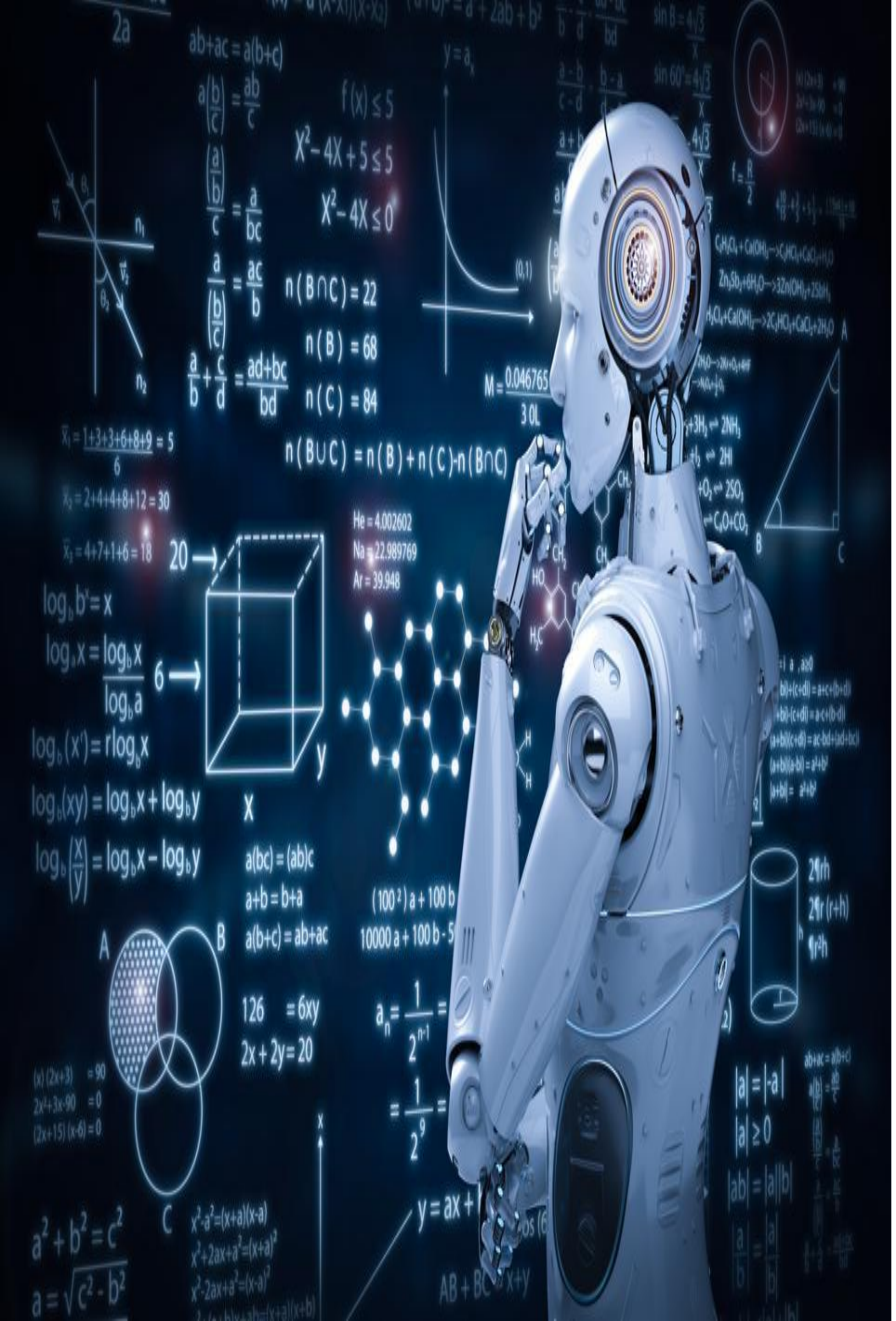


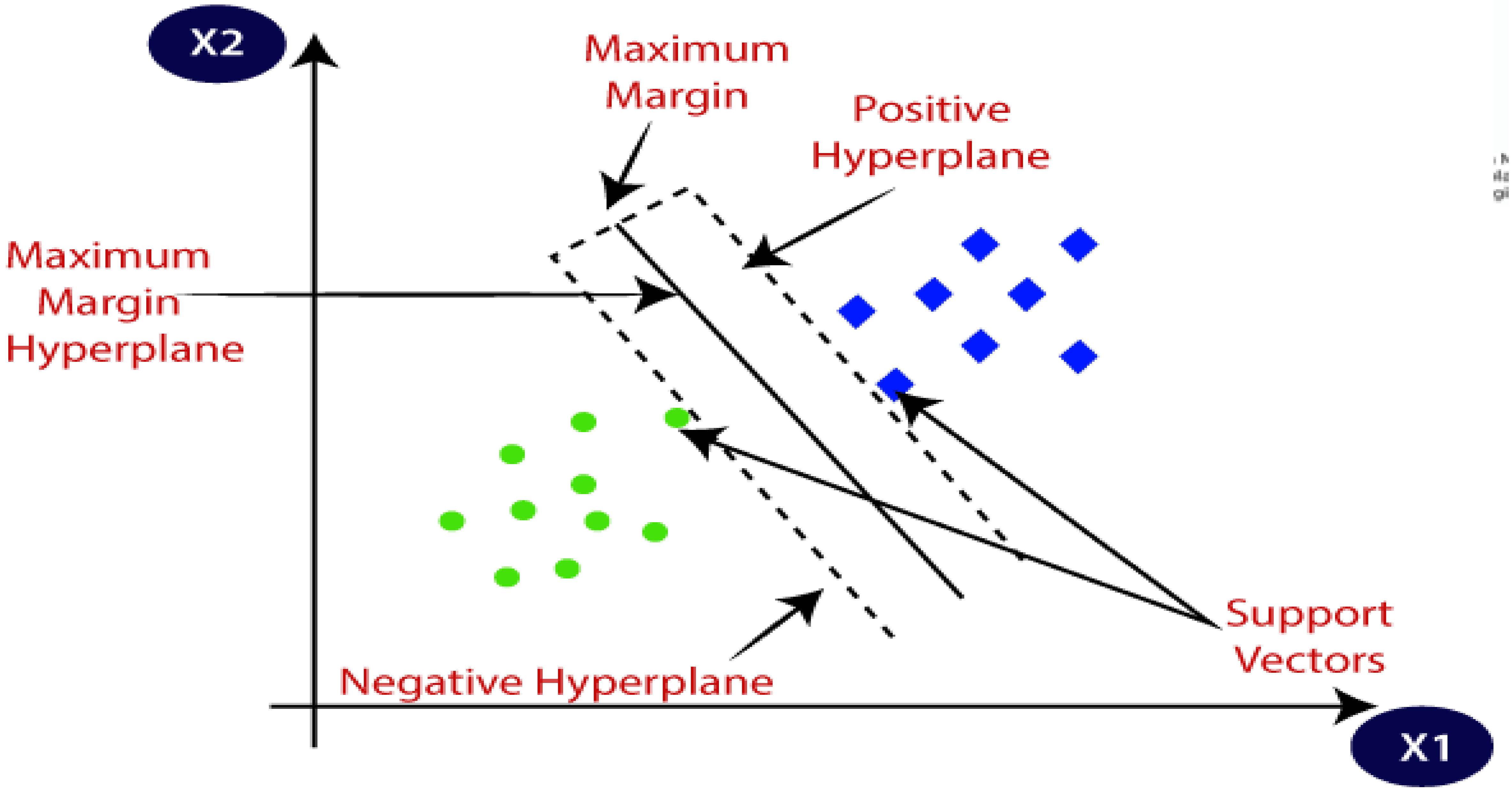
Practical Machine Learning

Day 9: Mar22 DBDA

Kiran Waghmare

Agenda

- SVM
- SVM-Kernel


X2
Maximum
Margin
Positive
Hyperplane
Maximum
Margin
Hyperplane
Support
Vectors
Negative Hyperplane
X1

Support Vector Machine

- A new classification method for both linear and nonlinear data
- It uses a nonlinear mapping to transform the original training data into a higher dimension
- With the new dimension, it searches for the linear optimal separating hyperplane (i.e., “decision boundary”)
- With an appropriate nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated by a hyperplane
- SVM finds this hyperplane using support vectors (“essential” training tuples) and margins (defined by the support vectors)

