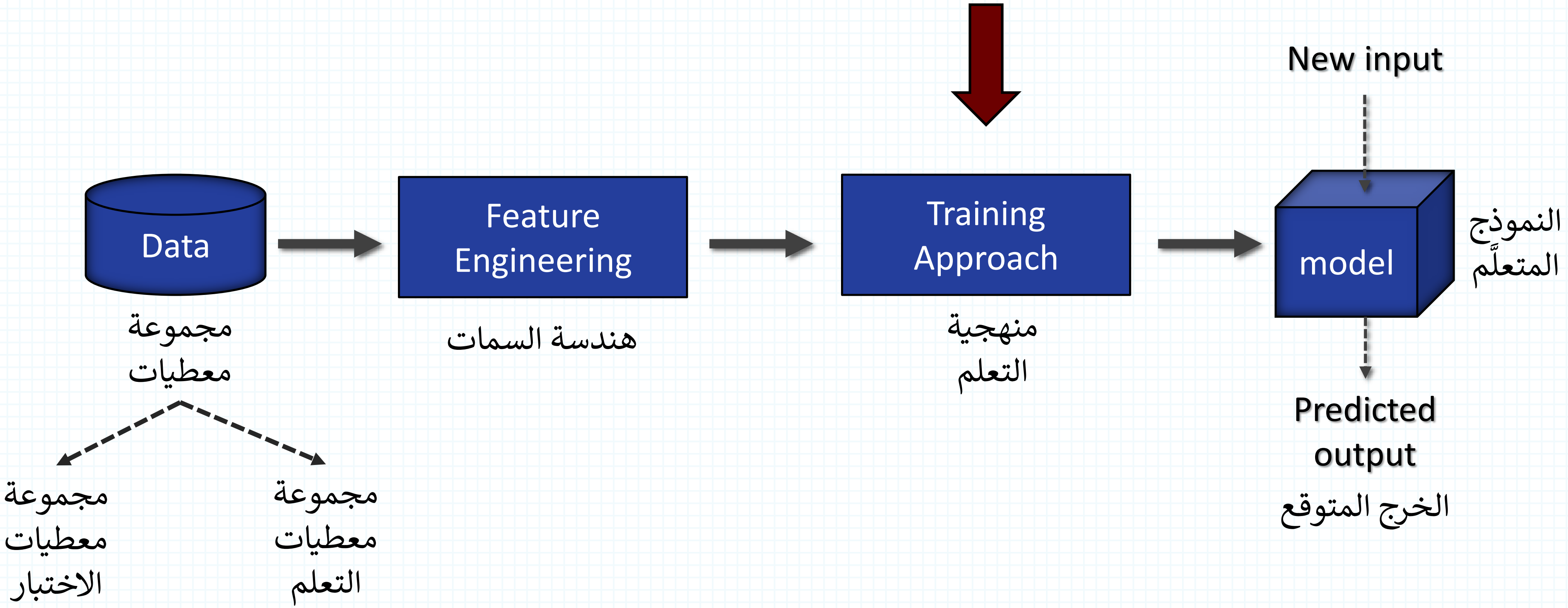
الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

مقرر تعلم الآلة | كلية الهندسة المعلوماتية | المحاضرة 6

Support Vector Machine (SVM) 1

د. رياض سنبل

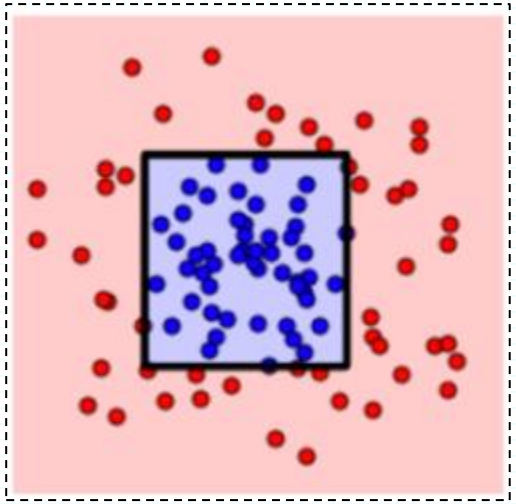
ML Pipeline


New input
Data
Feature Engineering
Training Approach
model
النموذج المتعلَّم
مجموعة معطيات
هندسة السمات
منهجية التعلم
Predicted output
الخرج المتوقع
مجموعة معطيات الاختبار
مجموعة معطيات التعلم

Why SVM? … Why not decision trees?

