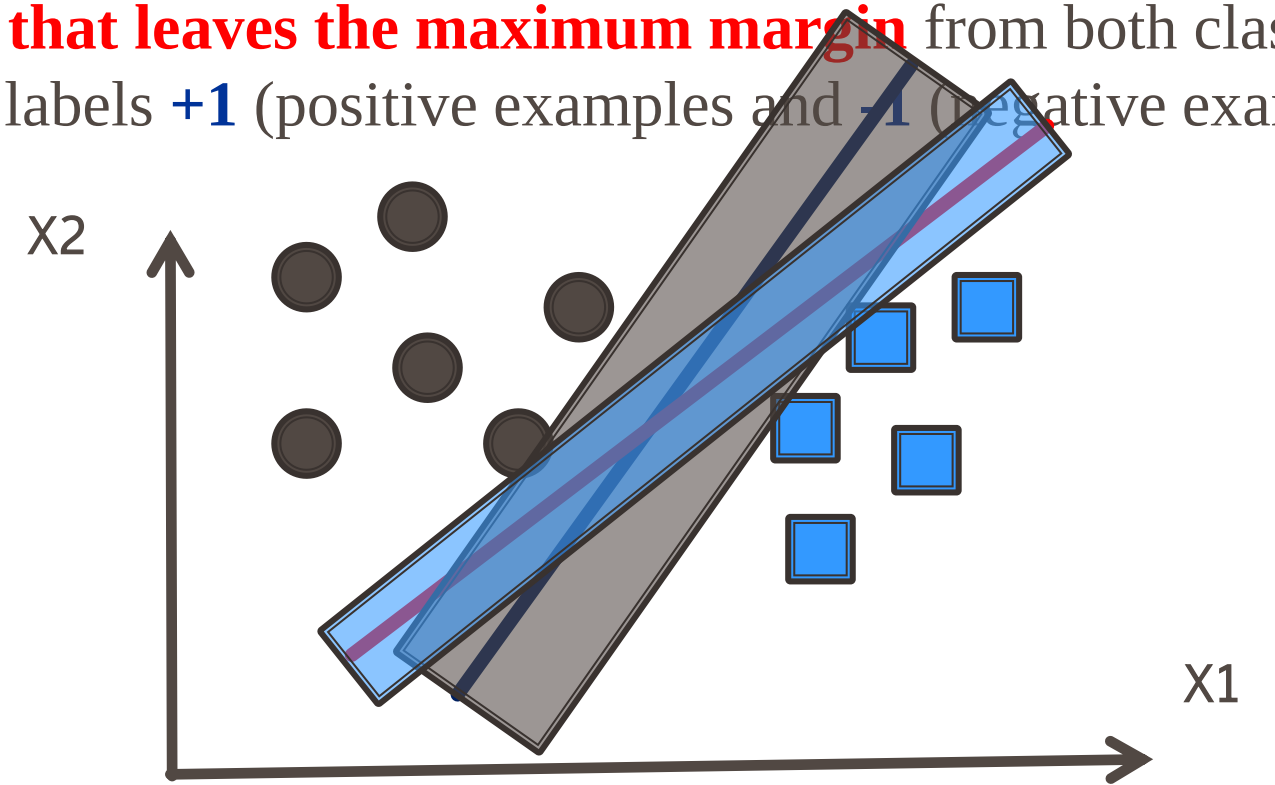
SVM

- SVM for **linearly separable** binary set

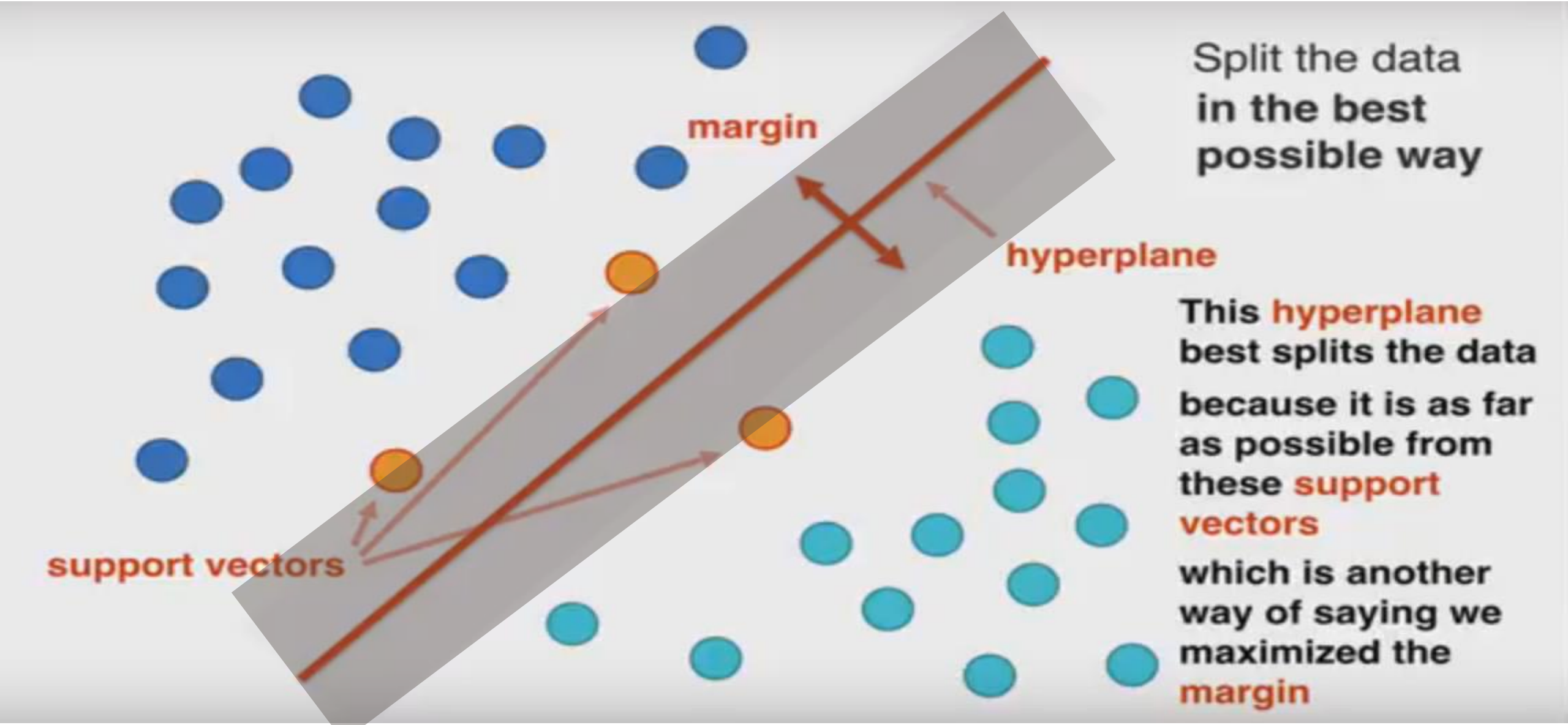
- **Main Goal** to design a hyper plane that classify all training vectors into two classes

•Support vectors are the data points that are most important for the construction of the optimal hyperplane that separates different classes in a binary classification problem.

- *The best model* **that leaves the maximum margin** from both classes
- the two classes labels **+1** (positive examples and **-1** (negative examples)