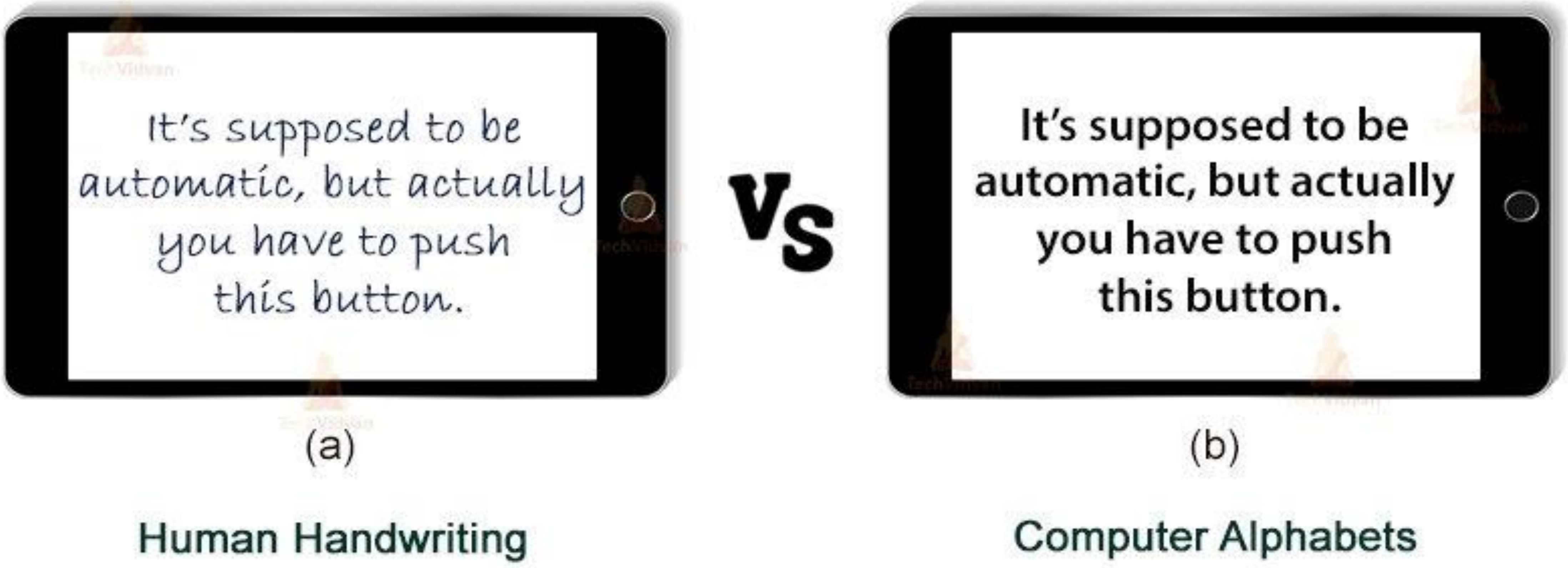
Support Vector Machine Algorithm

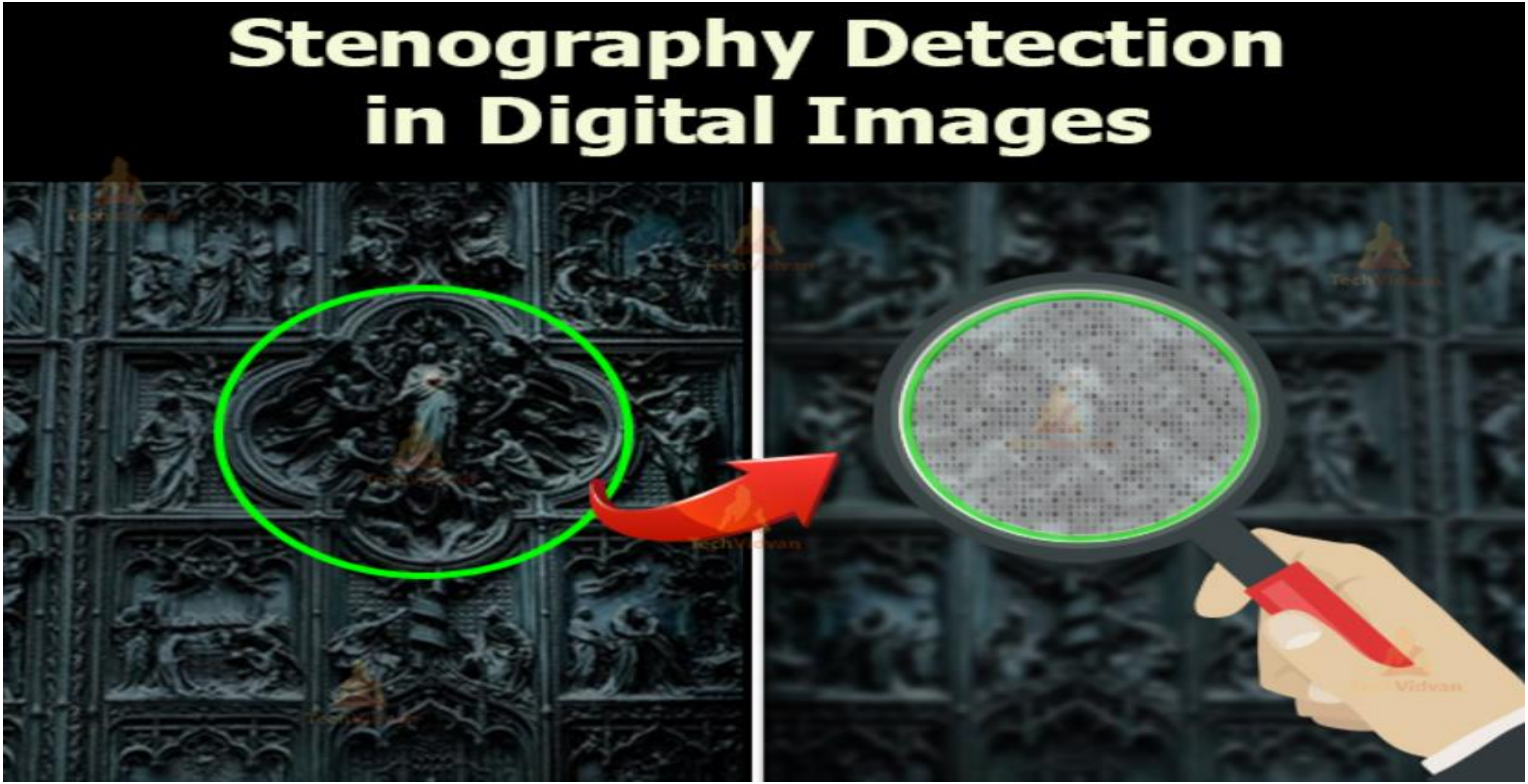
- **Goal :**
  - The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.
  - This best decision boundary is called **a hyperplane**
  - SVM chooses the extreme points/vectors that help in creating the hyperplane
  - These extreme cases are called as **support vectors**

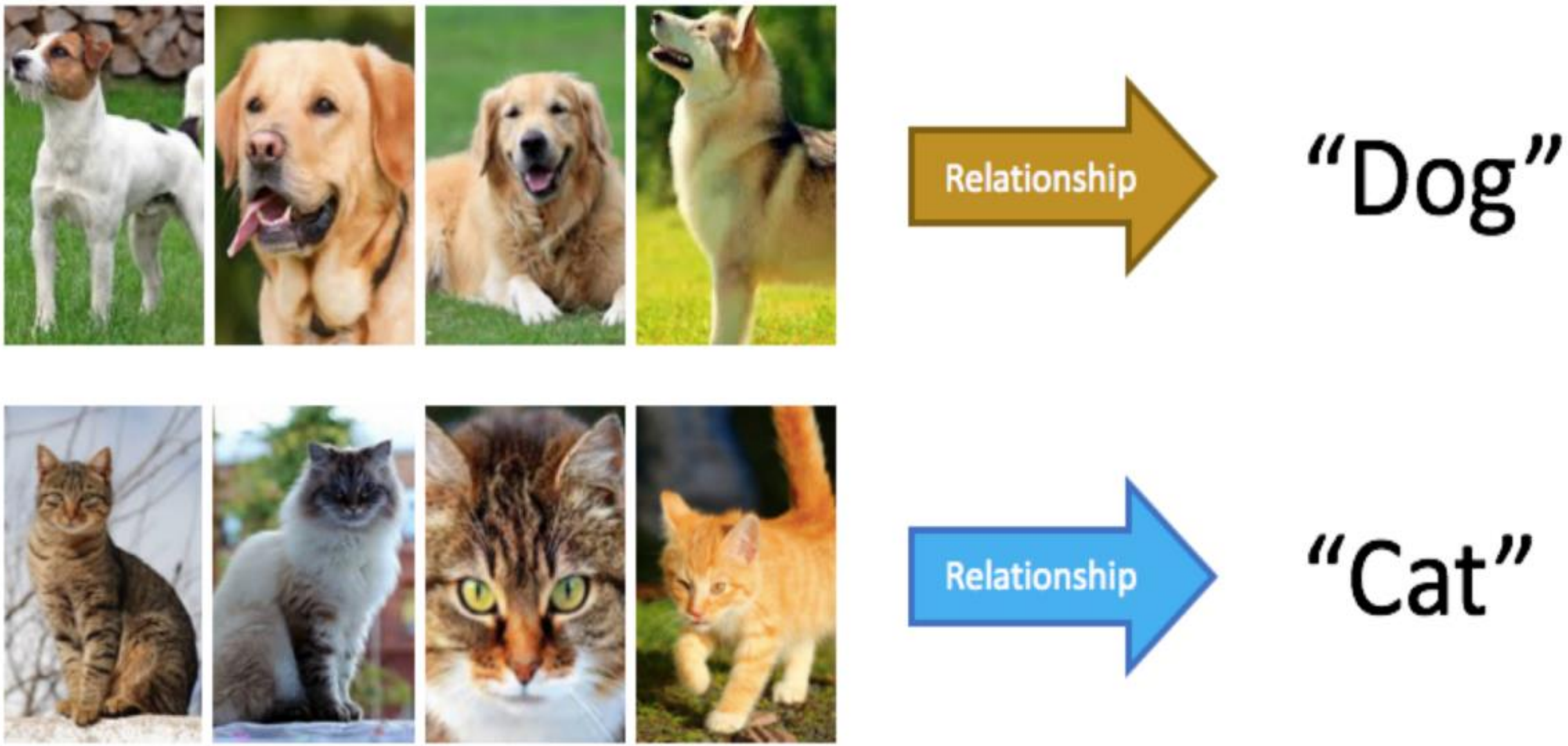
  and hence algorithm is termed as Support Vector Machine

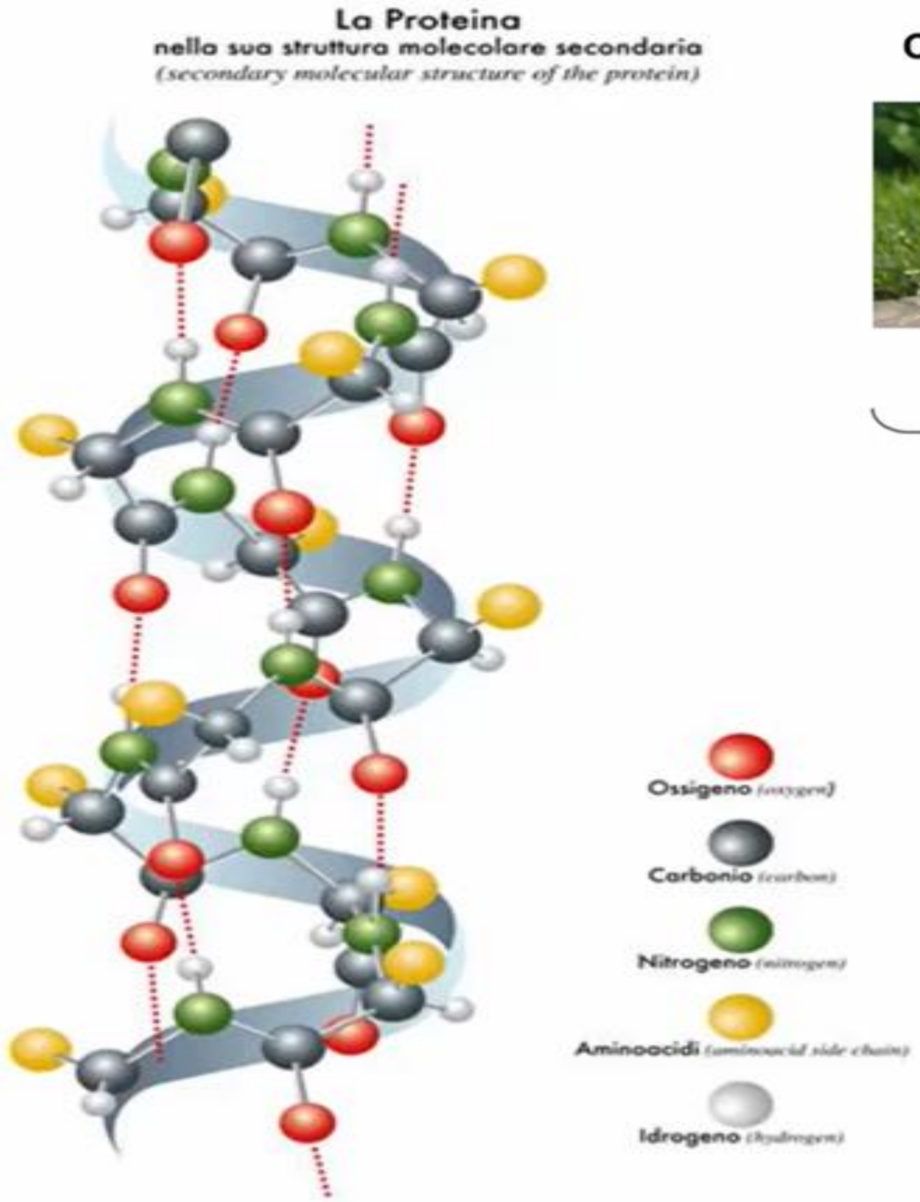
- . Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

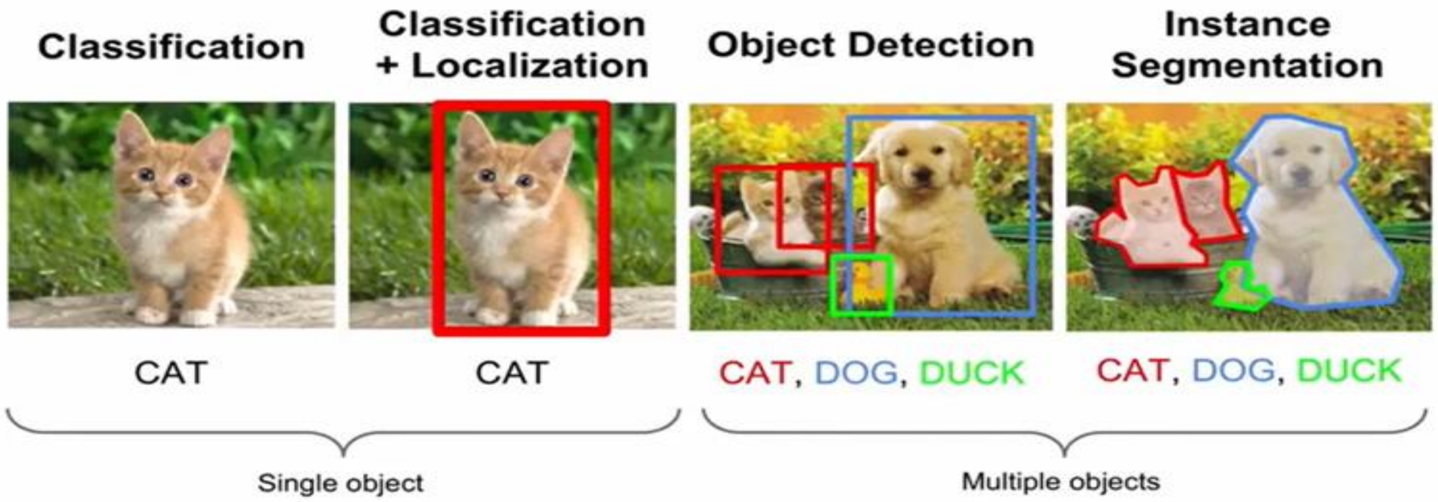
Text Classification using SVM

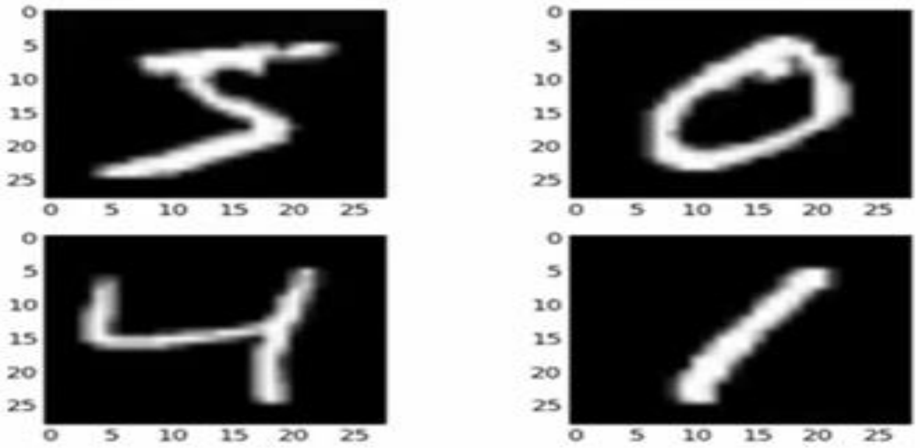



Stenography Detection
in Digital Images


Relationship
“Dog”
Relationship
“Cat”