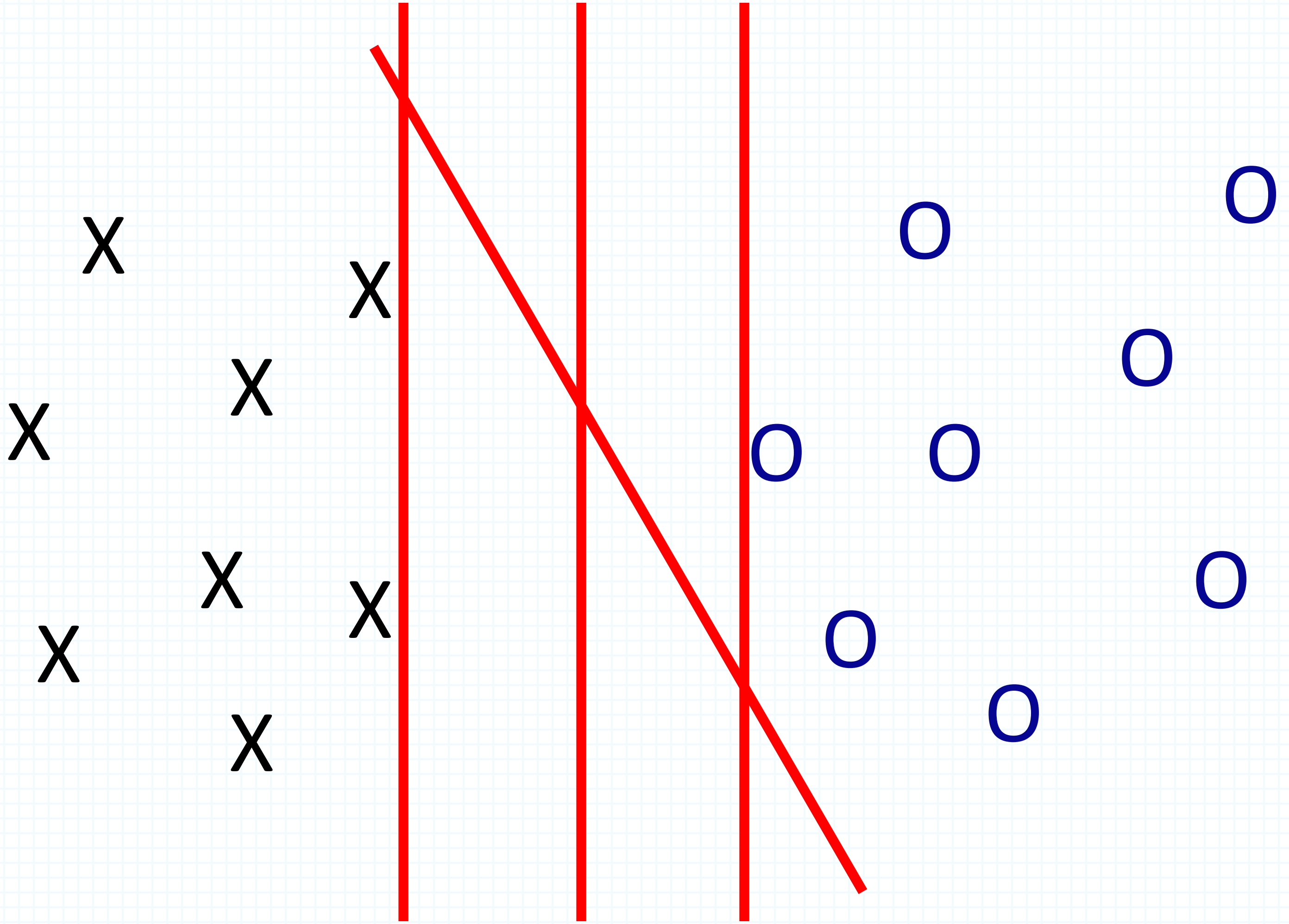
- Can Decision Trees detect non-linear models?
  - Yes, Decision trees can detect non-linear relationships

- What type of boundaries can be detected using decision trees in each step?
  - The decision boundary in a Decision Tree is linear and perpendicular to one of the input dimensions, which means that it is limited to finding only axis-parallel splits.