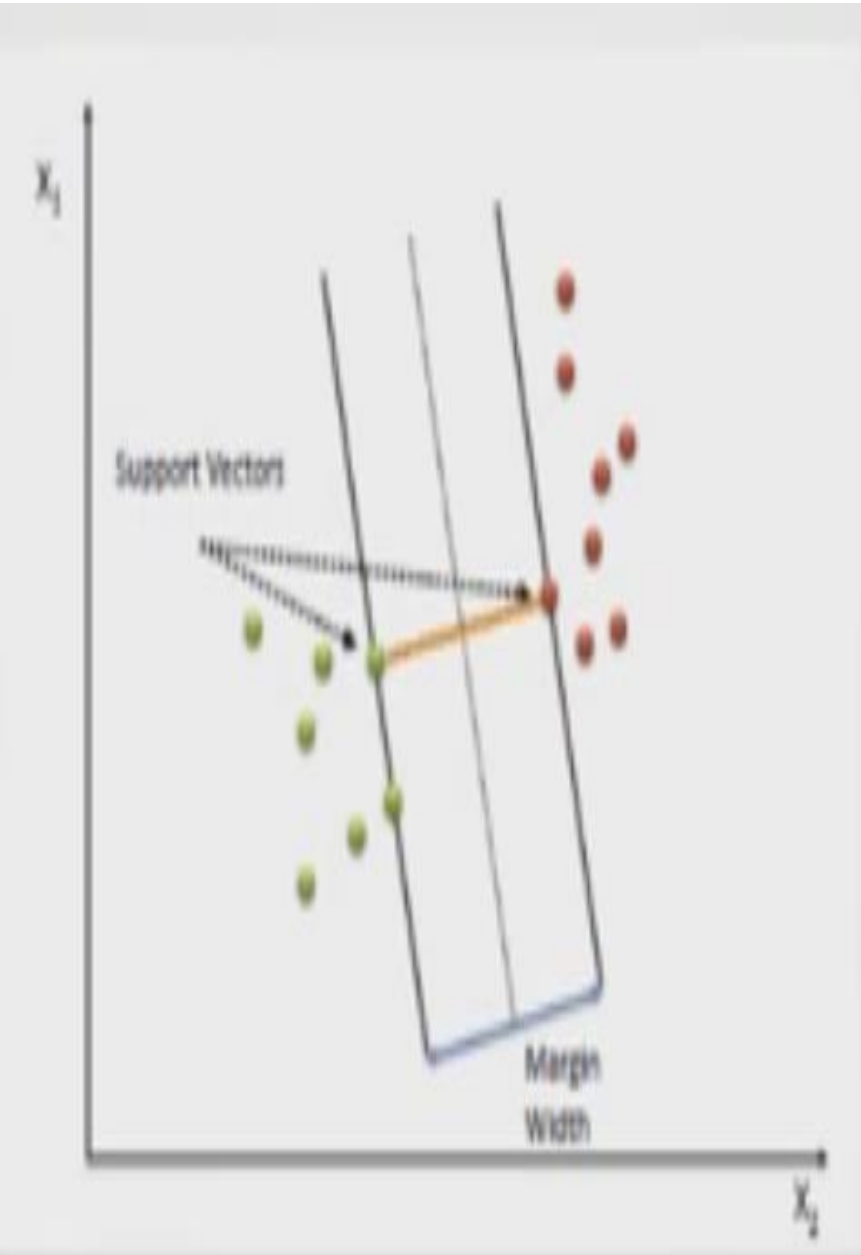


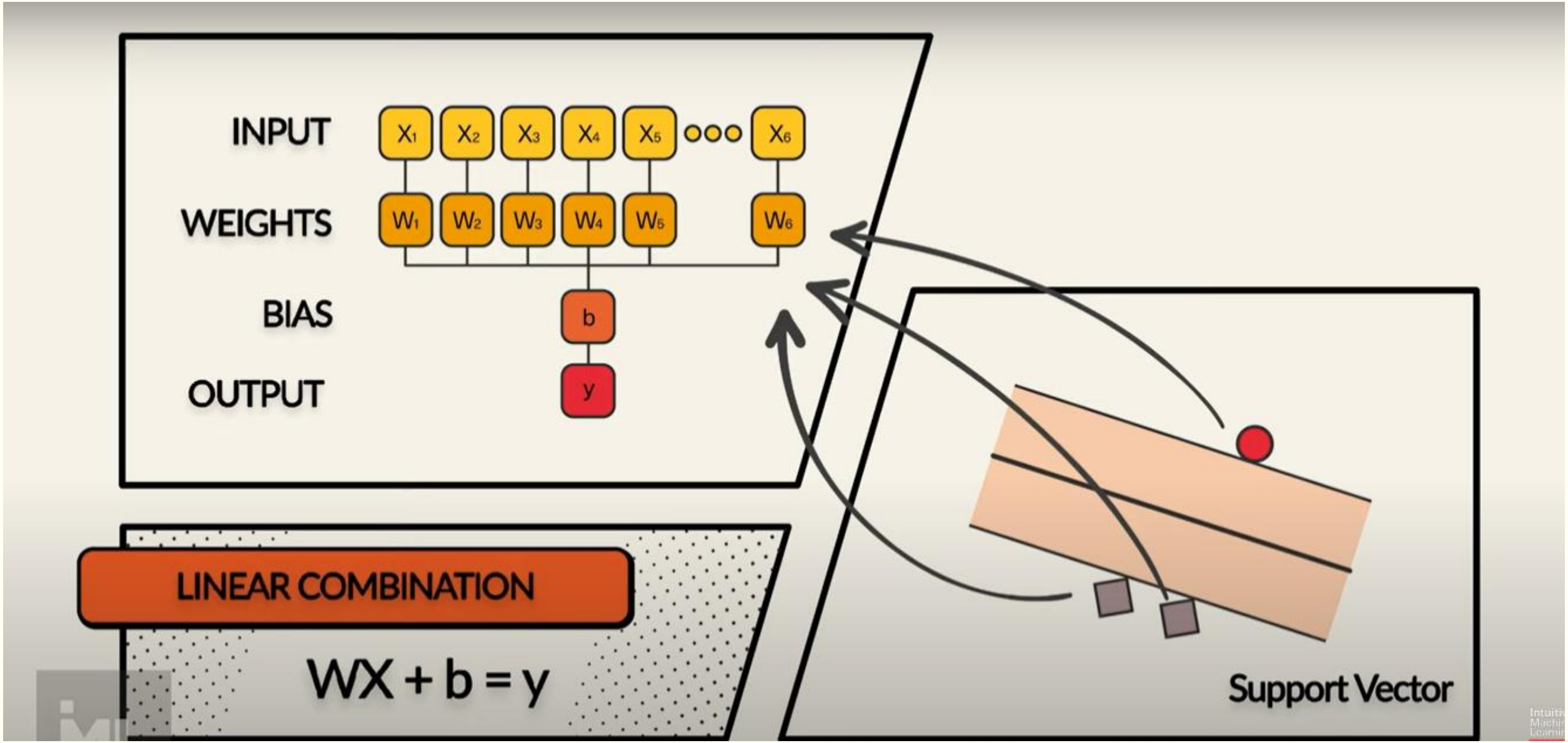
This is a constrained optimization problem

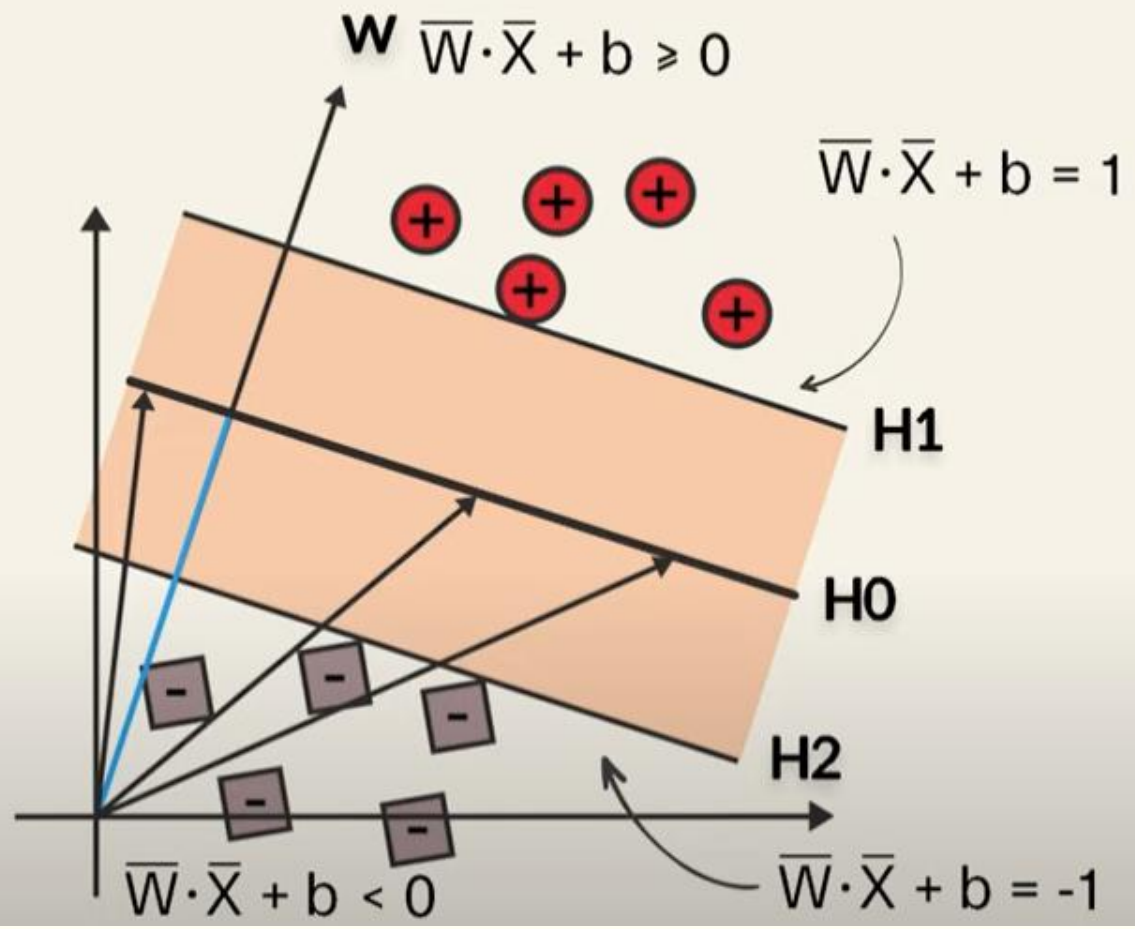


Intuition behind SVM

- Points (instances) are like vectors $p = (x1, x2, \ldots, xn)$
- 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

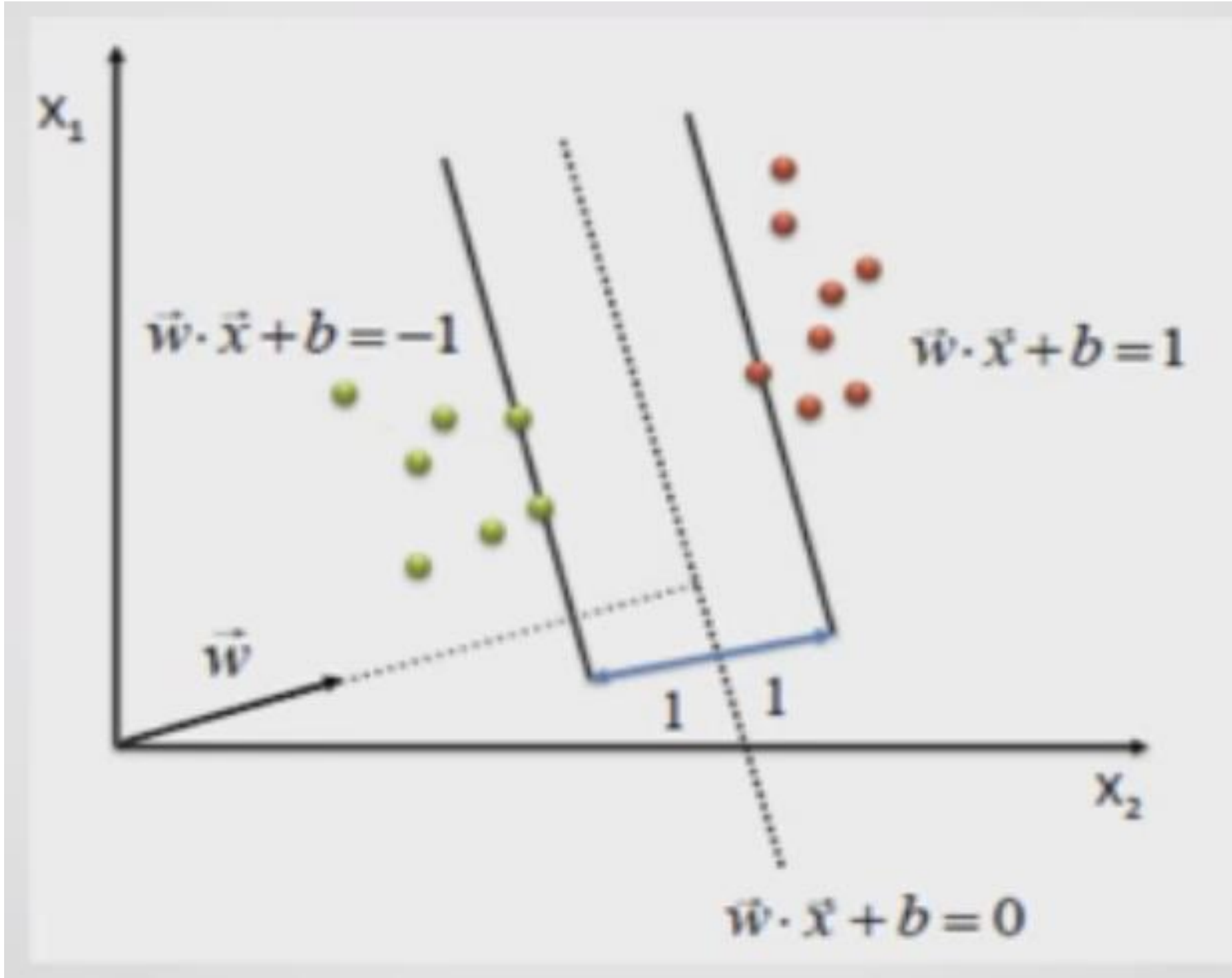



INPUT
X1
X2
X3
X4
X5
X6
WEIGHTS
W1
W2
W3
W4
W5
W6
BIAS
b
OUTPUT
y
LINEAR COMBINATION
WX + b = y
Support Vector