La Proteina
nella sua struttura molecolare secondaria
(secondary molecular structure of the protein)
Ossigeno
Carbonio
Nitrogeno
Aminoacidi
Idrogeno


Classification
Classification + Localization
Object Detection
Instance Segmentation
CAT
CAT
CAT, DOG, DUCK
CAT, DOG, DUCK
Single object
Multiple objects

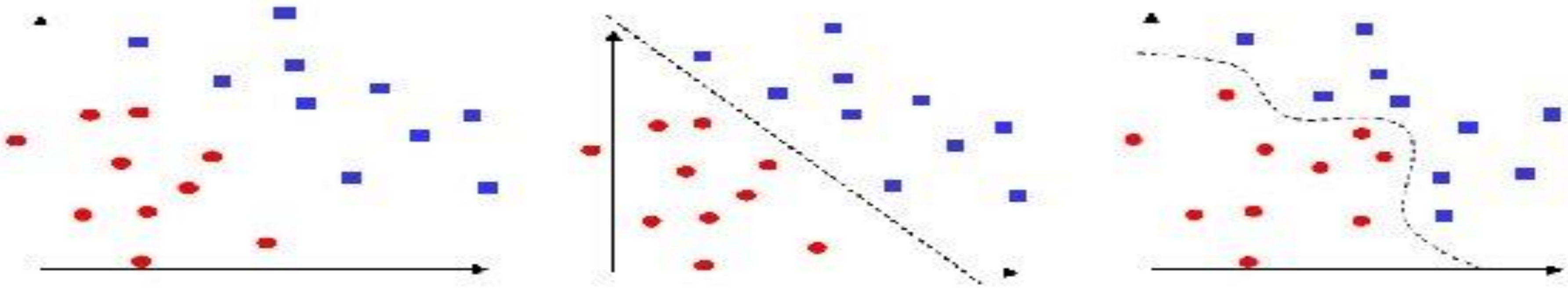


Support Vector Machine (SVM)

-- classifier, forward neural network, supervised learning

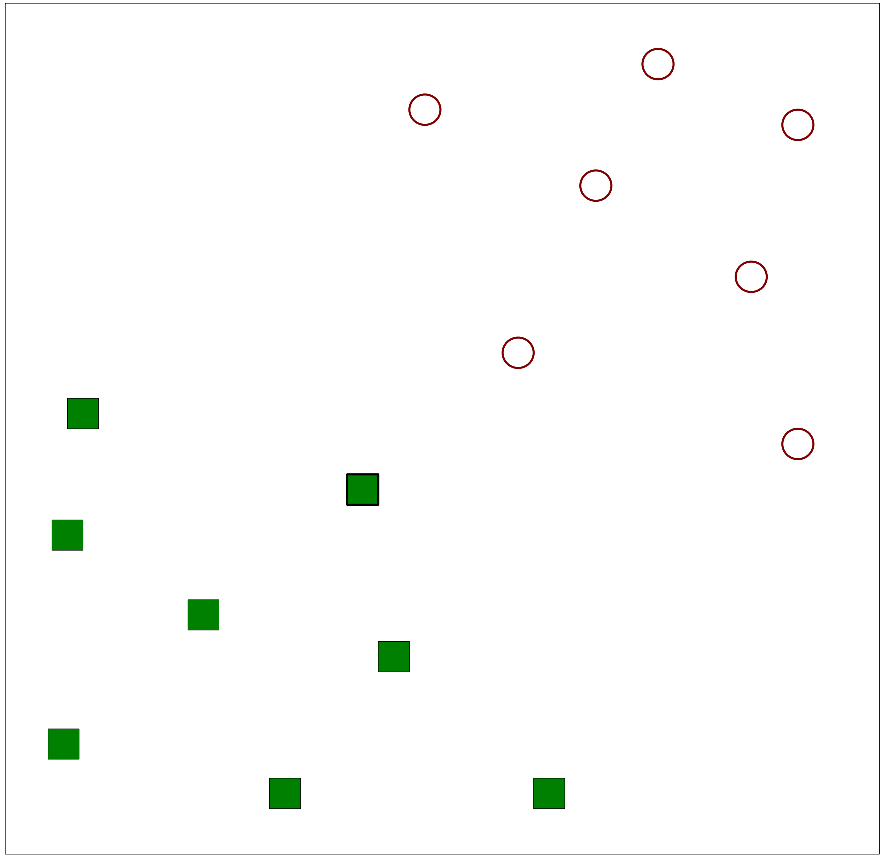
Difficulties with SVM:

i) binary classifier, ii) linearly separable patterns



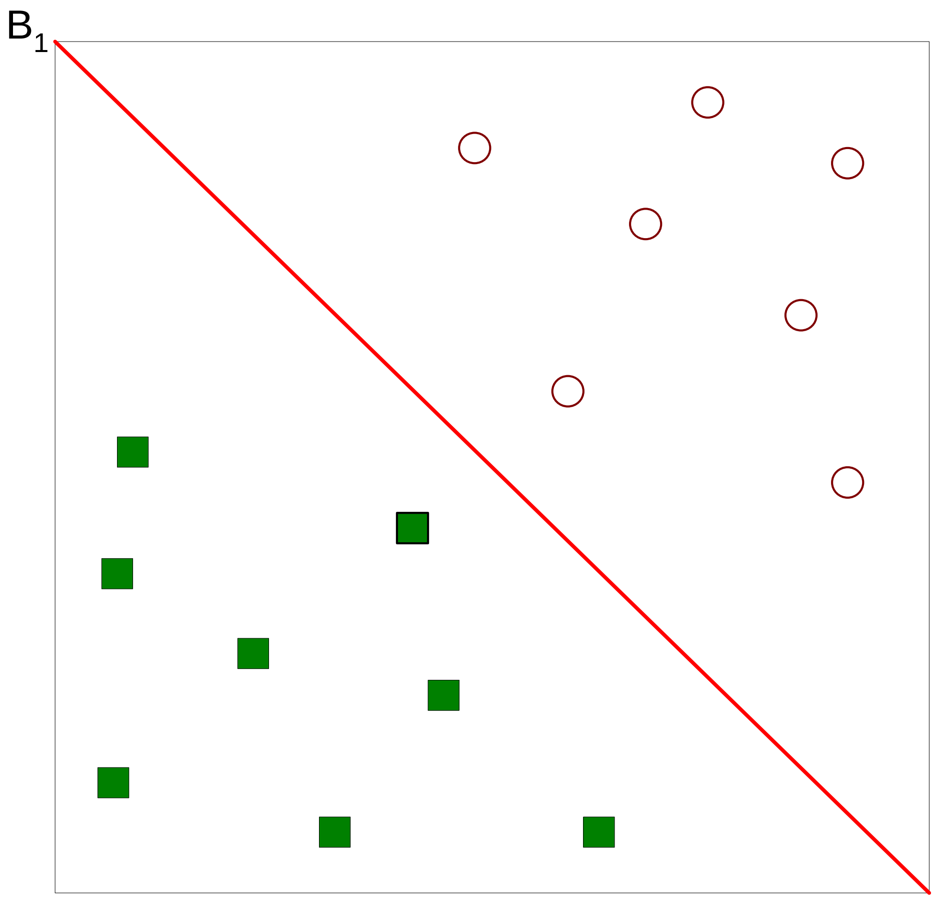
1

Support Vector Machines



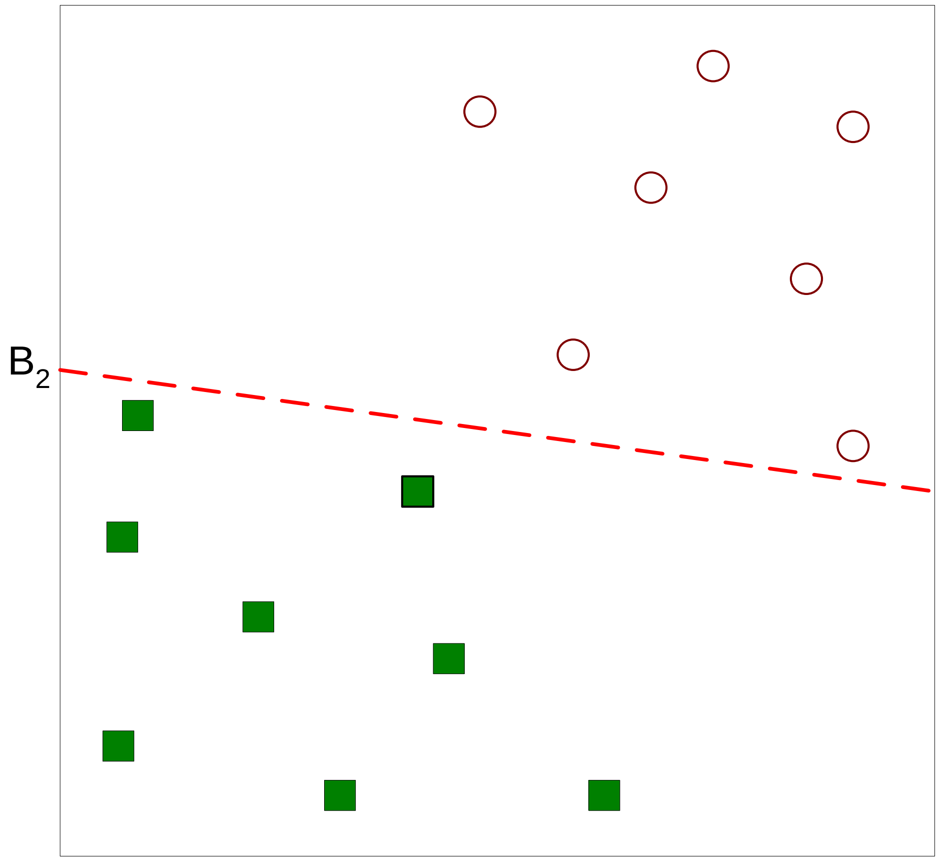
- Find a linear hyperplane (decision boundary) that will separate the data