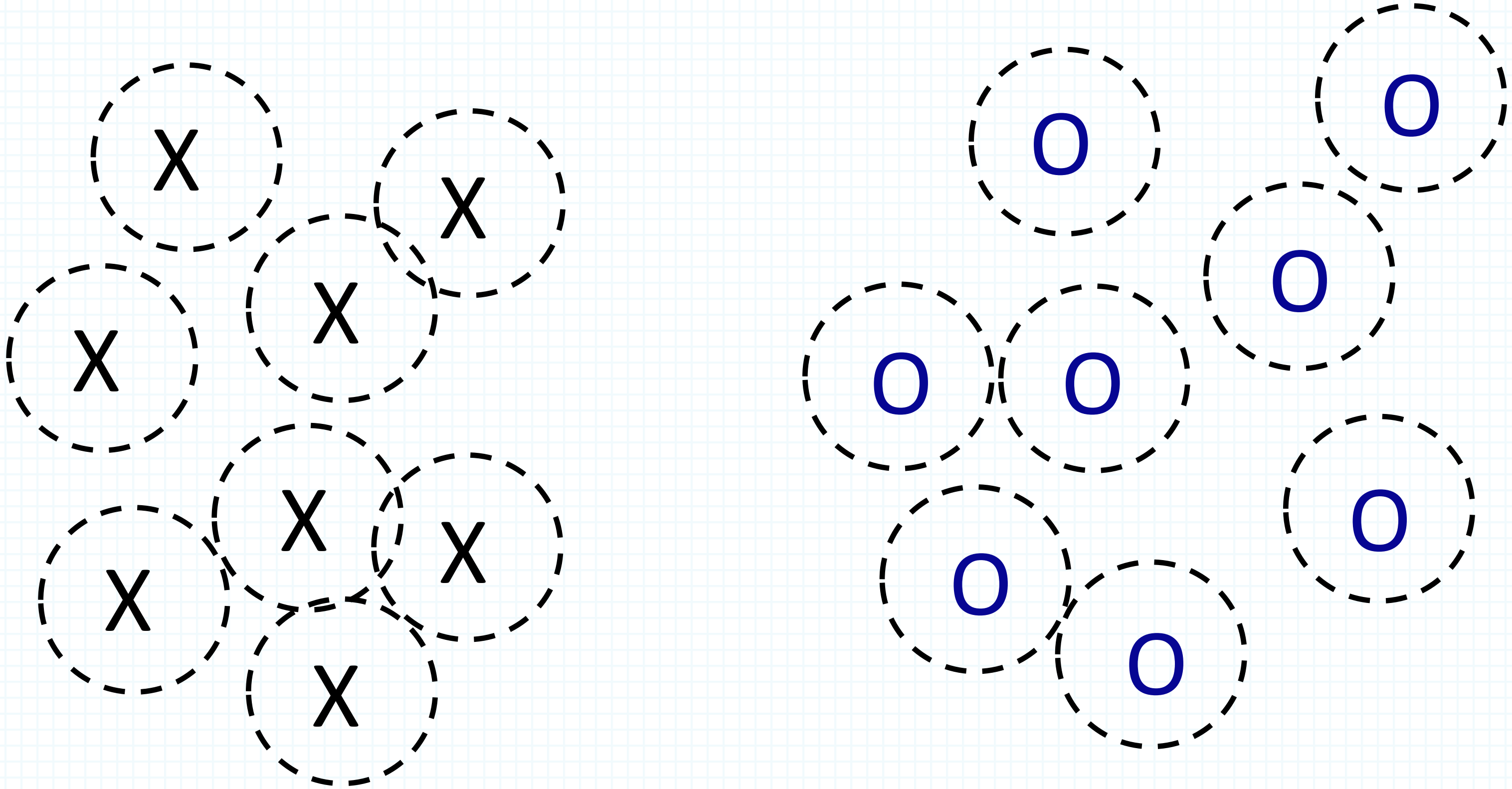


- What if we have higher-dimensional feature space, more complex relationships between input features and target class?
  - In the higher-dimensional feature space, the decision boundary can take on a more complex shape, such as a curved or nonlinear boundary.
  - More problems when the relationship between the input features and the target variable is complex (ex: image classification, sentiment analysis, etc)