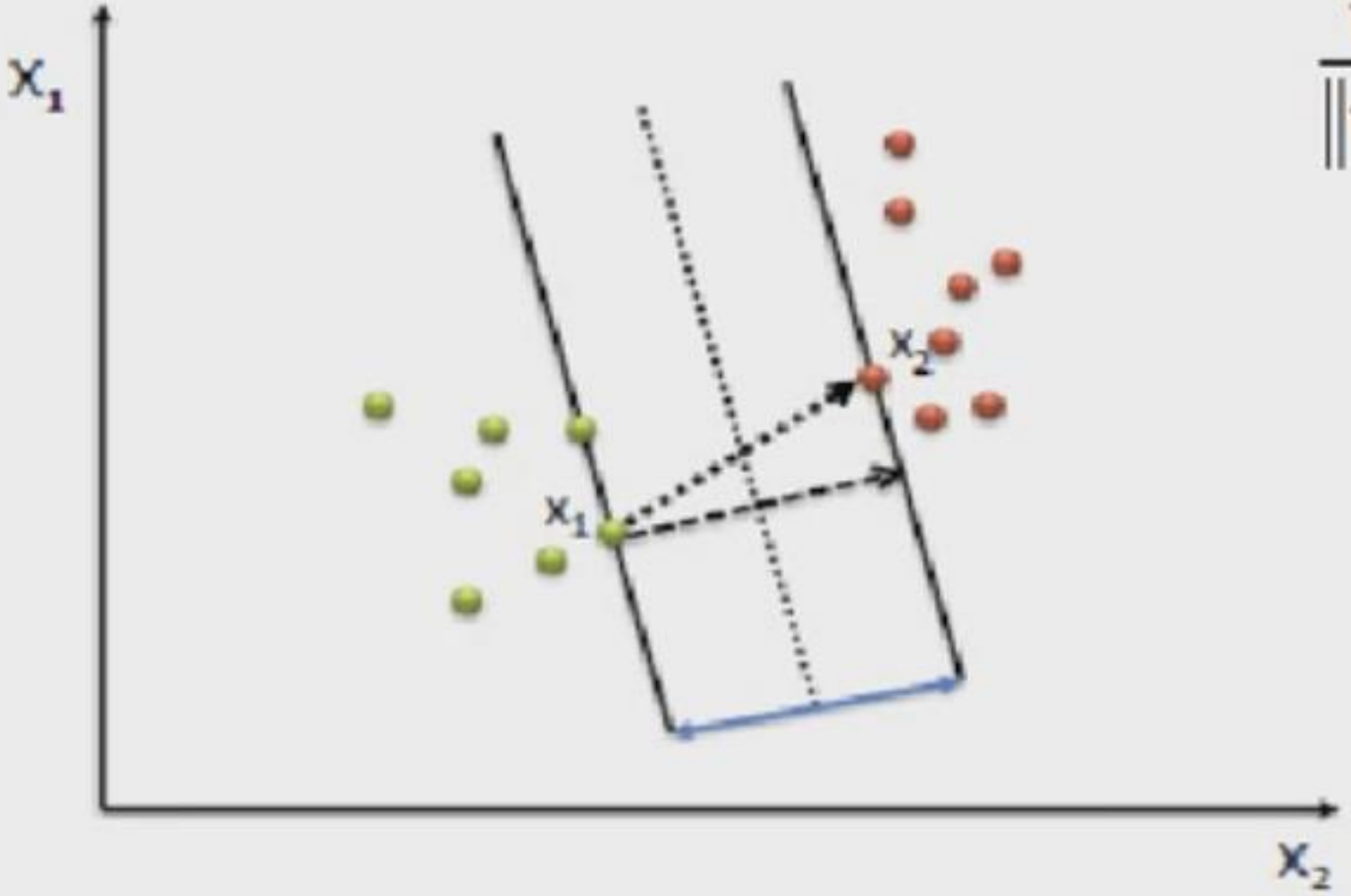


$$\overline{W} \cdot \overline{X} + b = 0$$

Hyperplane Equation


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 of W



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

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

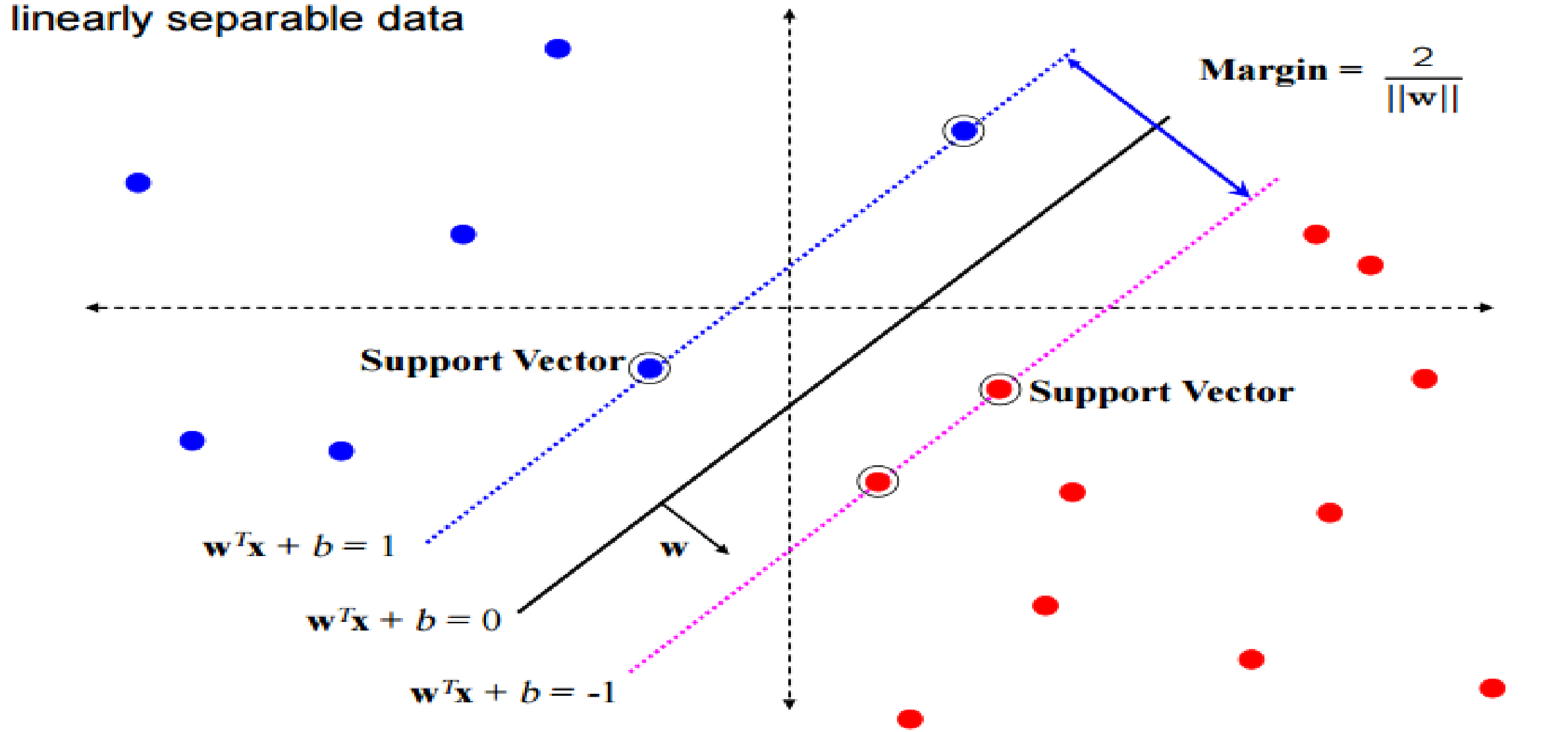
$$w\cdot x_1+b=-1$$

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

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

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

Support Vector Machine



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 →

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$y_n\left(\mathbf{w}^\intercal \mathbf{x}_n + b\right) \geq 1 \quad \text{for} \quad n = 1, 2, \ldots, N \quad \forall x_i$$

← 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

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

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:

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

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

Characteristics of the Solution

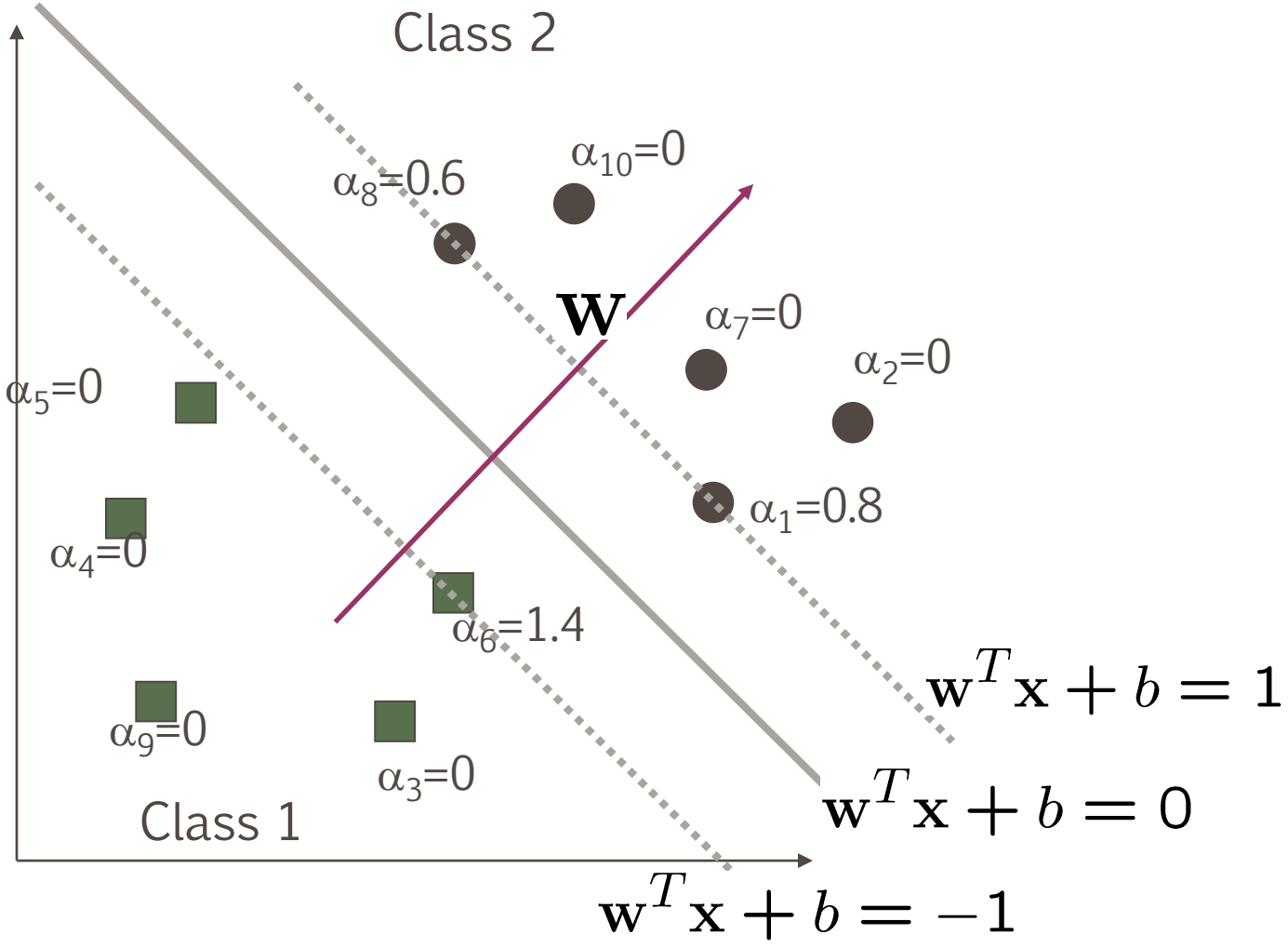
- Many of the α_i are zero (see next page for example)
  - **w** is a linear combination of a small number of data points
  - This "sparse" representation can be viewed as data compression as in the construction of knn classifier