Support Vector Machines



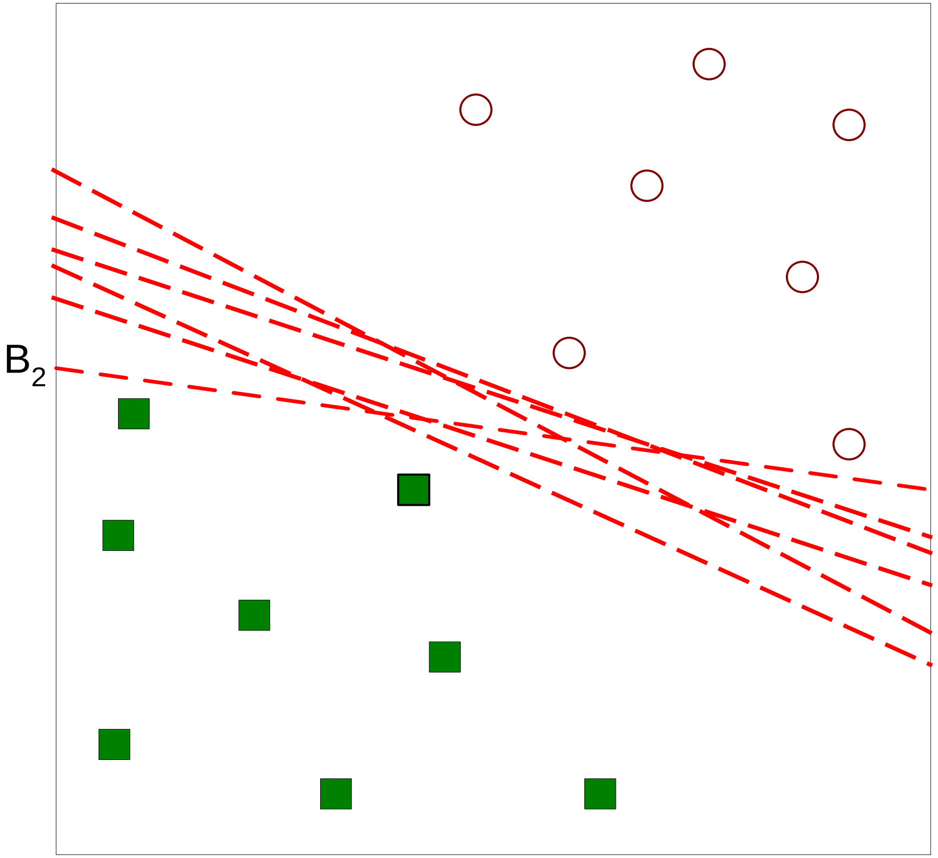
- One Possible Solution

Support Vector Machines



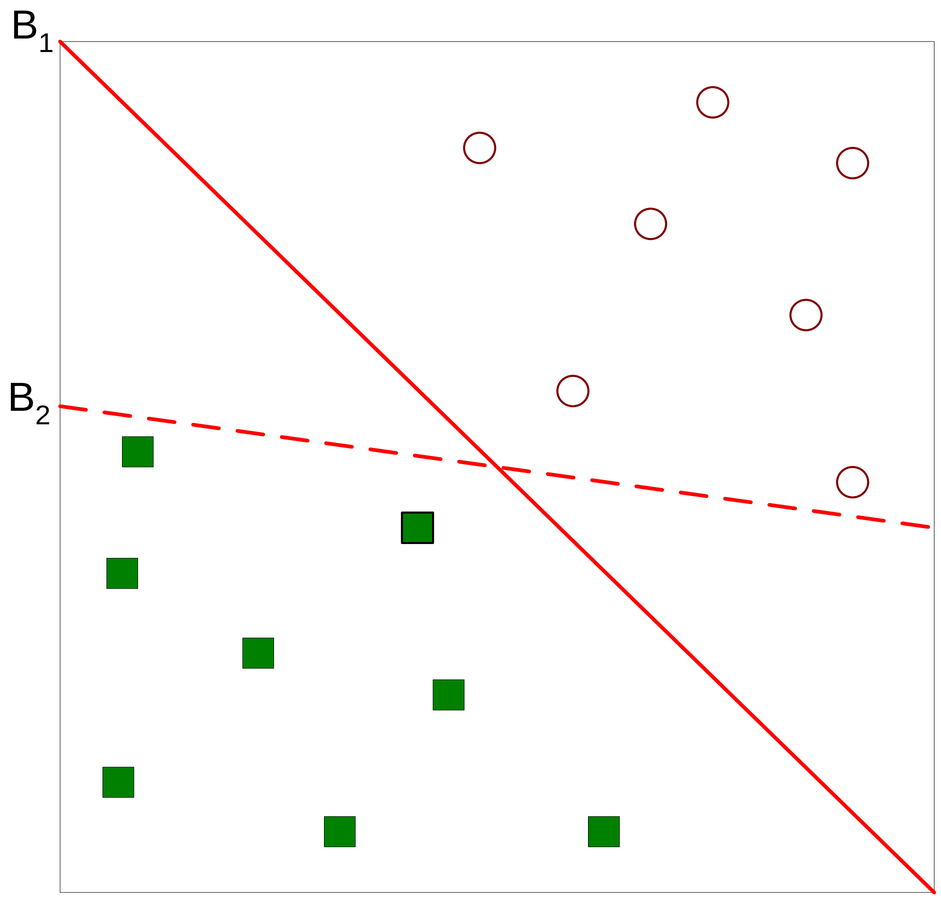
- Another possible solution

Support Vector Machines



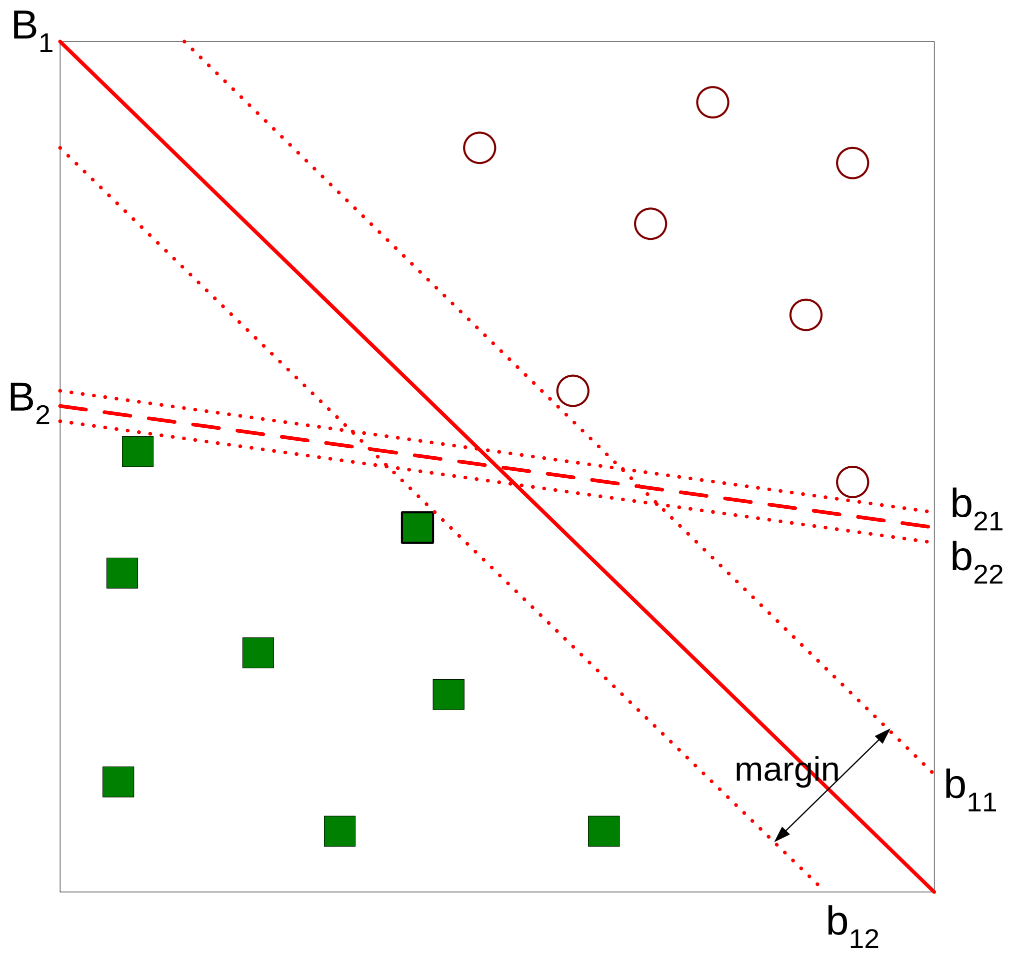
- Other possible solutions

Support Vector Machines

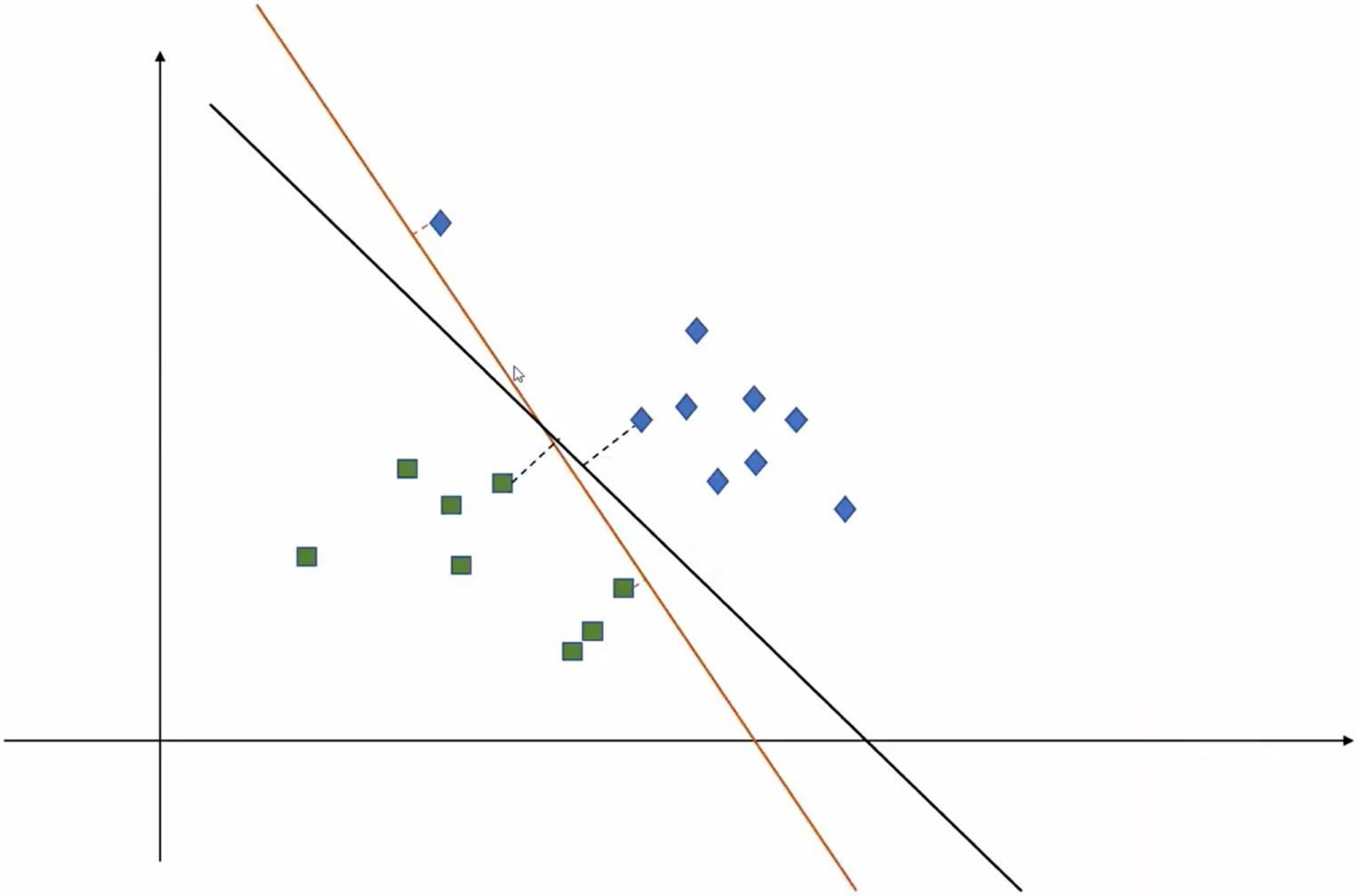


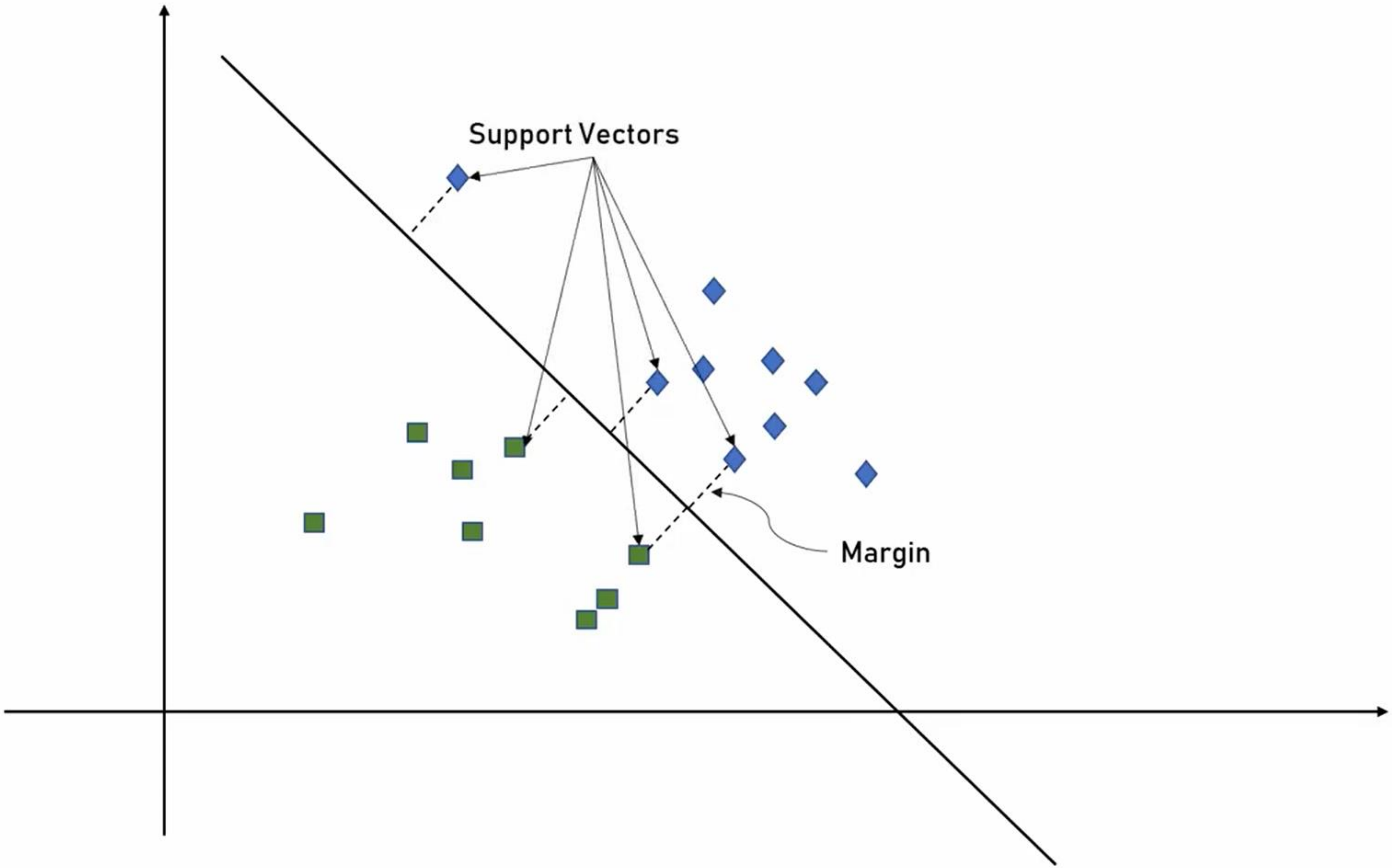
- Which one is better? B1 or B2?
- How do you define better?

Support Vector Machines



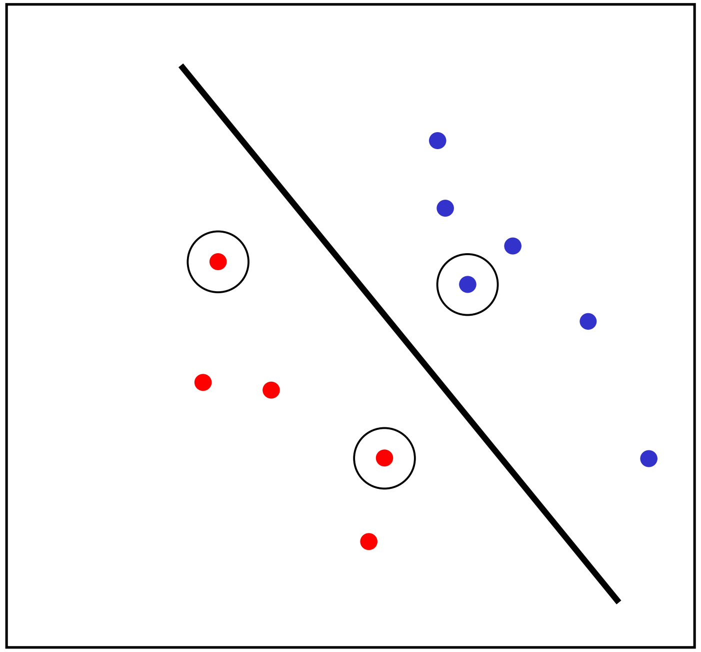
- Find hyperplane maximizes the margin => B1 is better than B2




Support Vectors
Margin

Support Vector Machines

- The line that maximizes the minimum margin is a good bet.
  - The model class of "hyper-planes with a margin of m" has a low VC dimension if m is big.
- This maximum-margin separator is determined by a subset of the datapoints.
  - Datapoints in this subset are called "support vectors".
  - It will be useful computationally if only a small fraction of the datapoints are support vectors, because we use the support vectors to decide which side of the separator a test case is on.

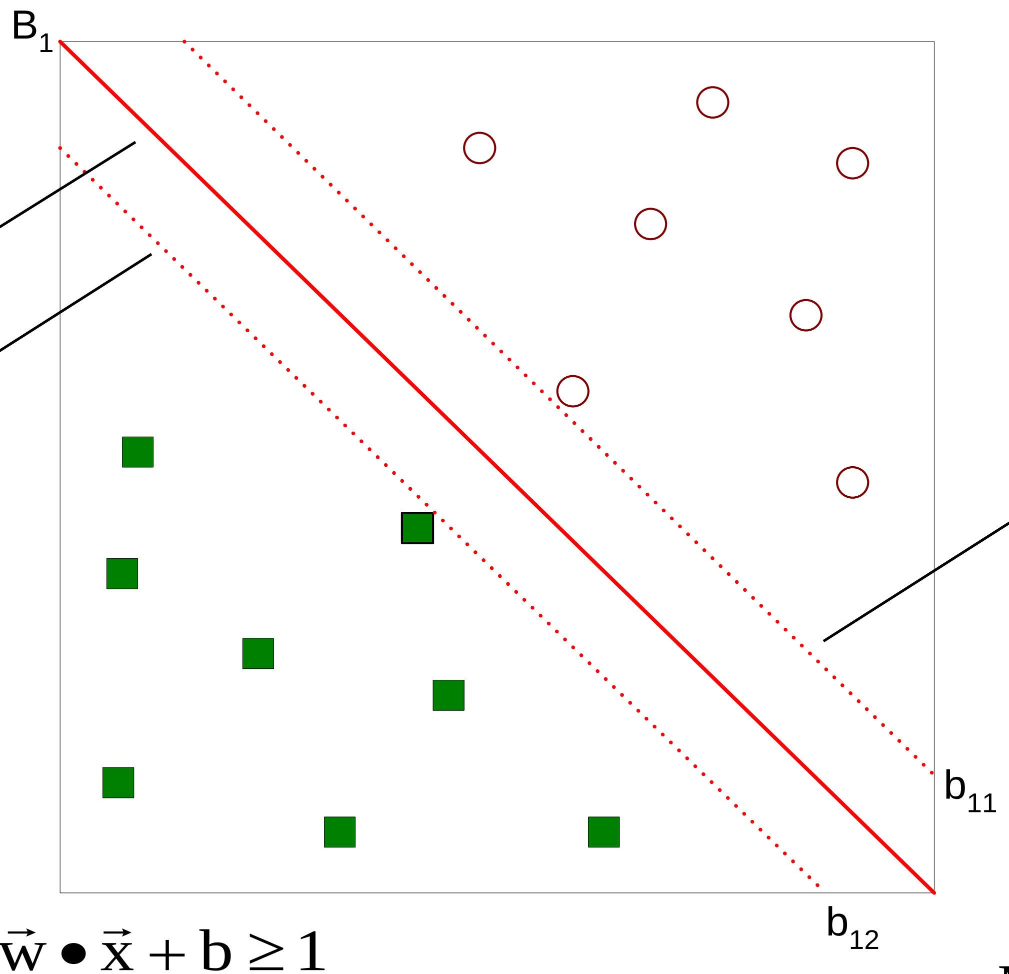


The support vectors are indicated by the circles around them.

Support Vector Machines

$$\vec{w}\bullet\vec{x}+b=0$$

$$\vec{w}\bullet\vec{x}+b=-1$$



$$\vec{w}\bullet\vec{x}+b=+1$$

$$f(\vec{x})=\begin{cases}1 & \text{if } \vec{\mathrm{w}}\bullet\vec{\mathrm{x}}+\mathrm{b}\geq 1\\ -1 & \text{if } \vec{\mathrm{w}}\bullet\vec{\mathrm{x}}+\mathrm{b}\leq -1\end{cases}$$

$$\text{Margin}=\frac{2}{\|\vec{w}\|}$$

Training a linear SVM

- To find the maximum margin separator, we have to solve the following optimization problem:

$$\mathbf{w}.\mathbf{x}^c + b > +1 \quad \textit{for positive cases}$$

$$\mathbf{w}.\mathbf{x}^c + b < -1 \quad \textit{for negative cases}$$

$$\textit{and} \quad \|\mathbf{w}\|^2 \quad \textit{is as small as possible}$$

- This is tricky but it's a convex problem. There is only one optimum and we can find it without fiddling with learning rates or weight decay or early stopping.
  - Don't worry about the optimization problem. It has been solved. Its called quadratic programming.
  - It takes time proportional to N^2 which is really bad for very big datasets
    - so for big datasets we end up doing approximate optimization!

Types of Kernel Functions

Some commonly used kernels

Polynomial: $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}.\mathbf{y}+1)^p$

Gaussian radial basis function $K(\mathbf{x},\mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2}$

Parameters that the user must choose

Neural net: $K(\mathbf{x},\mathbf{y}) = \tanh(k\,\mathbf{x}.\mathbf{y} - \delta)$

For the neural network kernel, there is one "hidden unit" per support vector, so the process of fitting the maximum margin hyperplane decides how many hidden units to use. Also, it may violate Mercer's condition.

Introducing slack variables

- Slack variables are constrained to be non-negative. When they are greater than zero they allow us to cheat by putting the plane closer to the datapoint than the margin. So we need to minimize the amount of cheating. This means we have to pick a value for lamba (this sounds familiar!)

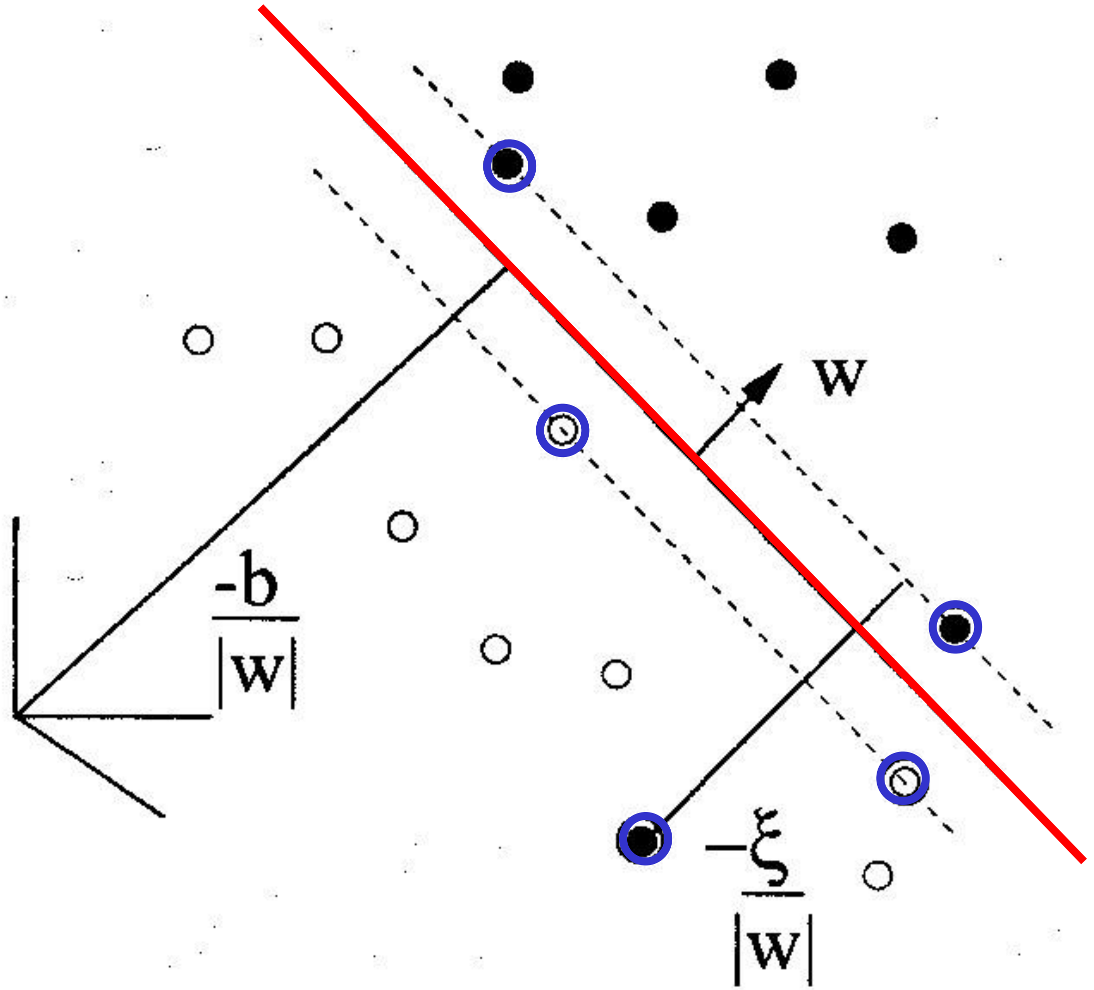
$$\mathbf{w}.\mathbf{x}^c + b \geq +1 - \xi^c \quad \textit{for positive cases}$$