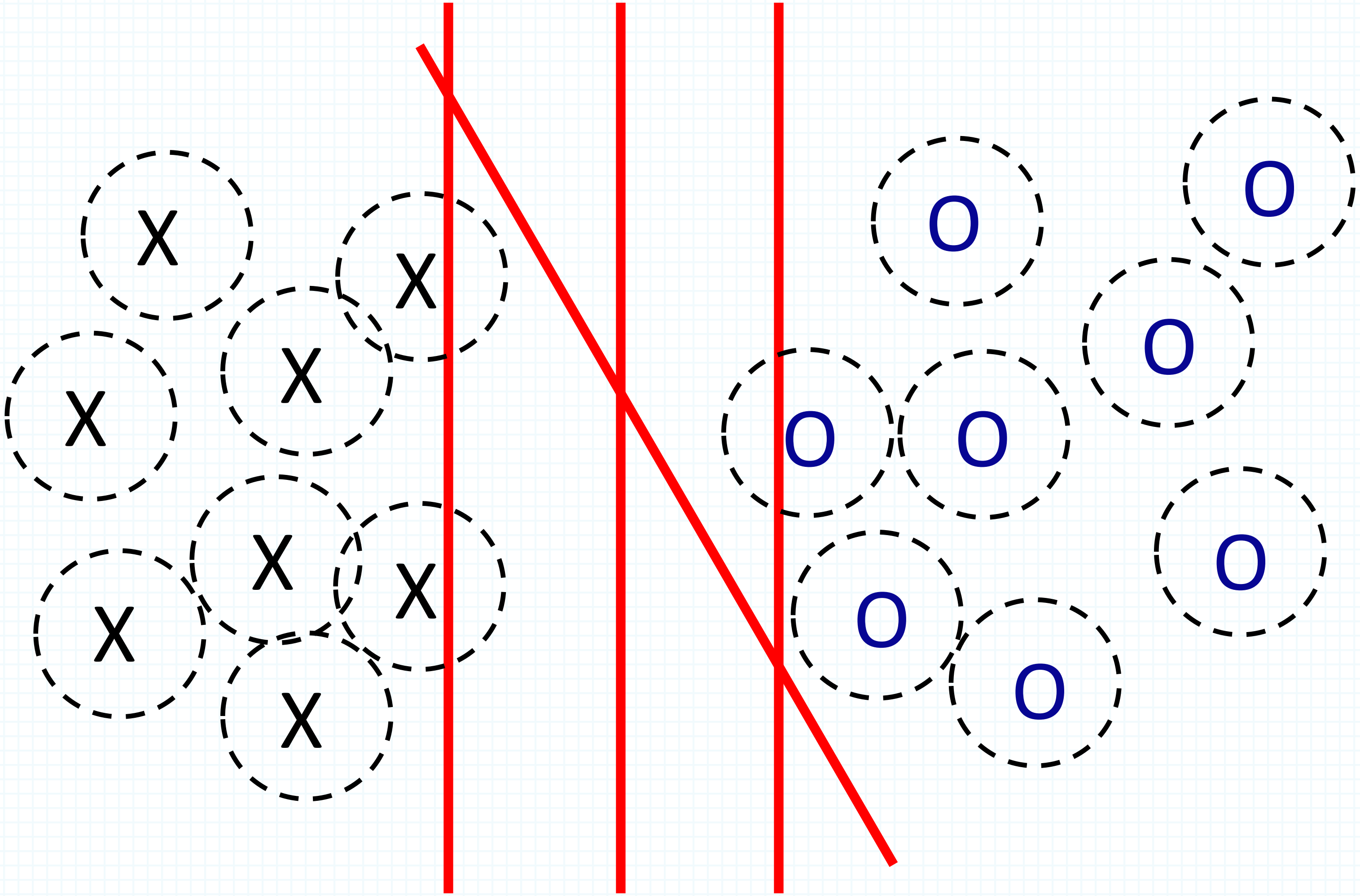
A “Good” Separator



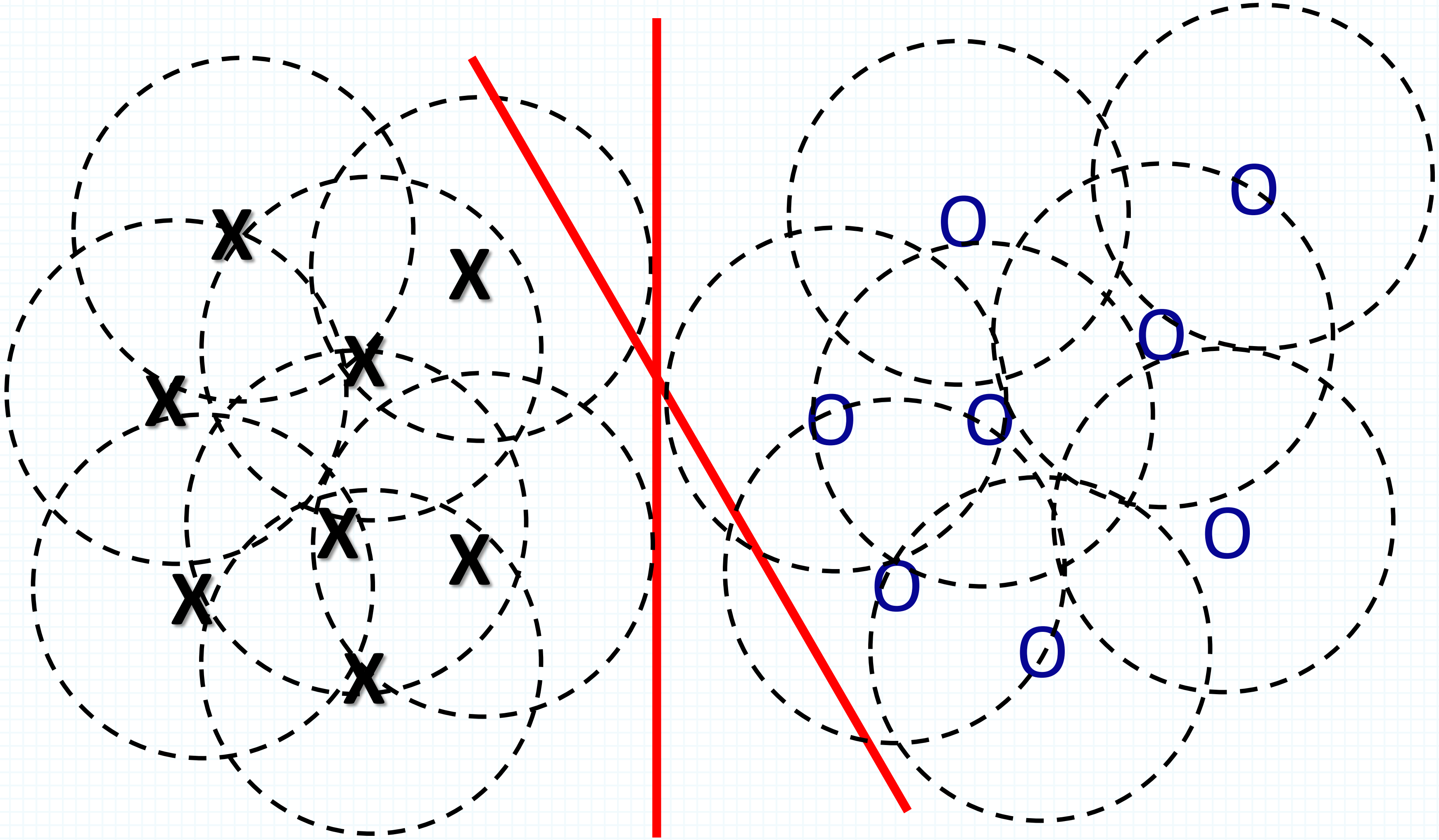
Noise in the Observations



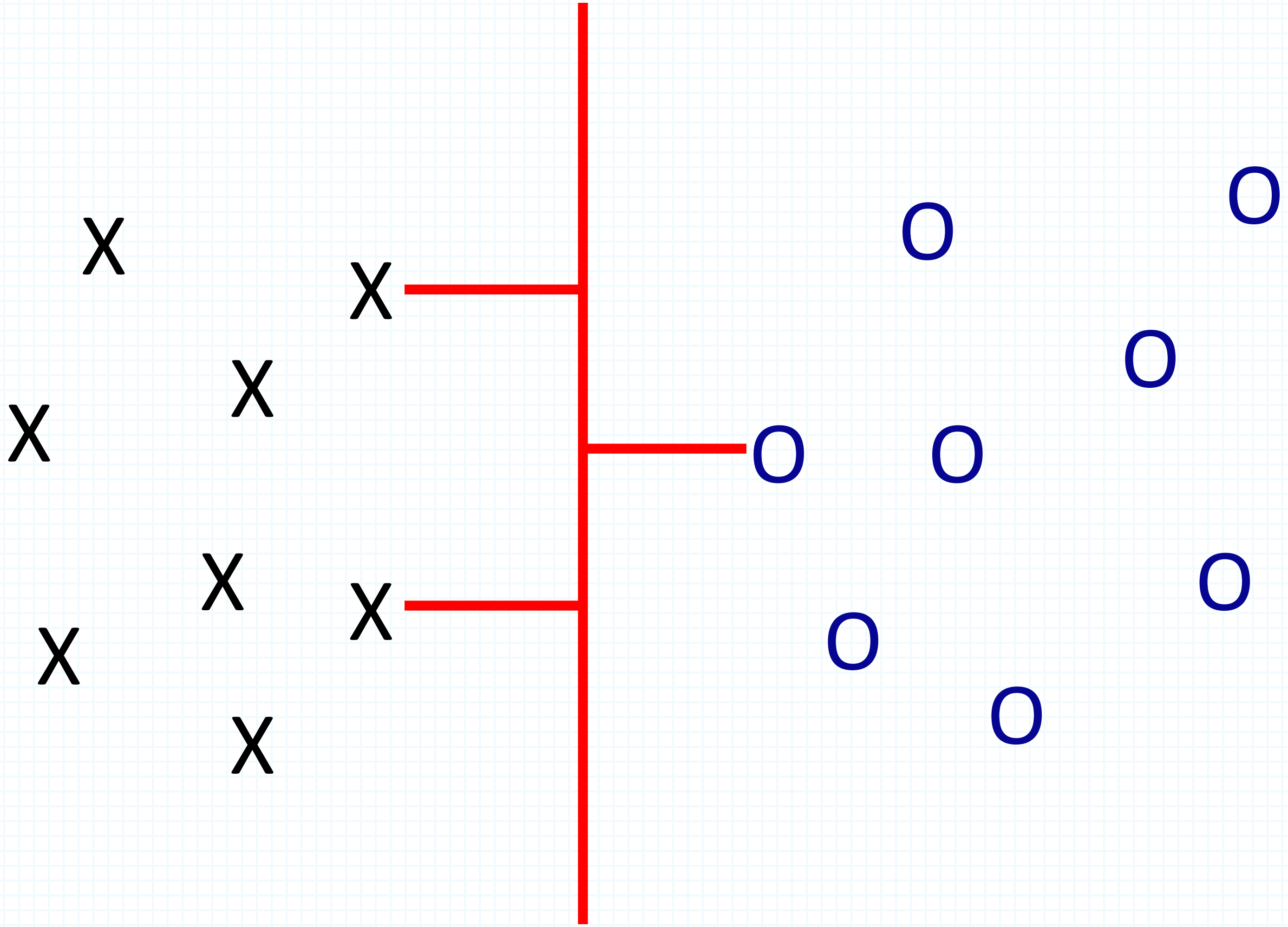
Ruling Out Some Separators



Lots of Noise

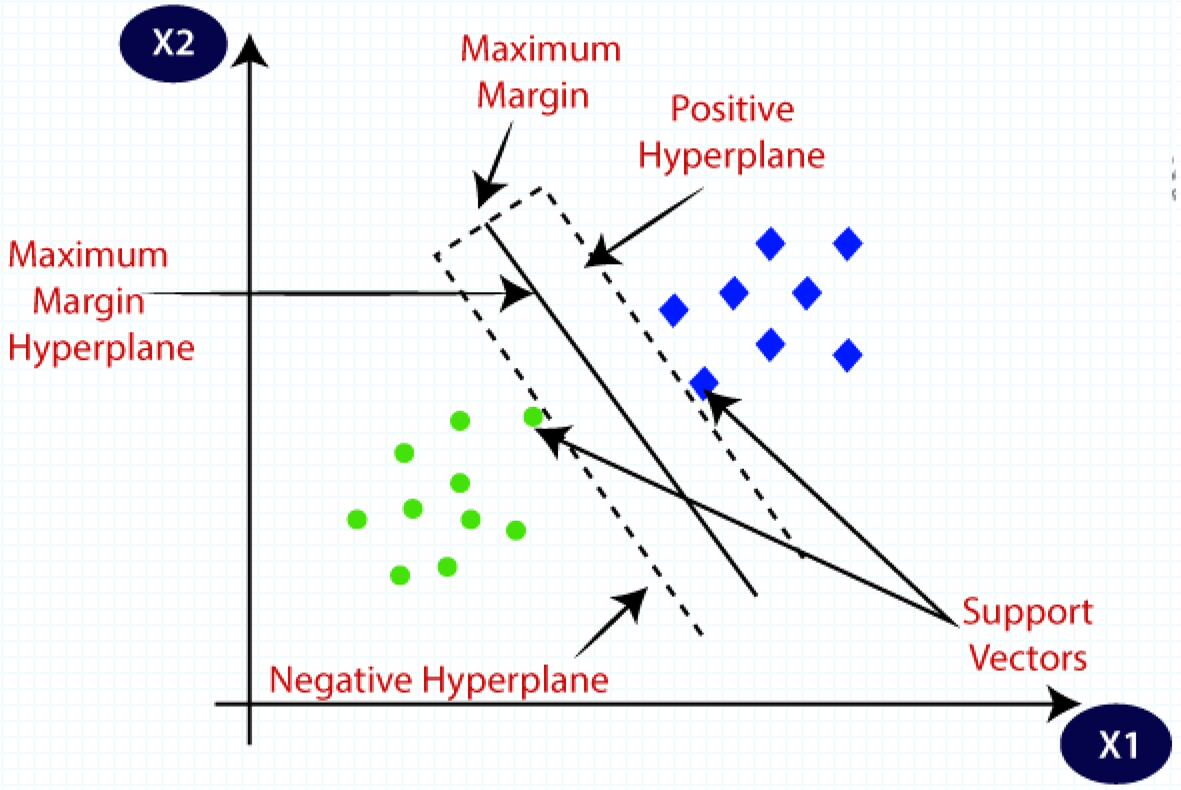


Maximizing the Margin



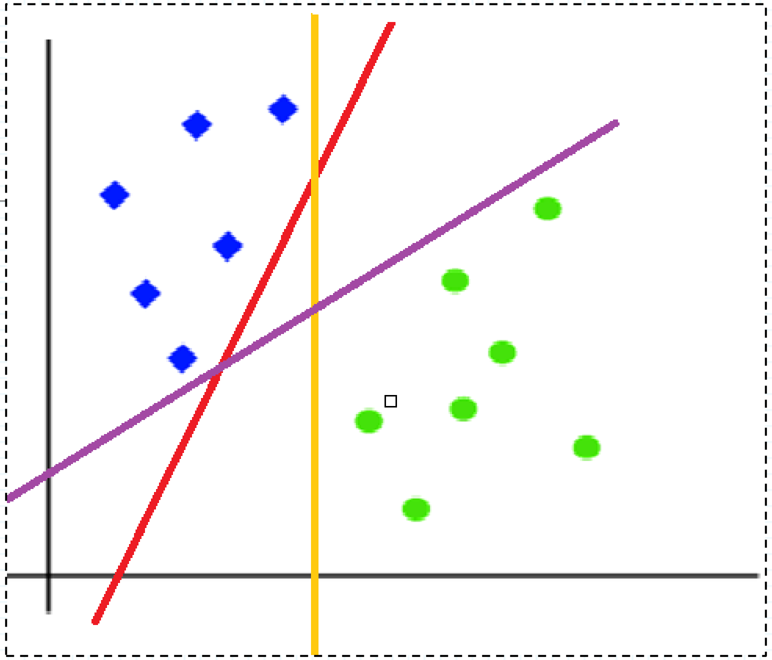
Terms

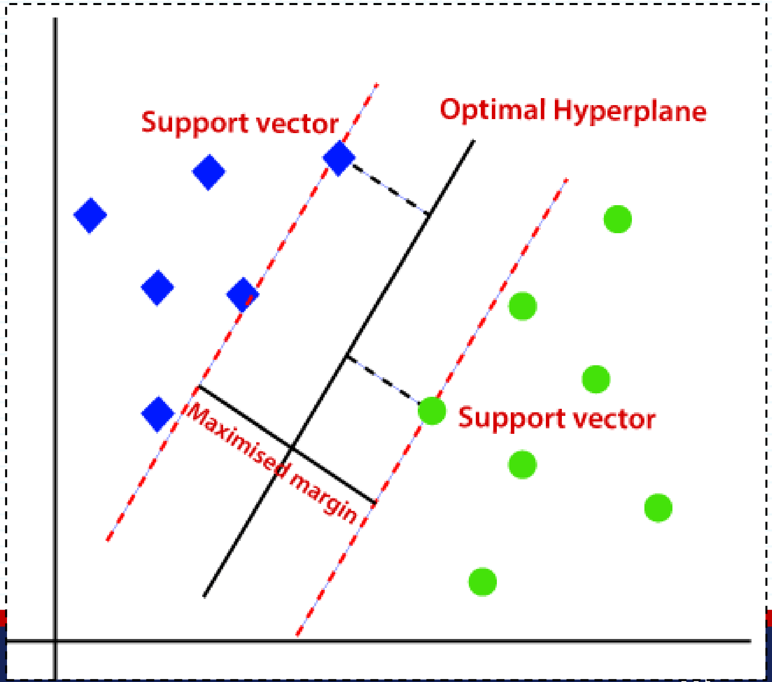
- **Support Vectors:**
  - These are the points that are closest to the hyperplane.
  - A separating line will be defined with the help of these data points.
- **Margin:**
  - It is the distance between the hyperplane and the observations closest to the hyperplane (support vectors).
  - In SVM large margin is considered a good margin.
  - There are two types of margins **hard margin** and **soft margin.**



How does SVM work?

- SVM is defined such that it is defined in terms of the support vectors only.
  - The margin is made using the points which are closest to the hyperplane (support vectors).
  - We don't have to worry about other observations
  - Hence SVM enjoys some natural speed-ups!
- The best hyperplane is that plane that has the maximum distance from both the classes.