- $\mathbf{x}_i$ with non-zero α_i are called support vectors (SV)
  - The decision boundary is determined only by the SV
  - L$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$ dices of the *s* support vectors. We can write

- For testing wi$\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$
  - Compute

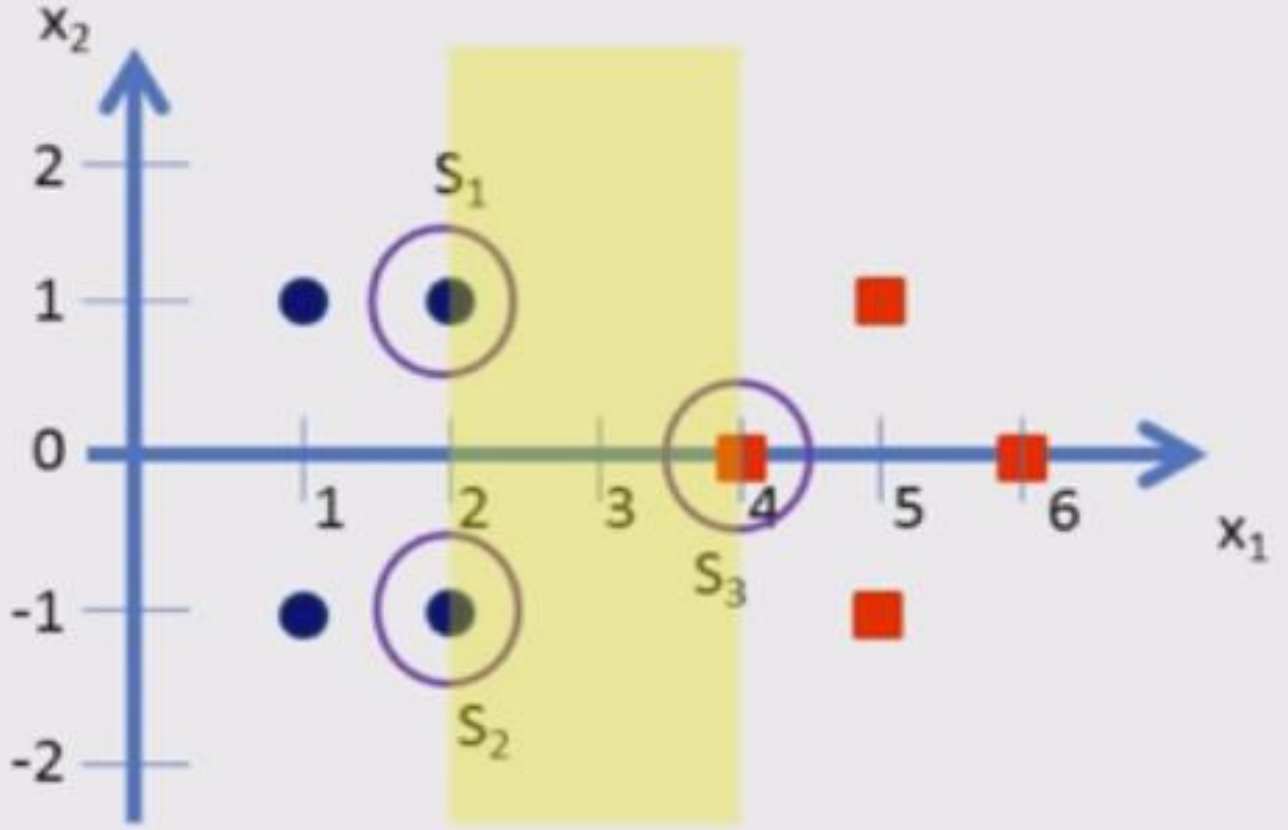
    and classify **z** as class 1 if the sum is positive, and class 2 otherwise
  - Note: **w** need not be formed explicitly

A Geometrical Interpretation



Example

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



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

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

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

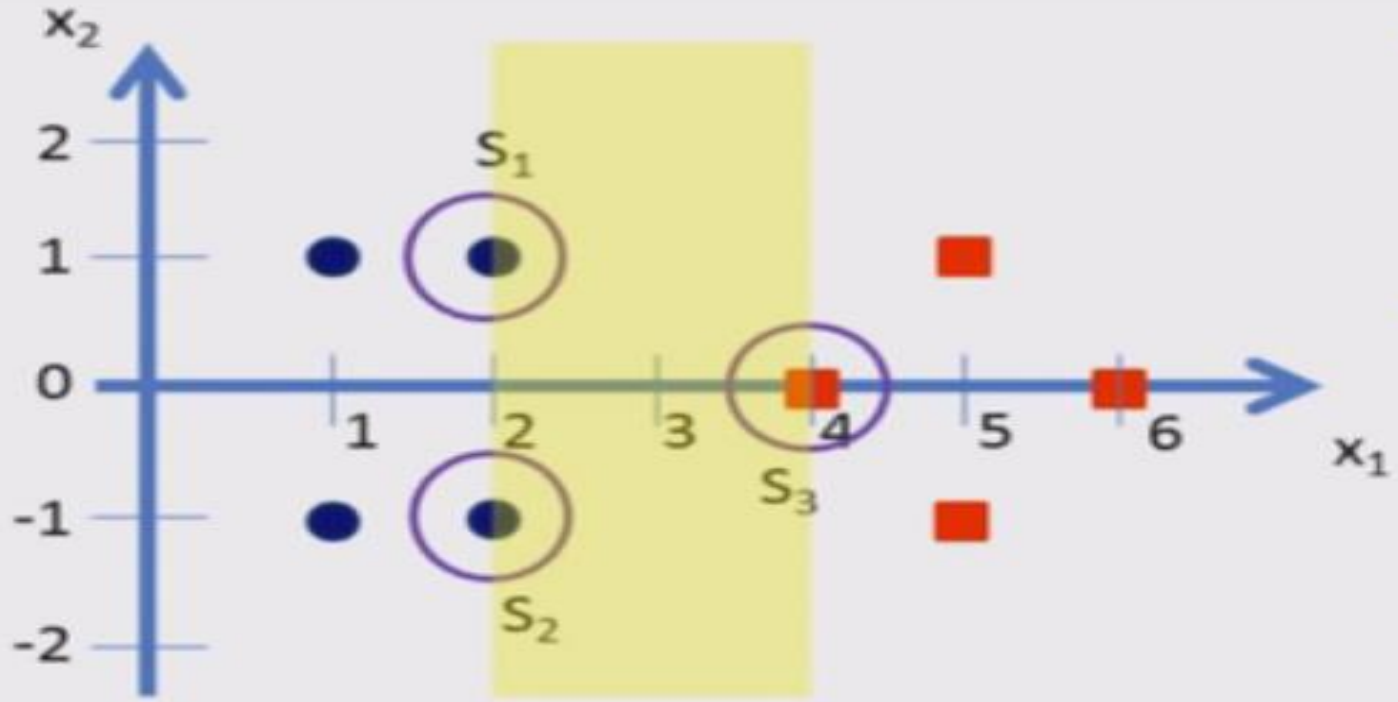
Example

- 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:

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

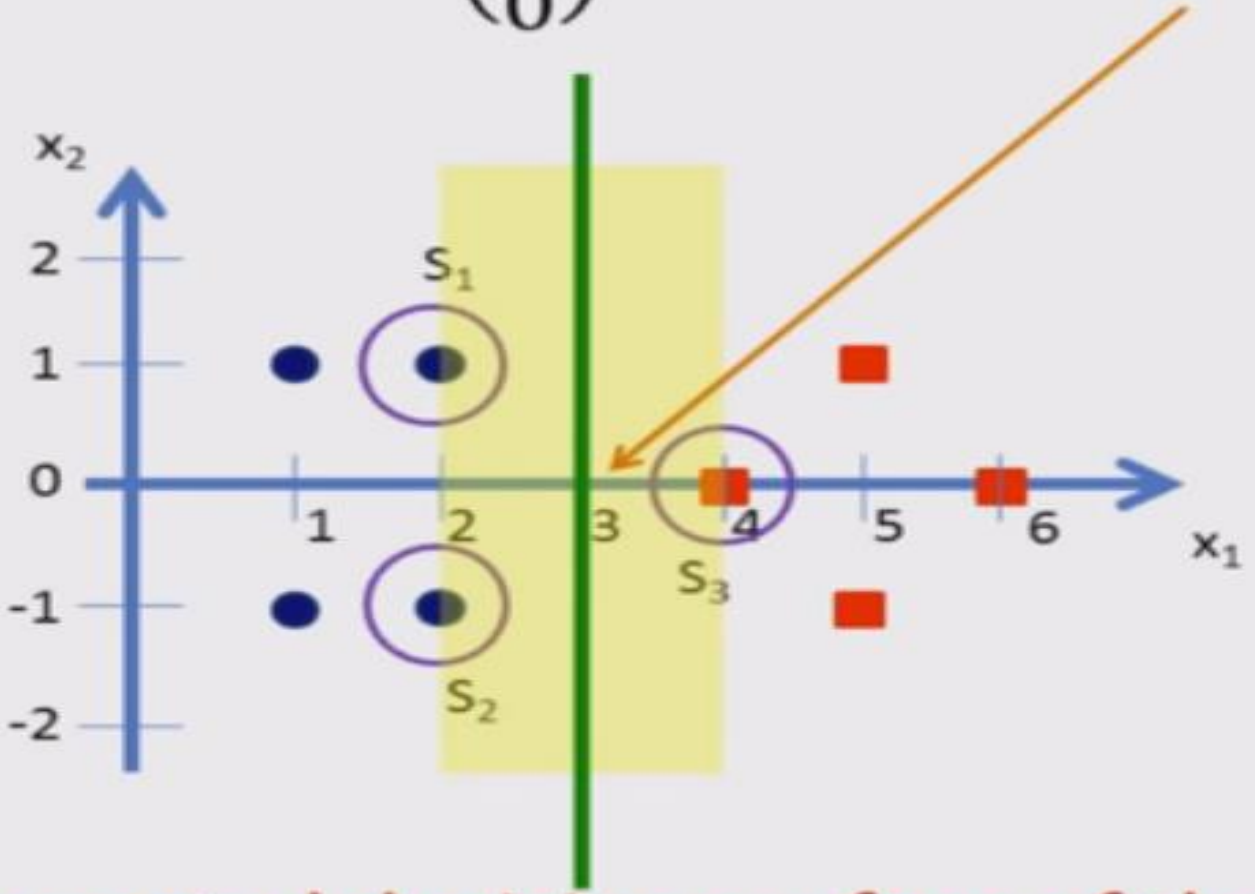
$$\widetilde{w} = (-3.25). \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25). \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5). \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