$$\mathbf{w}.\mathbf{x}^c + b \leq -1 + \xi^c \quad \textit{for negative cases}$$

$$\textit{with} \quad \xi^c \geq 0 \quad \textit{for all } c$$

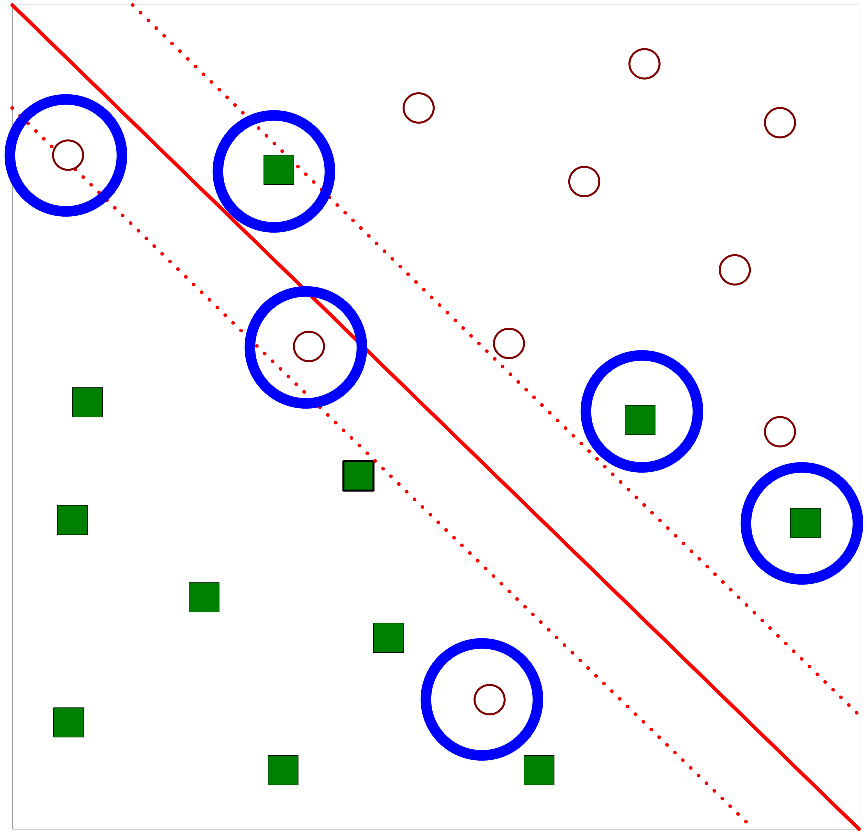
$$\textit{and} \quad \frac{\|\mathbf{w}\|^2}{2} + \lambda \sum_c \xi^c \quad \textit{as small as possible}$$

A picture of the best plane with a slack variable



Support Vector Machines

- What if the problem is not linearly separable?



Support Vector Machines

- What if the problem is not linearly separable?
  - Introduce slack variables
    - Need to minimize:

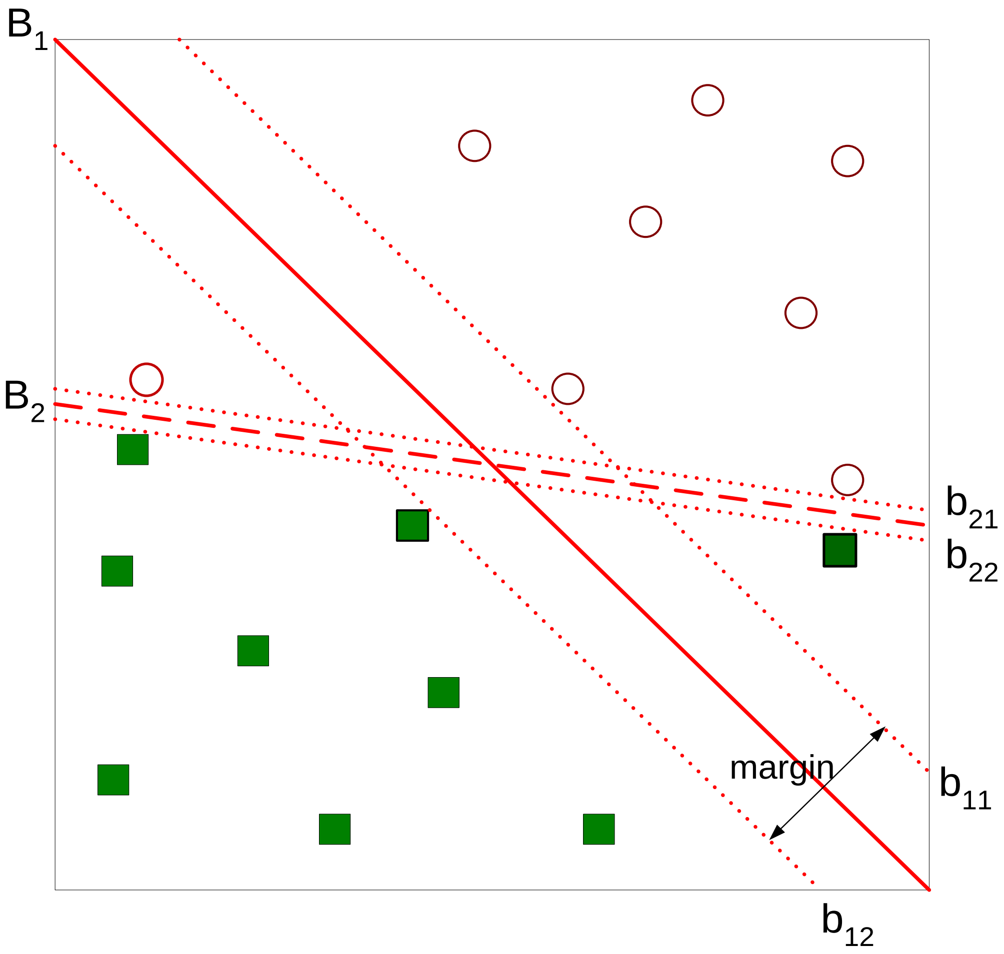
$$L(w) = \frac{\| \vec{w} \|^2}{2} + C\left(\sum_{i=1}^{N} \xi_i^k \right)$$

    - Subject to:

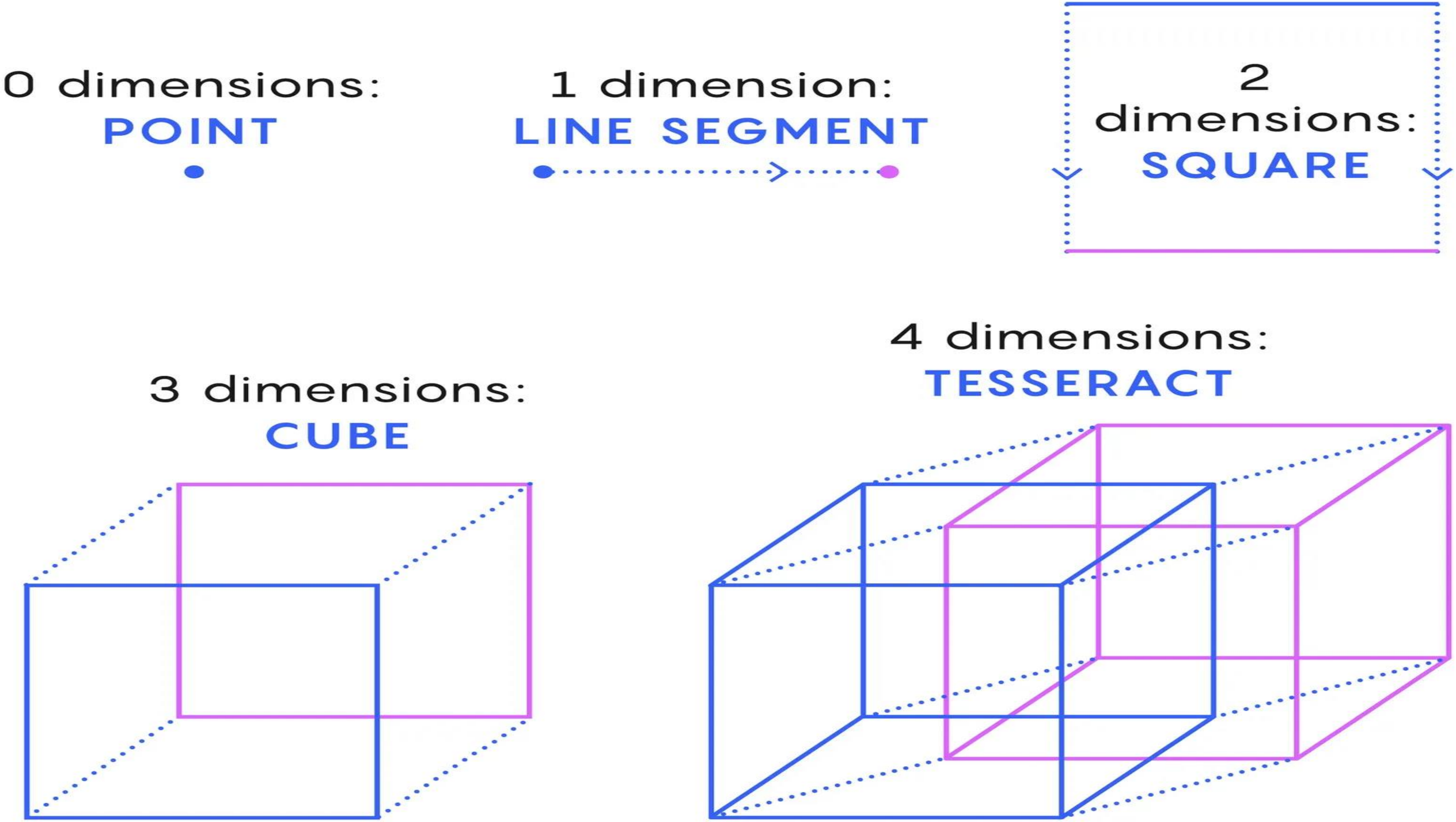
$$y_i = \begin{cases} 1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \geq 1 - \xi_\mathrm{i} \\ -1 & \text{if } \vec{\mathrm{w}} \bullet \vec{\mathrm{x}}_\mathrm{i} + \mathrm{b} \leq -1 + \xi_\mathrm{i} \end{cases}$$

    - If k is 1 or 2, this leads to similar objective function as linear SVM but with different constraints (see textbook)