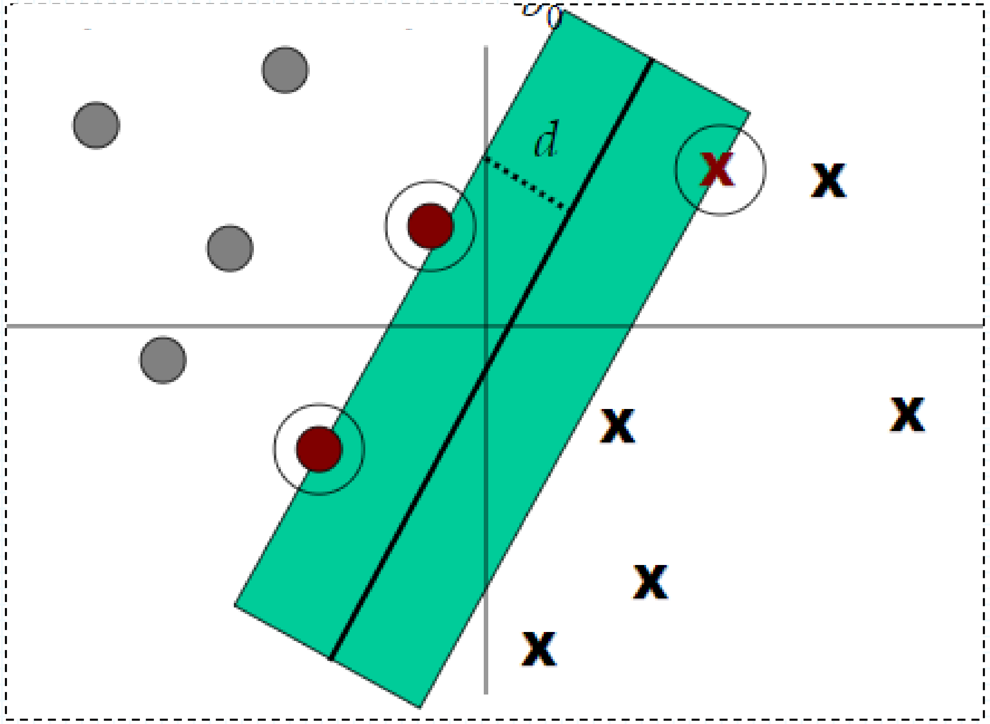


So.. What is our optimization problem?

- **Our problem:**

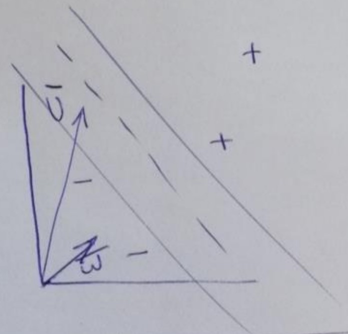
Maximizing the shortest distance to the closest positive or negative point.

$$w^{*} = arg_{w}max\,[min_{n}\, d_{H}(\phi(x_{n}))]$$



Note that W represents all parameters i.e. w and b

للاطلاع



$\bar{w} \cdot \bar{u} \geq c$ the N^+

$\bar{w} \cdot \bar{u} + b \geq 0$

$\bar{w}\, x_+^* + b \geq 1$

$w\, x_- + b \leq -1$

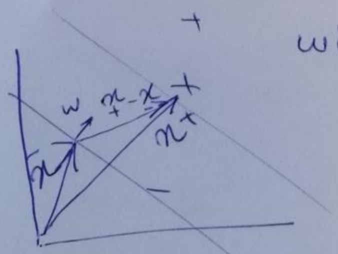
$y_i = +1$ for + samples

-1 for − samples

constraint

$y_i(\bar{w} \cdot \bar{x}_i + b) \geq 1$ for all points

(*) $y_i(\bar{w}\bar{x}_i + b) = 1$ for support vector



width $= (\bar{x}_+ - \bar{x}_-) \cdot \frac{\bar{w}}{\|w\|}$

support vector

(*)

$= (\bar{x}_+ \bar{w} - \bar{x}_- \bar{w}) \frac{1}{\|w\|}$

from (*)

$x_+ \to \bar{w} x_+ + b = 1 \to \bar{w} x_+ = 1 - b$

$x_- \to \bar{w} x_- + b = -1 \to -\bar{w} x_- = 1 + b$

$\Downarrow$

width $= \frac{2}{\|w\|}$

Goal Maximize width $\frac{2}{\|w\|}$

$\Downarrow$

Minimize $\|w\|$

$\Downarrow$

Goal $\boxed{\text{Minimize } \frac{1}{2}\|w\|^2}$

---

true only if the constraint is satisfied

$\Downarrow$

use lagrange Multiplier

$L = \frac{1}{2}\|\bar{w}\|^2 - \sum \alpha_i [y_i(\bar{w}\bar{x}_i + b) - 1]$

SVM Optimization

$$w^*, b^* = \arg \underset{w,b}{Min} \frac{1}{2} \|w\|^2 \;, \quad s.t. \;\; y_n (w^T (\emptyset(x_n)) + b) \geq 1 \quad \forall n$$

Solved by Lagrange multiplier method:

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 \; - \sum_n \alpha_n [y_n (w^T (\emptyset(x_n)) + b) - 1]$$

where $\boldsymbol{\alpha}$ is the Lagrange multiplier

The optimization problem can be solved by setting **derivatives** of *Lagrangian* to 0