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

- 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

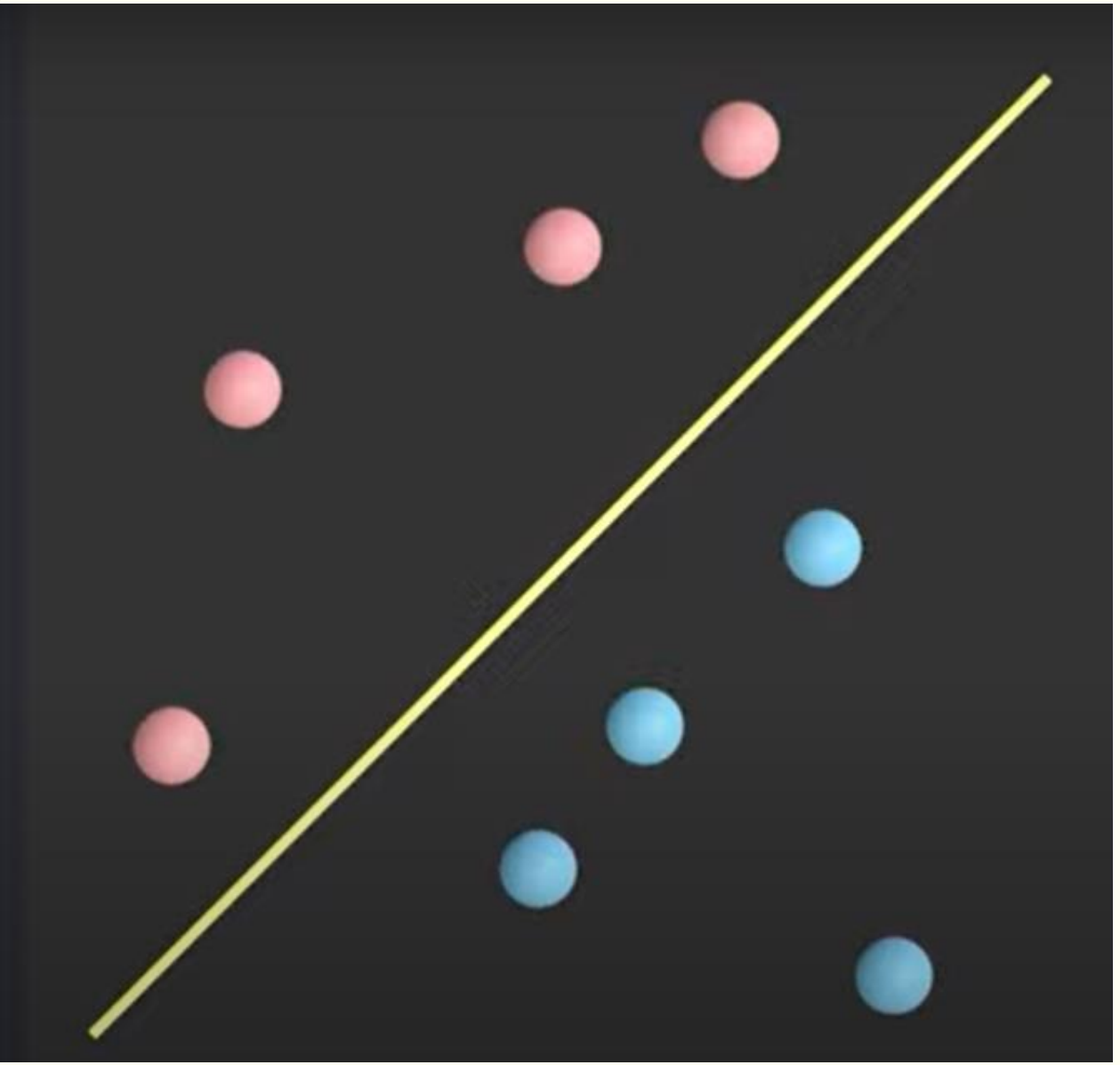
- $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and offset $b = -3$.



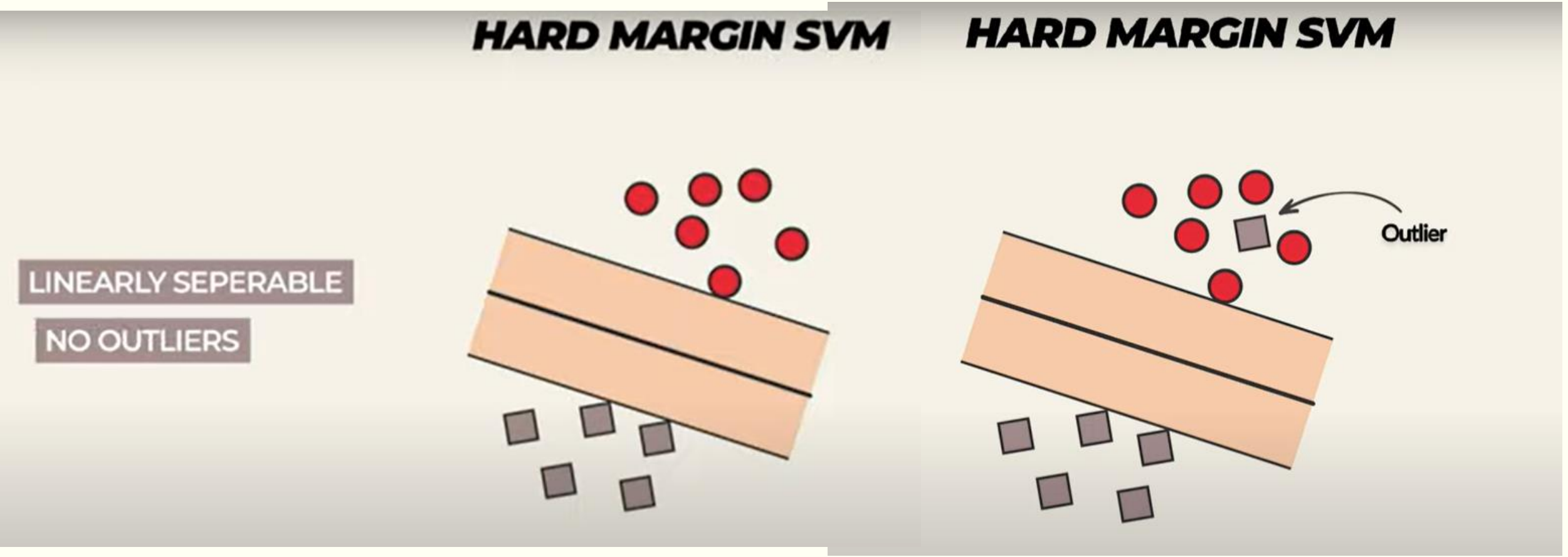
- This is the expected decision surface of the LSVM.

Implementing of SVM

```
# features
X = [
    [-3, -1],
    [0, -2],
    [-2.5, 2],
    [-1, -1],
    [3, .5],
    [.5, 3],
    [-3, -3],
]


# labels
y = [0, 1, 0, 1, 1, 0, 1]


# fit
clf = svm.SVC(kernel='linear').fit(X, y)


# predict
clf.predict([[2, 4]]) => 0
```



Hard SVM


HARD MARGIN SVM
LINEARLY SEPERABLE
NO OUTLIERS
HARD MARGIN SVM
Outlier

Soft Margin SVM

SVM Algorithm

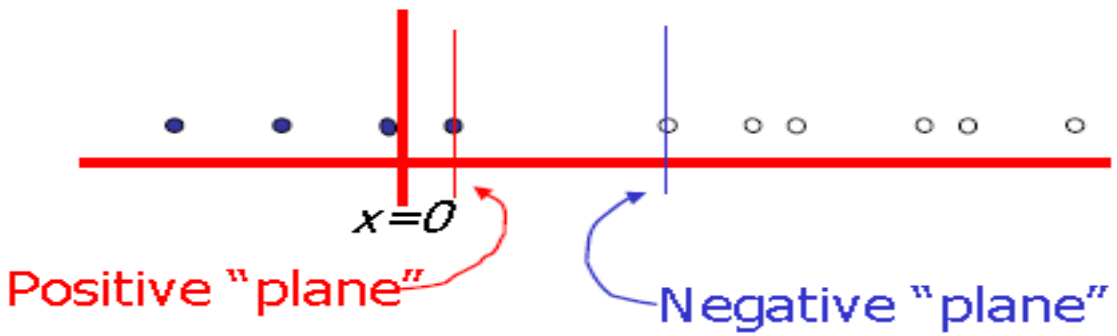
1- Define an optimal hyperplane: maximize margin

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

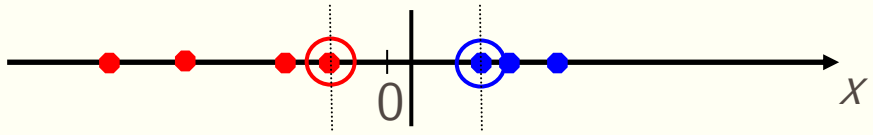
Suppose we're in 1-dimension

Not a big surprise

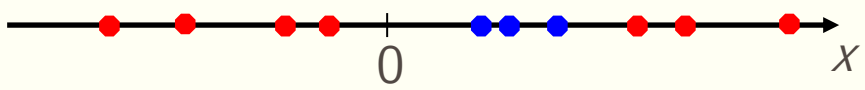


Non-linear SVMs

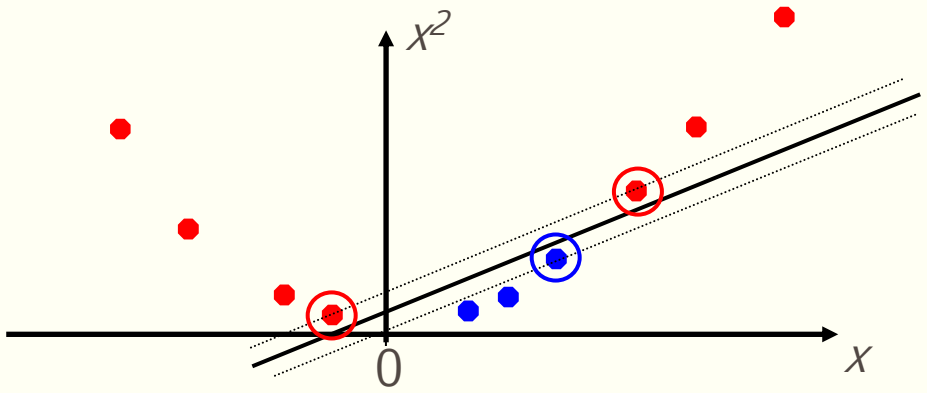
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?