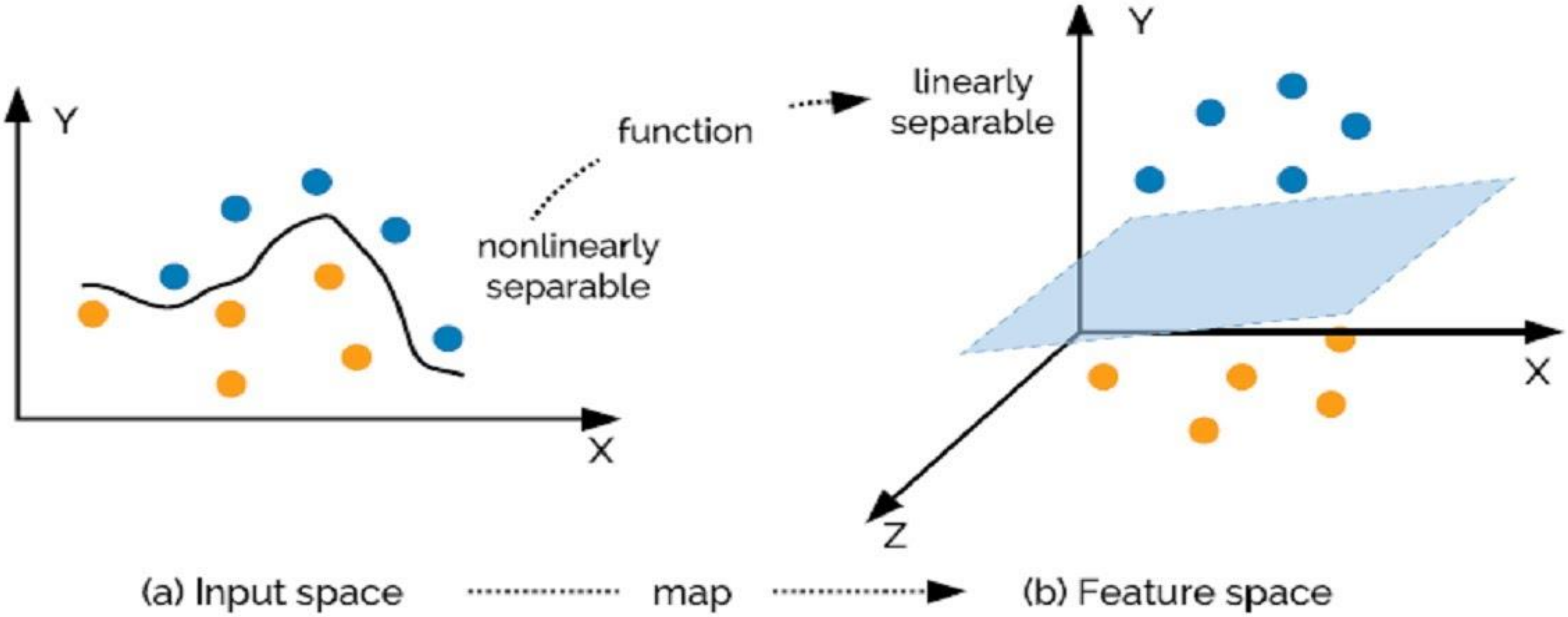
Support Vector Machines



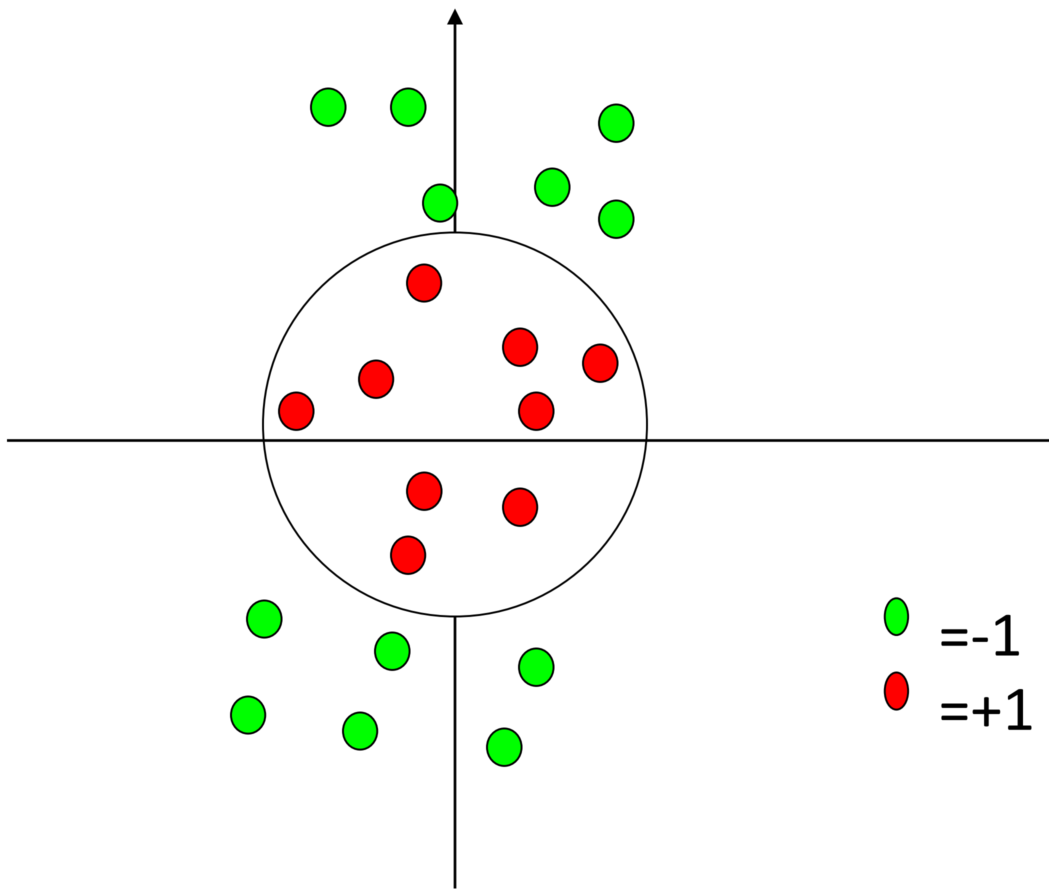
- Find the hyperplane that optimizes both factors


0 dimensions:
POINT
1 dimension:
LINE SEGMENT
2 dimensions:
SQUARE
3 dimensions:
CUBE
4 dimensions:
TESSERACT

Kernal Trick (SVM)…



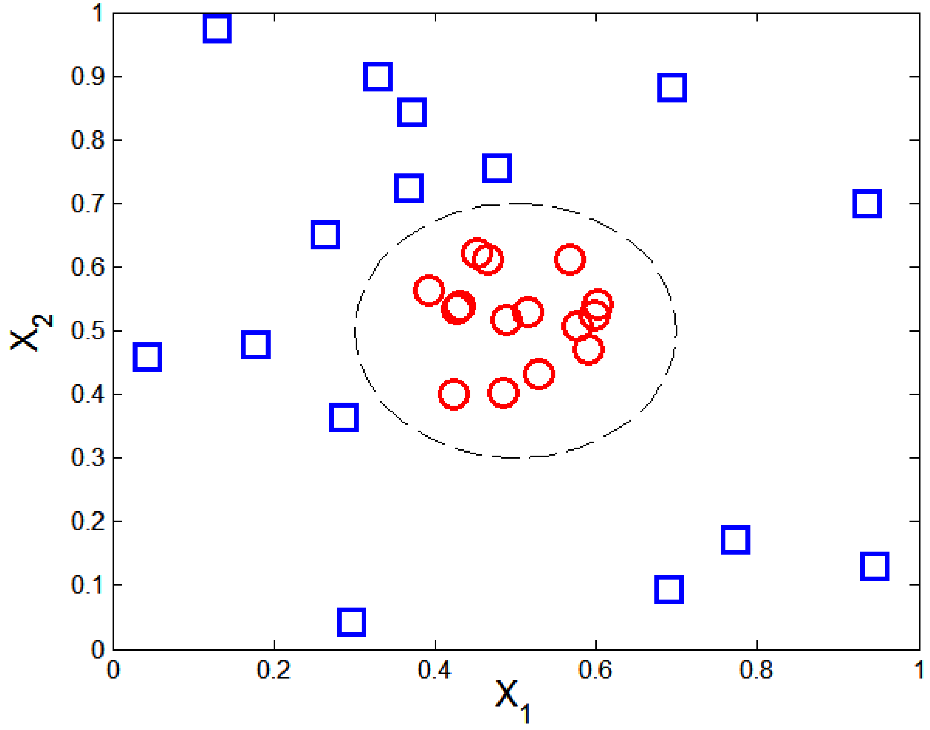
Problems with linear SVM



What if the decision function is not a linear?

Nonlinear Support Vector Machines

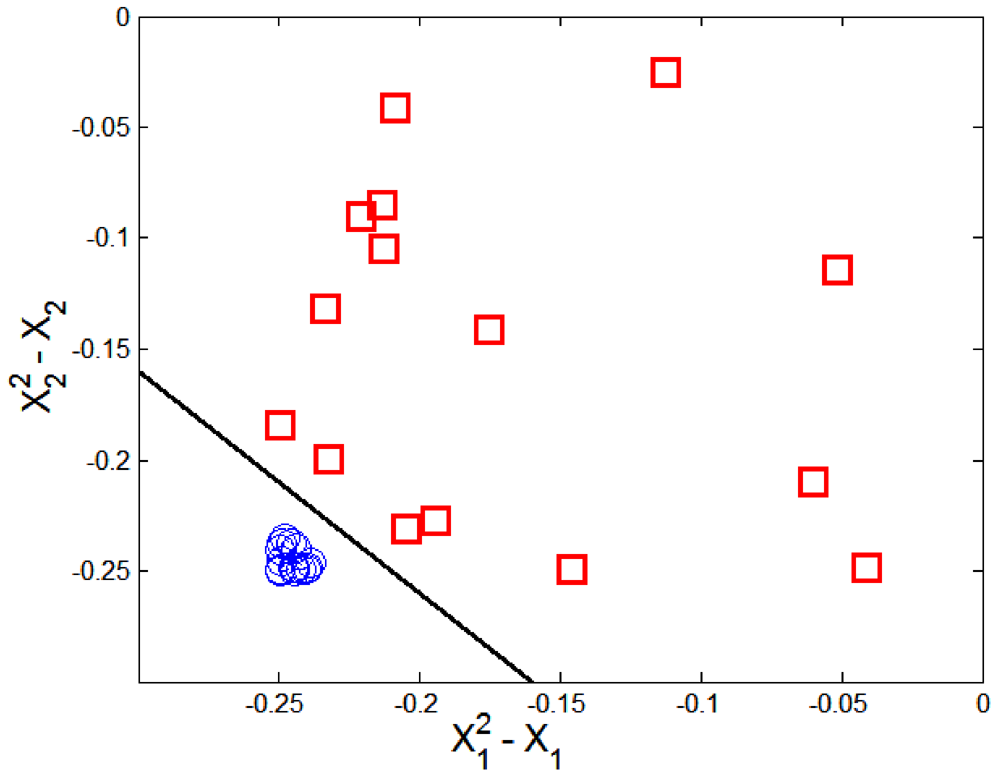
- What if decision boundary is not linear?



$$y(x_1, x_2) = \begin{cases} 1 & \text{if } \sqrt{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} > 0.2 \\ -1 & \text{otherwise} \end{cases}$$

Nonlinear Support Vector Machines

- Transform data into higher dimensional space



$$x_1^2 - x_1 + x_2^2 - x_2 = -0.46.$$

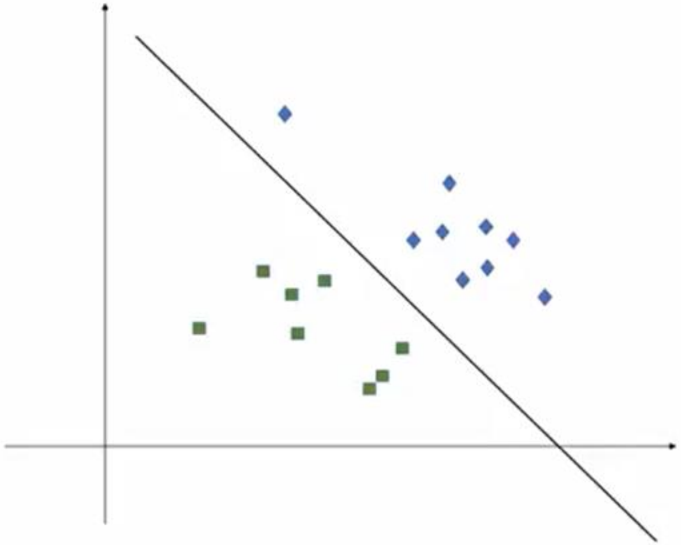
$$\Phi : (x_1, x_2) \longrightarrow (x_1^2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

$$w_4 x_1^2 + w_3 x_2^2 + w_2\sqrt{2}x_1 + w_1\sqrt{2}x_2 + w_0 = 0.$$

Decision boundary:

$$\vec{w} \bullet \Phi(\vec{x}) + b = 0$$

2D



3D

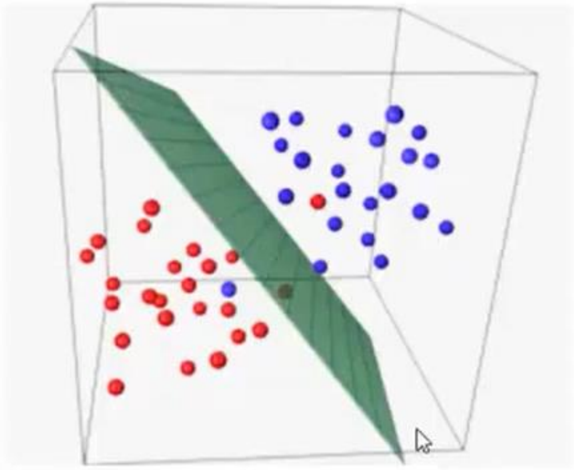
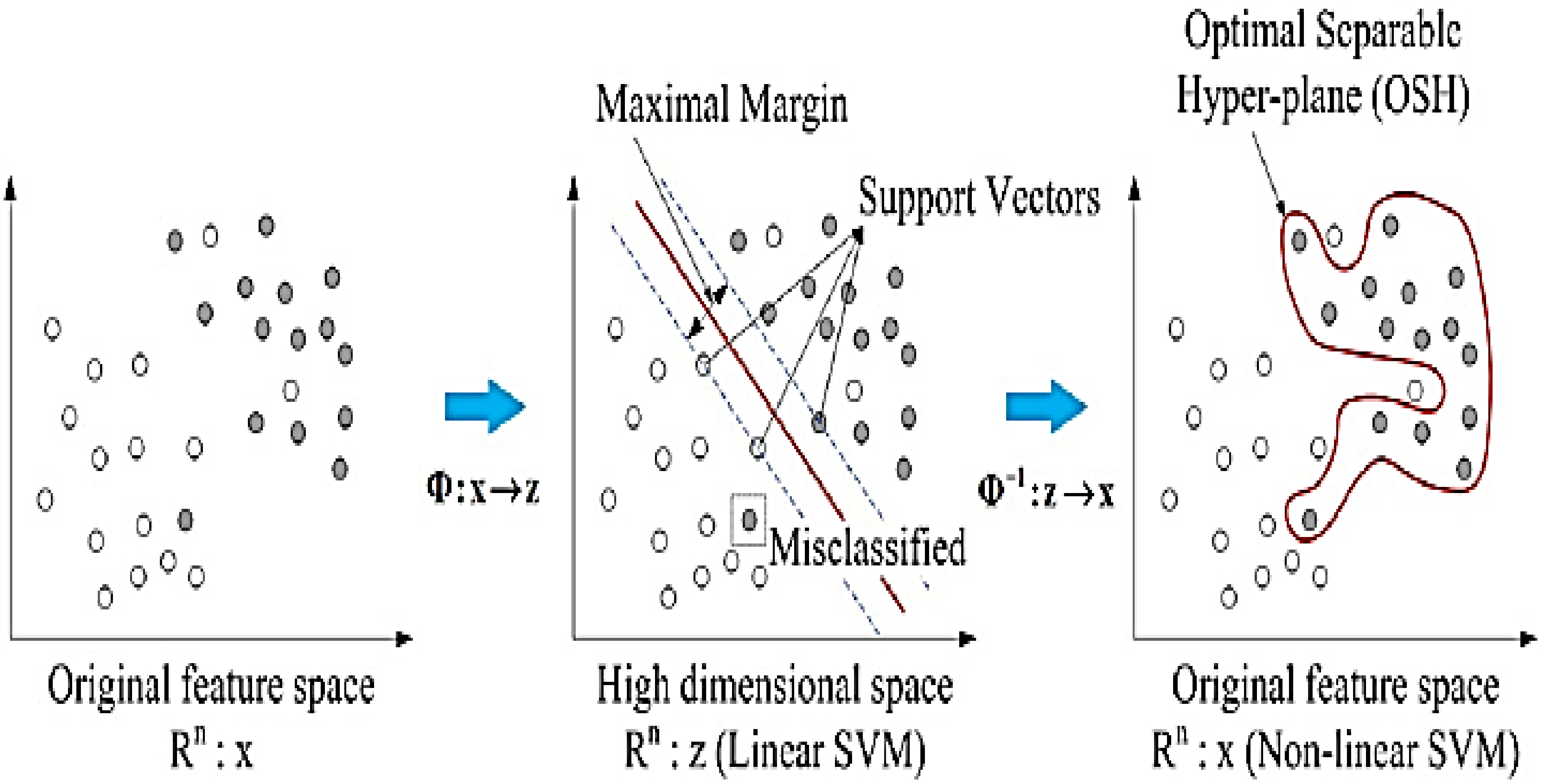
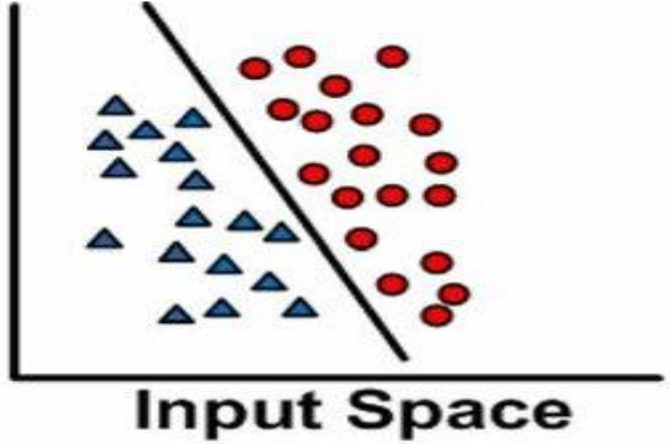
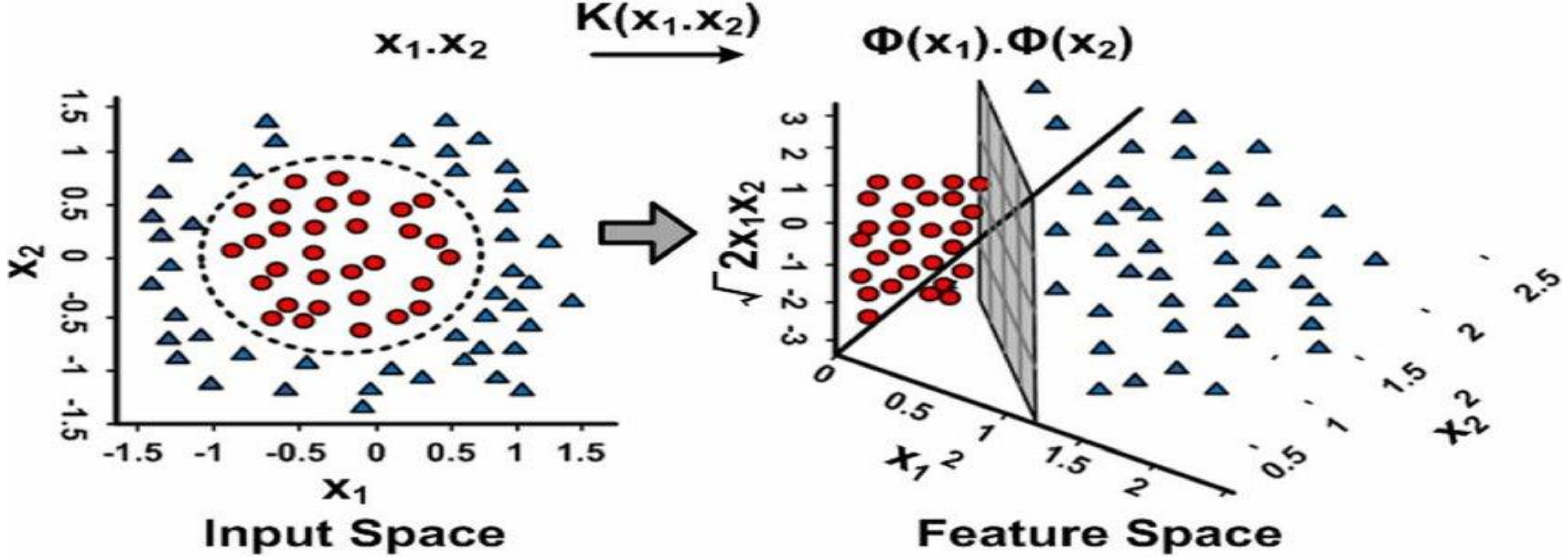


Image Credit: https://appliedmachinelearning.blog

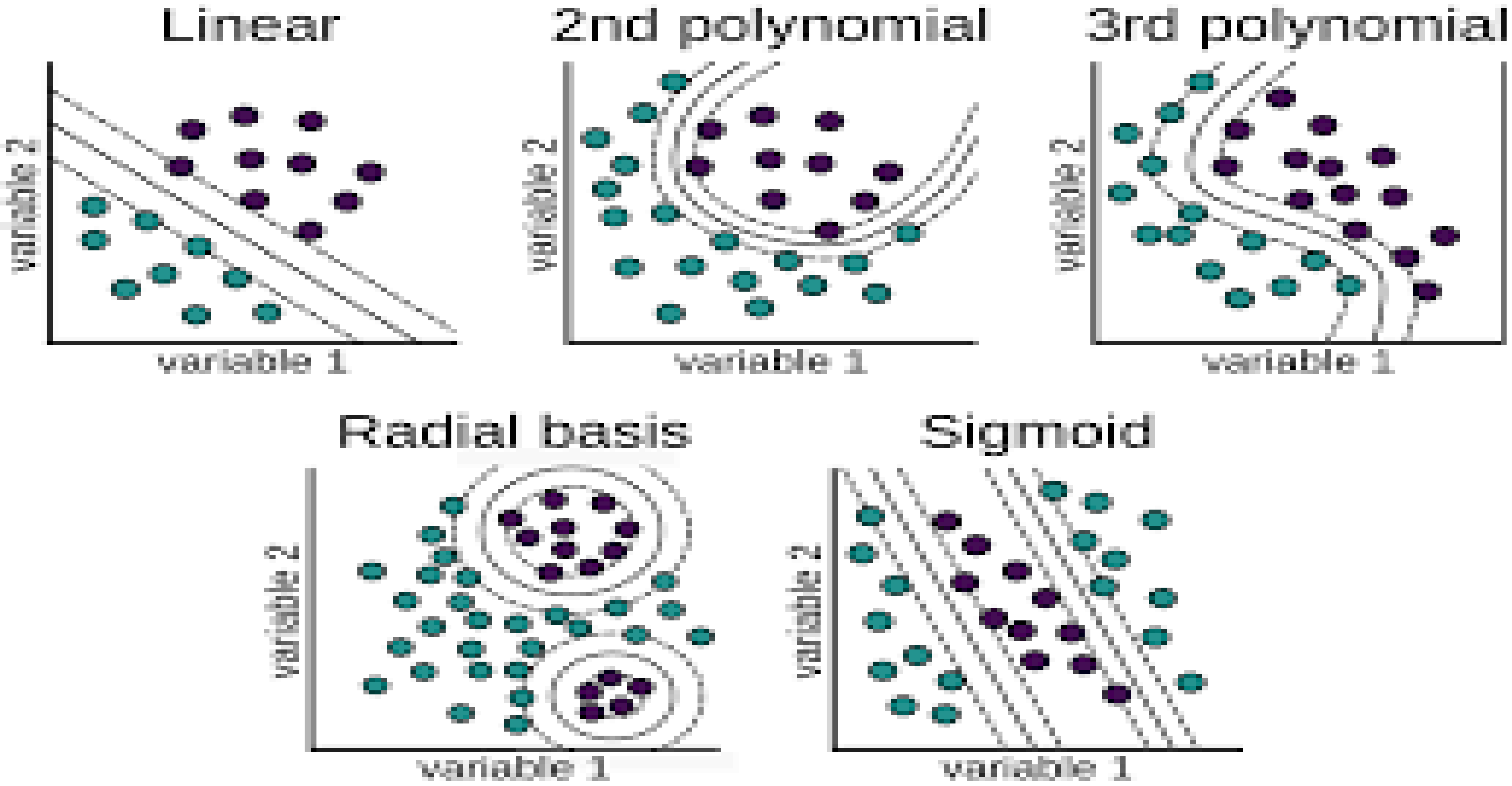

Optimal Separable
Hyper-plane (OSH)
Maximal Margin
Support Vectors
Misclassified
Φ:x→z
Φ^-1:z→x
Original feature space
R^n : x
High dimensional space
R^n : z (Linear SVM)
Original feature space
R^n : x (Non-linear SVM)


Input Space

(a)


K(x1.x2)
x1.x2
Φ(x1).Φ(x2)
x2
x1
Input Space
√2x1x2
x1 2
x2 2
Feature Space

(b)


Linear
2nd polynomial
3rd polynomial
Radial basis
Sigmoid
variable 1
variable 2