SVM Optimization

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 \; - \sum_n \alpha_n [y_n \left(w^T \left(\emptyset(x_n)\right) + b\right) - 1]$$

$$\begin{cases} \dfrac{\partial L}{\partial w} = w \; - \displaystyle\sum_n \alpha_n y_n \emptyset(x_n) \quad = 0 \;\; \Rightarrow \;\; w = \displaystyle\sum_n \alpha_n y_n \emptyset(x_n) \\ \\ \dfrac{\partial L}{\partial b} = \displaystyle\sum_n \alpha_n y_n = 0 \;\; \Rightarrow \displaystyle\sum_n \alpha_n y_n \; = 0 \end{cases}$$

SVM Optimization

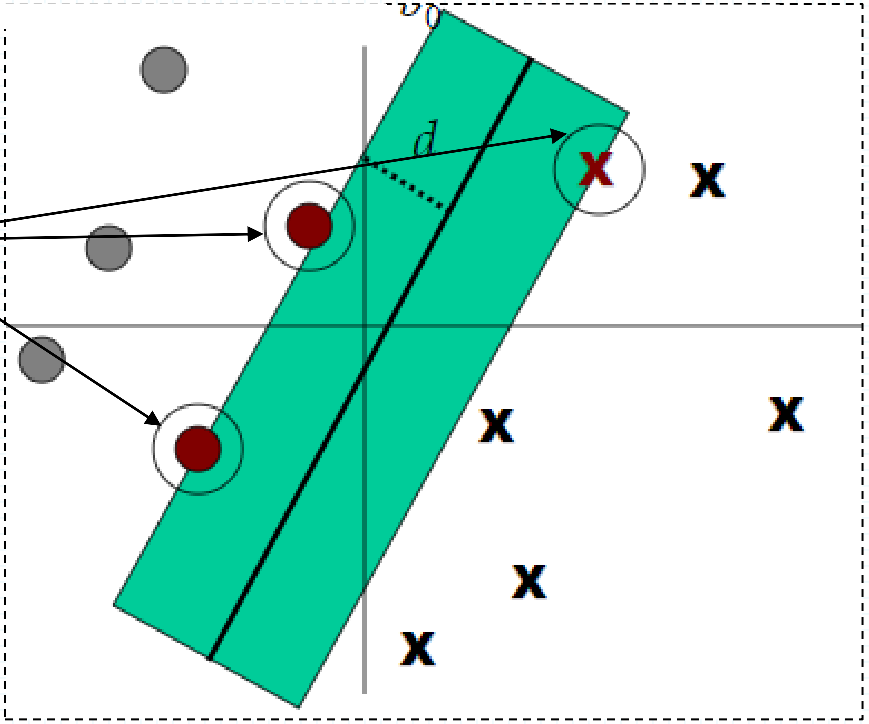
$$w^*, b^* = \arg \underset{w,b}{Min} \frac{1}{2} \|w\|^2 , \quad s.t. \;\; y_n (w^T (\emptyset(x_n)) + b) \geq 1 \quad \forall n$$

$$\text{Y} = w^T (\emptyset(x)) + b = \sum_n \alpha_n y_n \emptyset^T (x_n) \; \emptyset(x)$$

The decision rule in SVMs only depends on the dot product with support vectors

several important implications

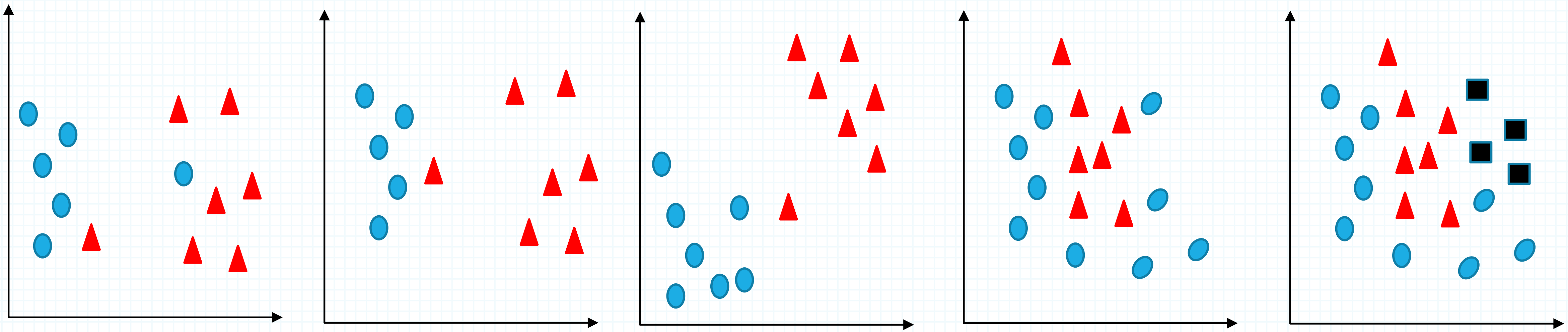
Computational efficiency
Memory efficiency
Robustness to noise and outliers



What if?

What are the problems of the current version for SVM?

(A) (B) (C) (D) (E)



1st Improvement

Soft Margin SVM (allows few misclassifications)

C Hyper-parameter

- When **C** is high it will classify all the data points correctly, also there is a chance to overfit.

$$\underset{(w^*, b^*)}{\text{argmin}} \frac{\|w\|}{2} + c\sum_{i=1}^{n} \zeta_i$$

- ***SVM Error = Margin Error + Classification Error***