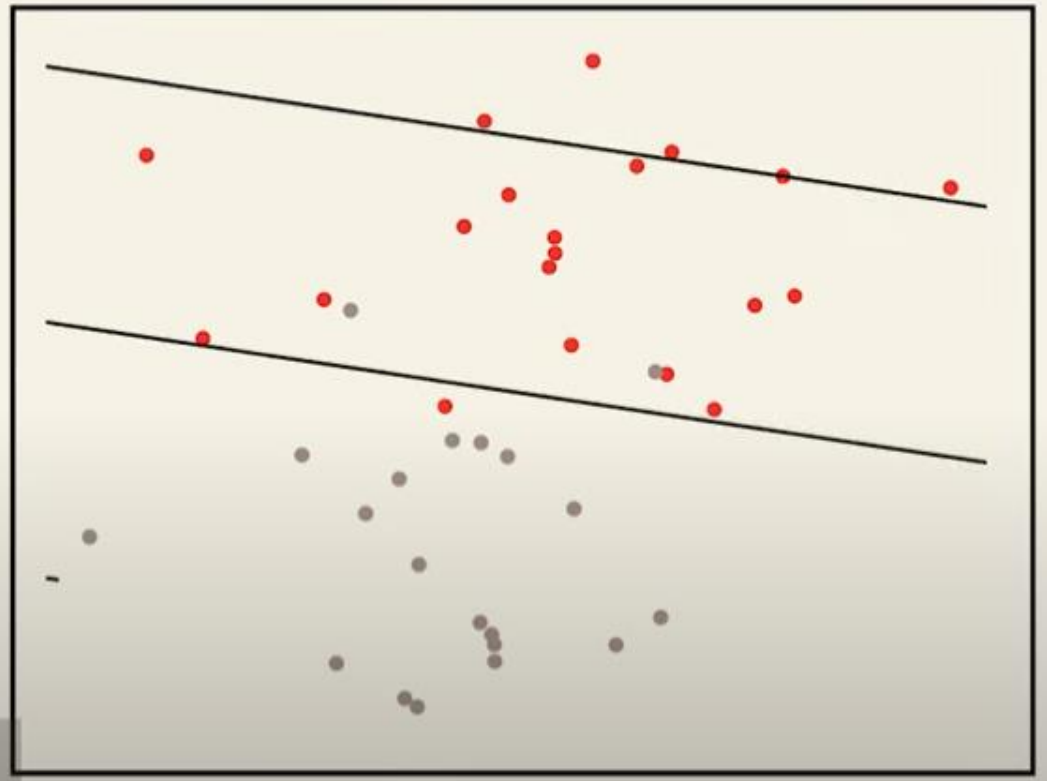
- How about… mapping data to a higher-dimensional space:



Regularization

SIMULATE WITH SCIKIT LEARN

C = 0.003



Constraints

$$y_i (\overline{W} \cdot \overline{X}_i + b) \geq 1 - \zeta_i$$

$$(\zeta_i \geq 0 , i = 1, \ldots , m)$$

A smaller C emphasizes the importance of ζ and a larger C diminishes the importance of ζ.

Function needs to be optimized

$$\frac{1}{2} \overline{W} \cdot \overline{W} + C \sum_{i=1}^{m} \zeta_i$$

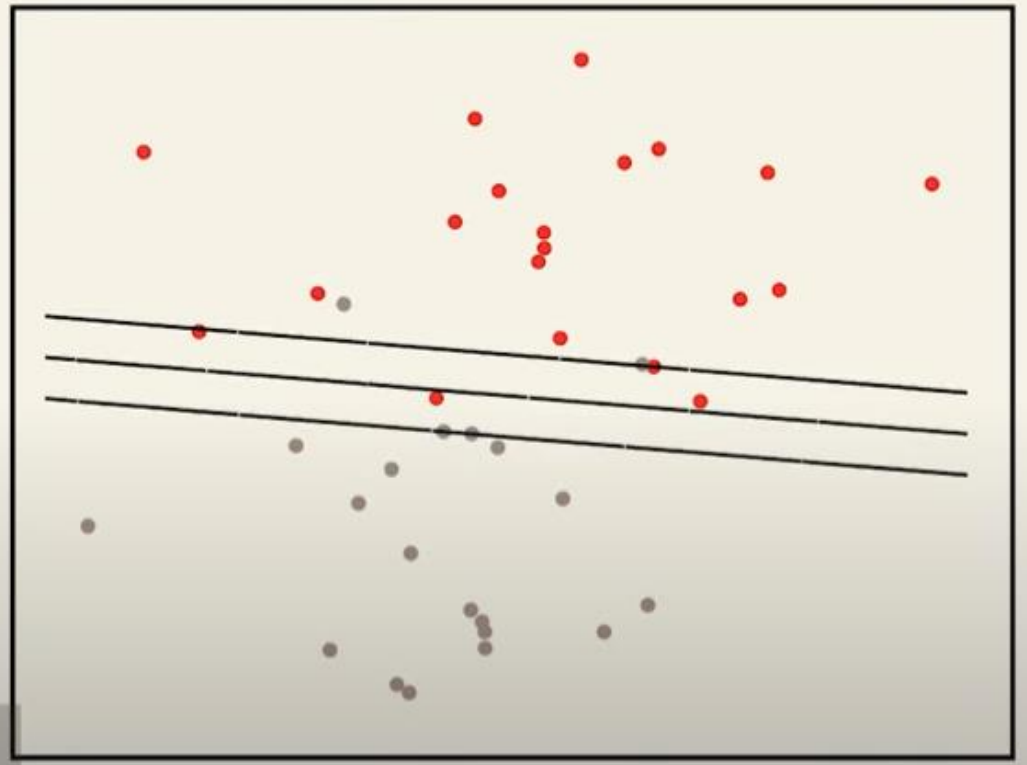
L1 regularization

Change of value of C === Penalize the misclassification

Regularization

SIMULATE WITH SCIKIT LEARN

C = 100



Constraints

$$y_i (\overline{W} \cdot \overline{X}_i + b) \geq 1 - \zeta_i$$

$$(\zeta_i \geq 0 , i = 1, \ldots , m)$$

Function needs to be optimized

$$\frac{1}{2} \overline{W} \cdot \overline{W} + C \sum_{i=1}^{m} \zeta_i$$

A smaller C emphasizes the importance of ζ and a larger C diminishes the importance of ζ .

L1 regularization

Intuitive Machine

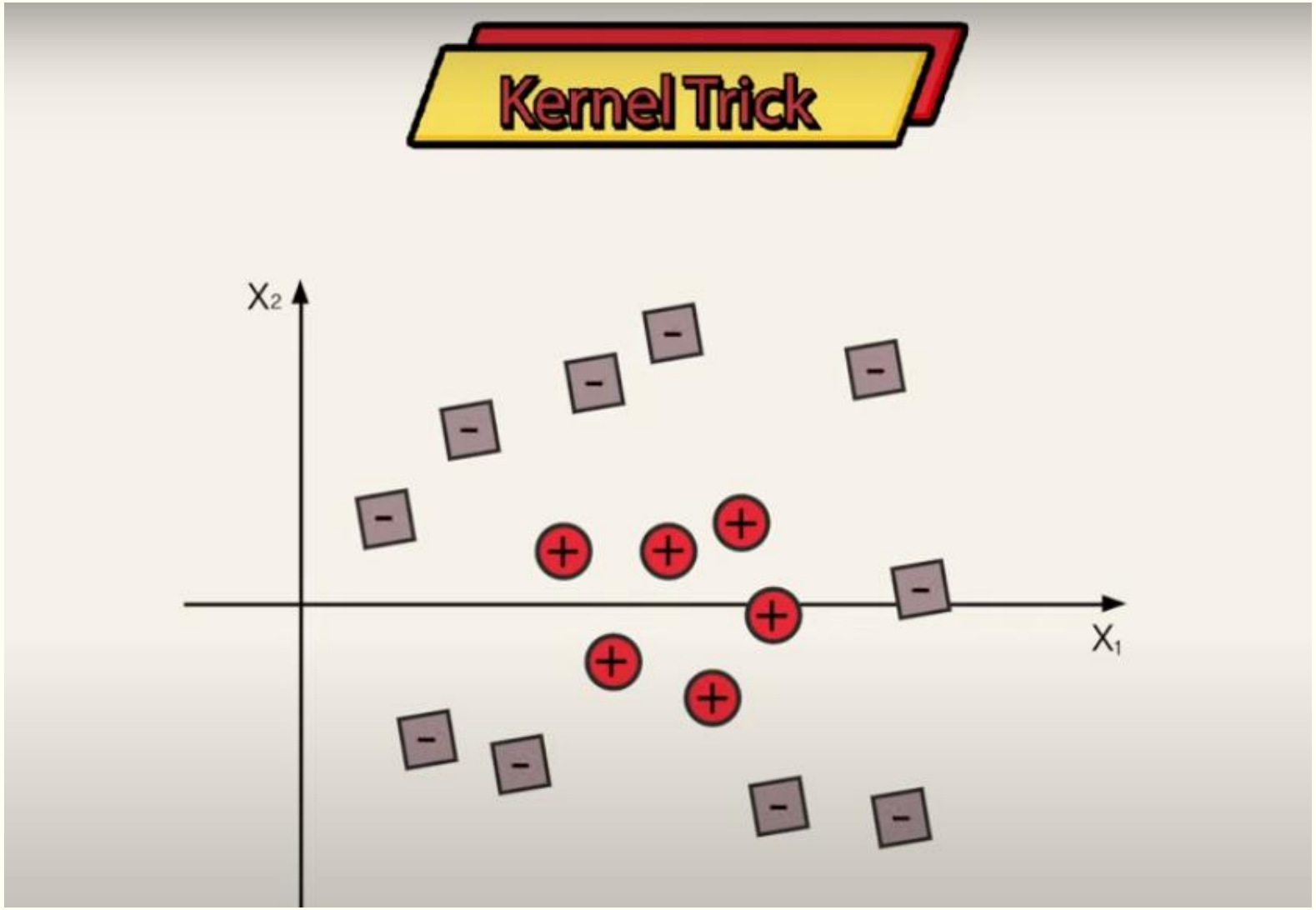
Change of value of C === Penalize the misclassification

Kernel Trick

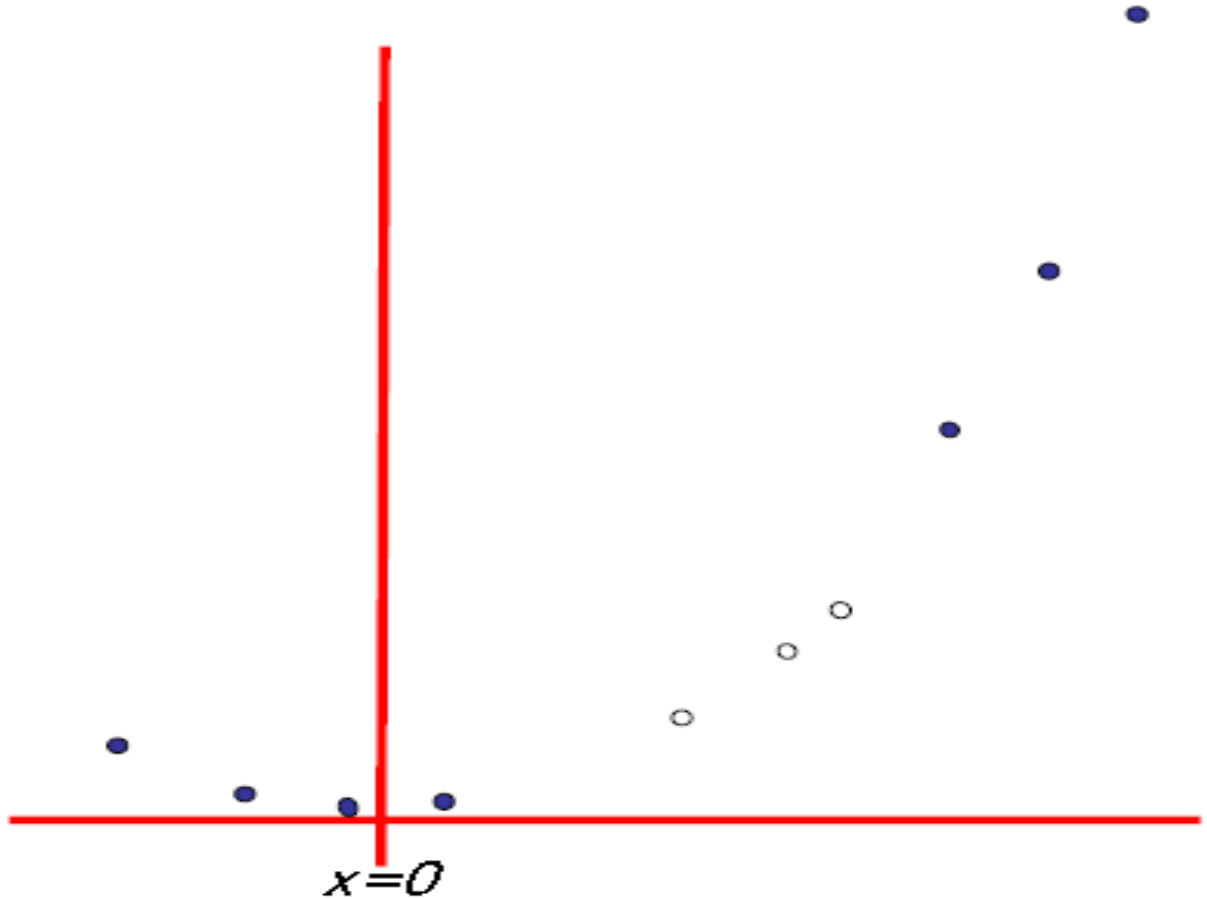
Harder 1-dimensional dataset

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

What can be done about this?


Kernel Trick
X2
X1

Harder 1-dimensional dataset

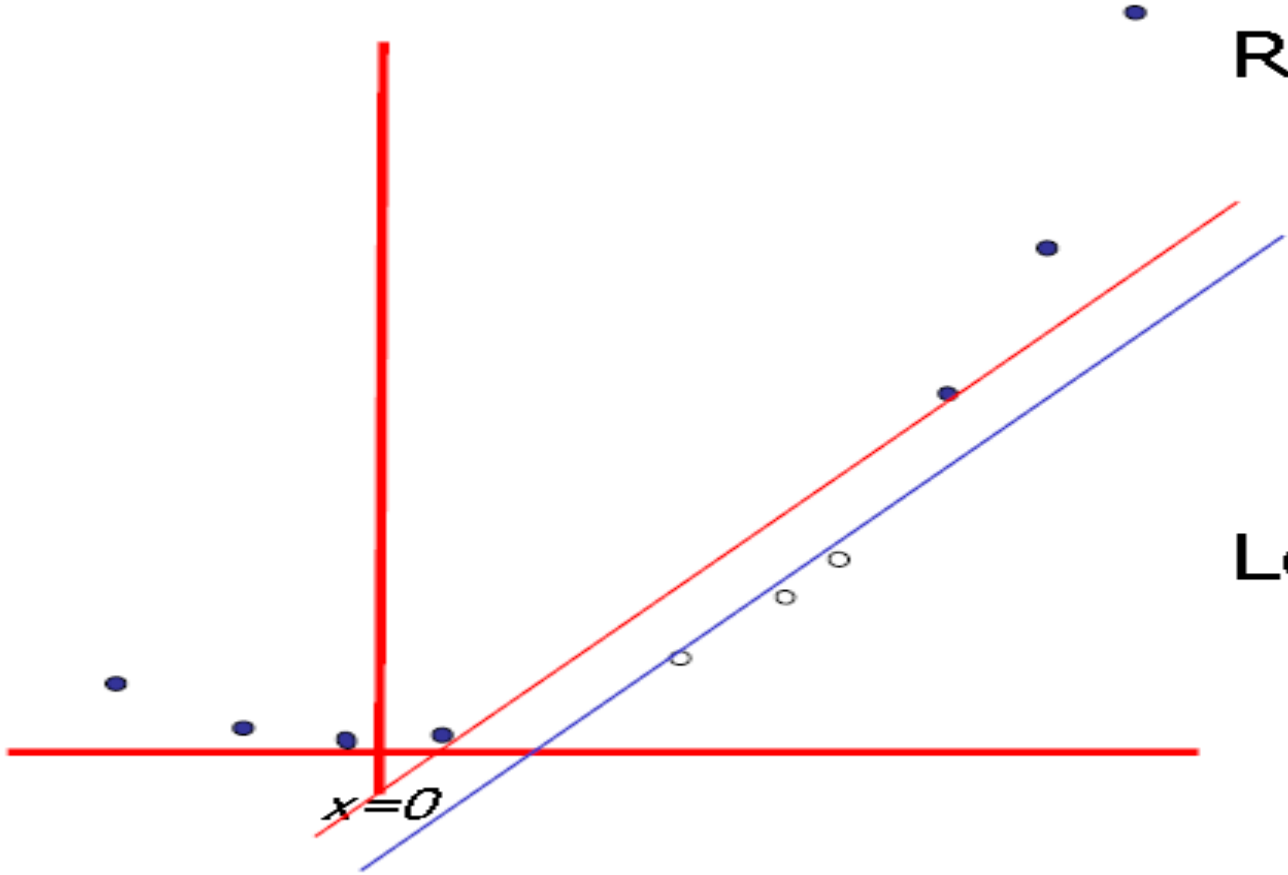


Remember how permitting non-linear basis functions made linear regression so much nicer?

Let's permit them here too

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

Harder 1-dimensional dataset



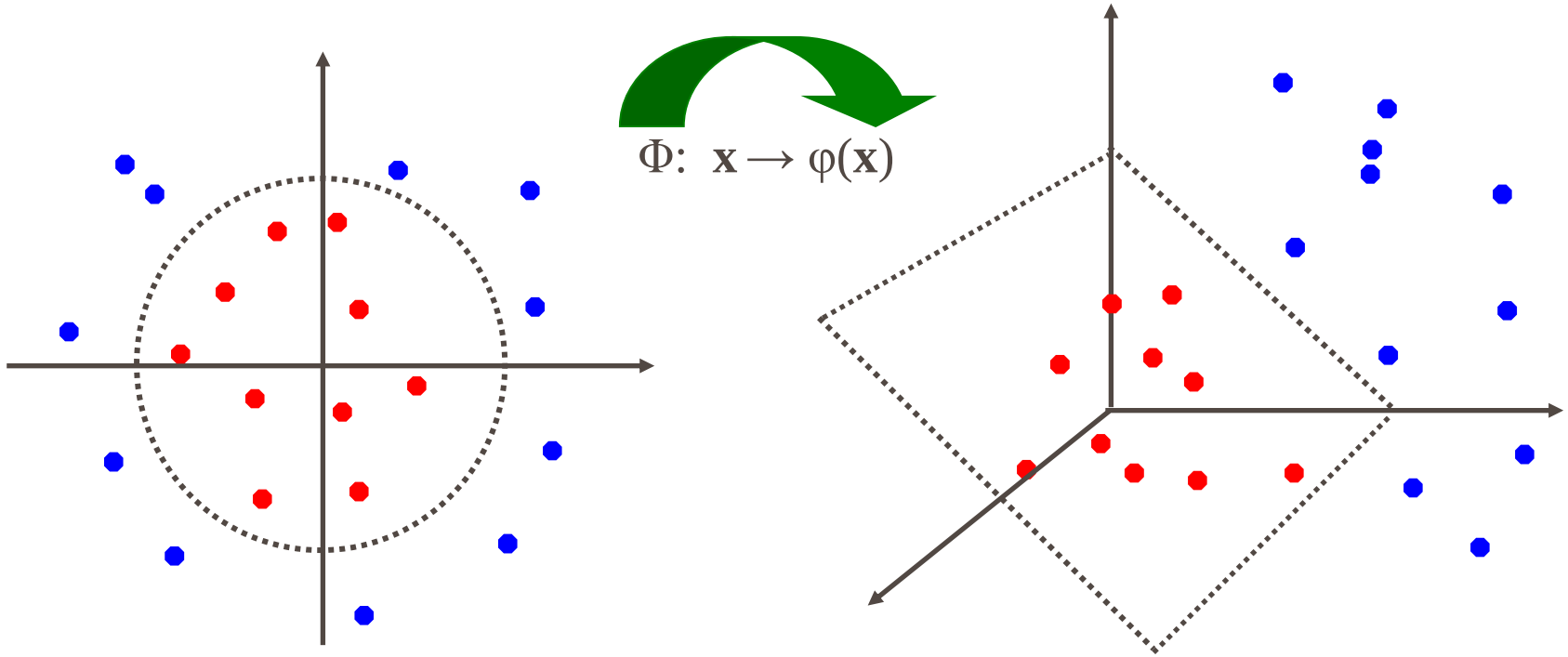
Remember how permitting non-linear basis functions made linear regression so much nicer?

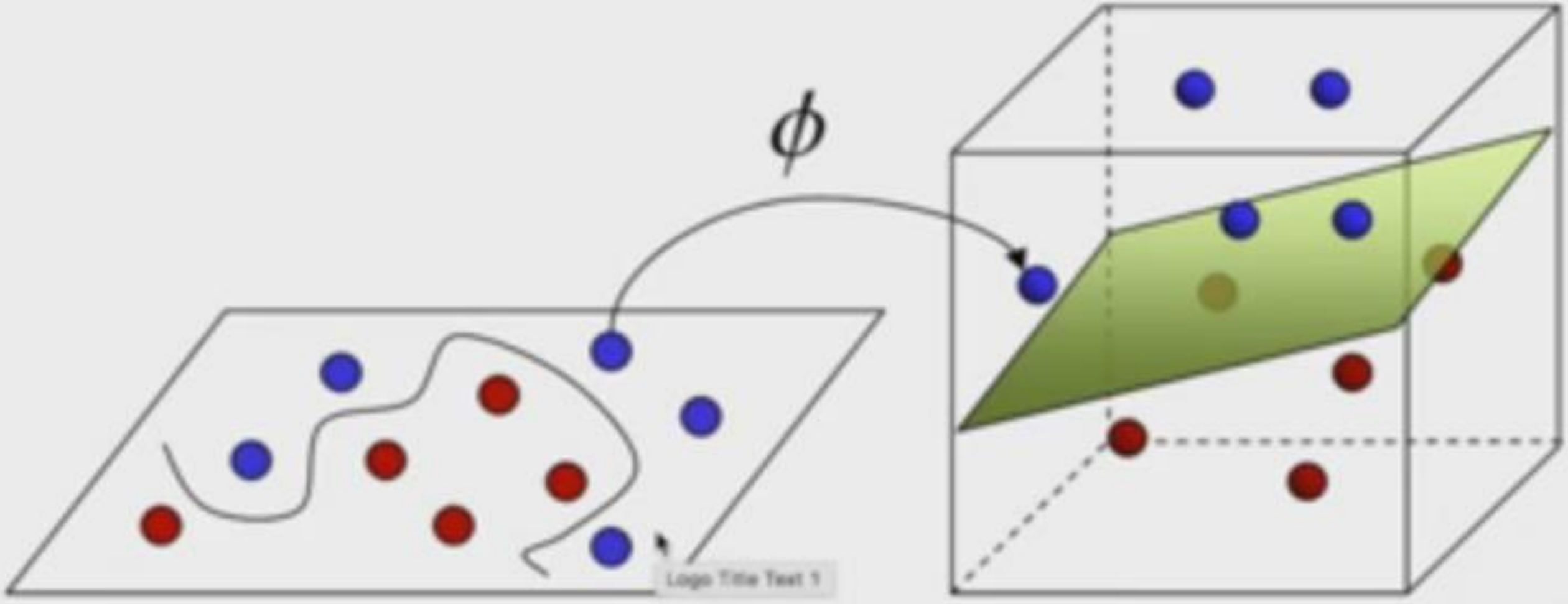
Let's permit them here too

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

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:





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

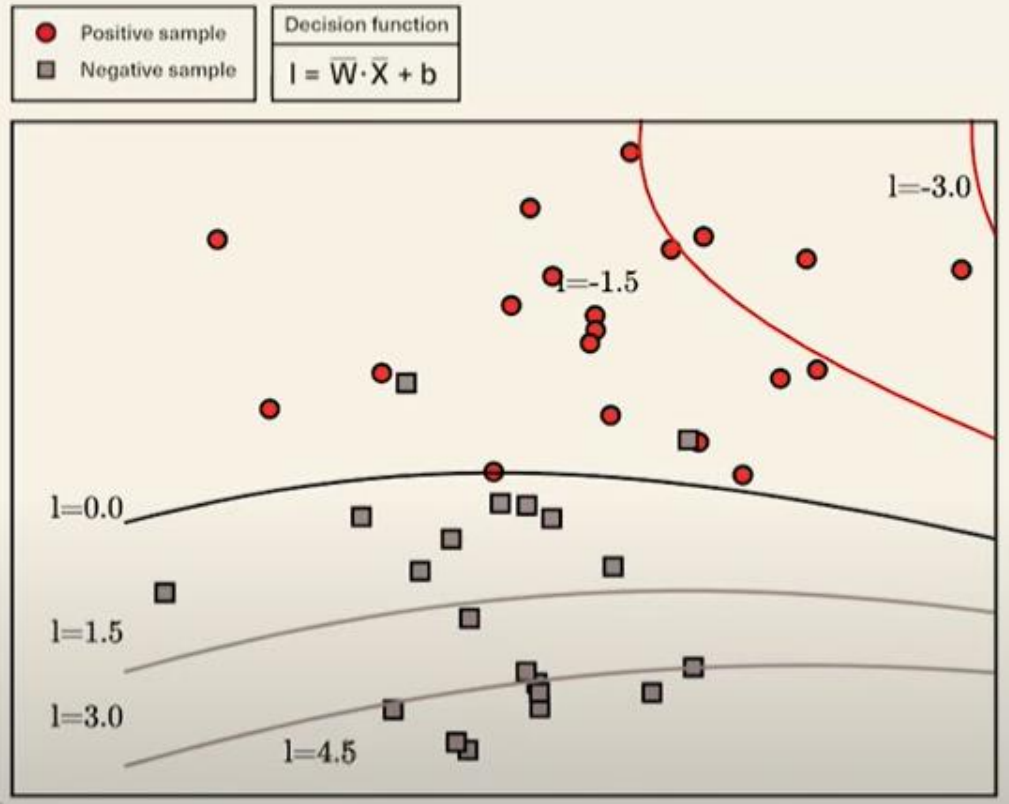
Kernels

- Why use kernels?
  - Make non-separable problem separable.
  - Map data into better representational space
- Common kernels
  - Linear
  - Polynomial $K(x,z) = (1+x^Tz)^d$
    - 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

Polynomial Kernel & RBF Kernel



Simulated with scikit learn

Degree of freedom

$$K(X_i, X_j) = (X_i \cdot X_j + c)^d$$

Constant

Polynomial Kernel

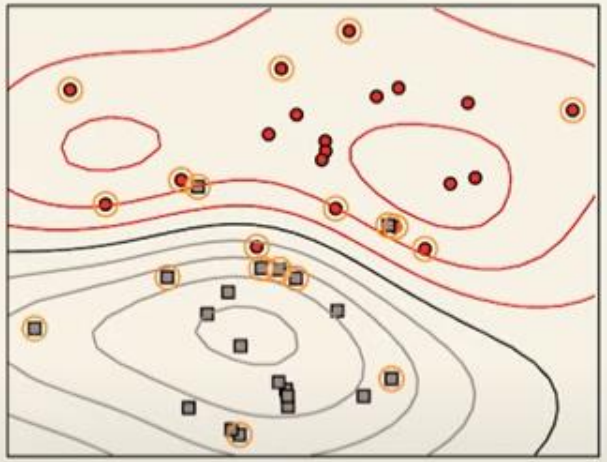
Polynomial Kernel & RBF Kernel

Positive sample
Negative sample
Support Vector

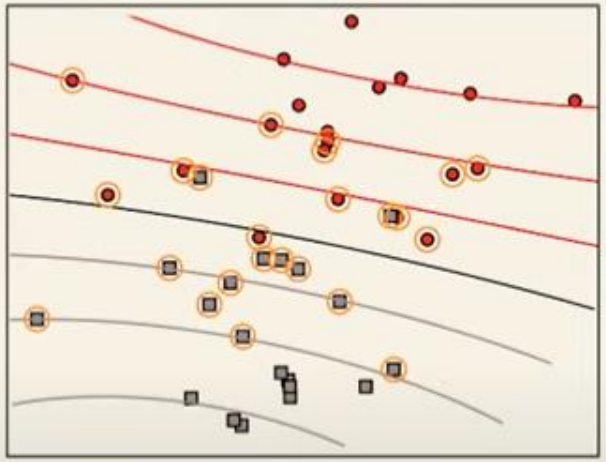
Decision function

$I = \overline{W} \cdot \overline{X} + b$

$\gamma = 1$



$\gamma = 0.01$



$\gamma = 0.005$

Gaussian Kernel

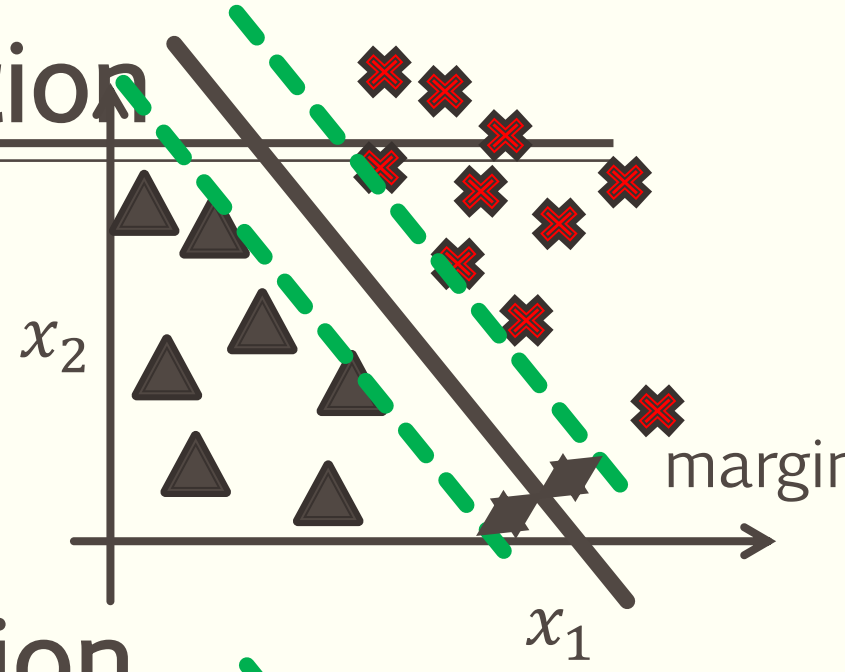
$$K(X_i, X_j) = \exp(-\gamma \| X_i - X_j \|^2)$$

RBF Kernel

Hard-margin SVM formulation

$$\min_{\theta} \frac{1}{2}\sum_{j=}^{n} \mathrm{W}_j^2$$

$$\text{s.t.} \quad \mathrm{W}^{\top} x^{(i)} \geq 1 \quad \text{if} \quad \mathrm{y}^{(i)} = 1$$
$$\mathrm{W}^{\top} x^{(i)} \leq -1 \quad \text{if} \quad \mathrm{y}^{(i)} = 0$$

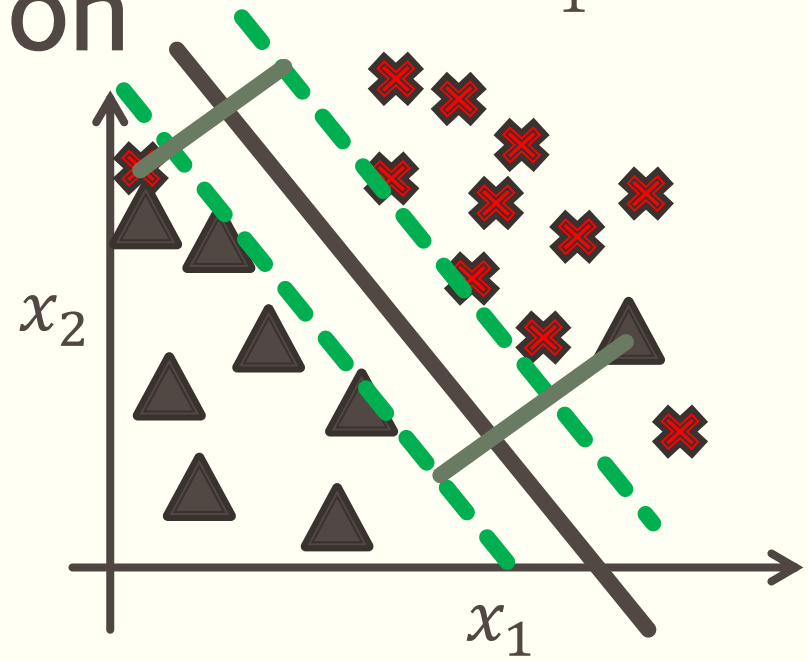


Soft-margin SVM formulation

$$\min_{\theta} \frac{1}{2}\overline{\mathrm{W}}\cdot\overline{\mathrm{W}} + \mathrm{C}\sum_{i=1}^{m}\zeta_i$$

$$\text{s.t.} \quad \mathrm{W}^{\top} x^{(i)} \geq 1 - \xi^{(i)} \quad \text{if} \quad \mathrm{y}^{(i)} = 1$$
$$\mathrm{W}^{\top} x^{(i)} \leq -1 + \xi^{(i)} \quad \text{if} \quad \mathrm{y}^{(i)} = 0$$

$$\xi^{(i)} \geq 0 \qquad \forall\, i$$



SVM parameters

- $C\left(=\frac{1}{\lambda}\right)$
  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

Slide credit: Andrew Ng