الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

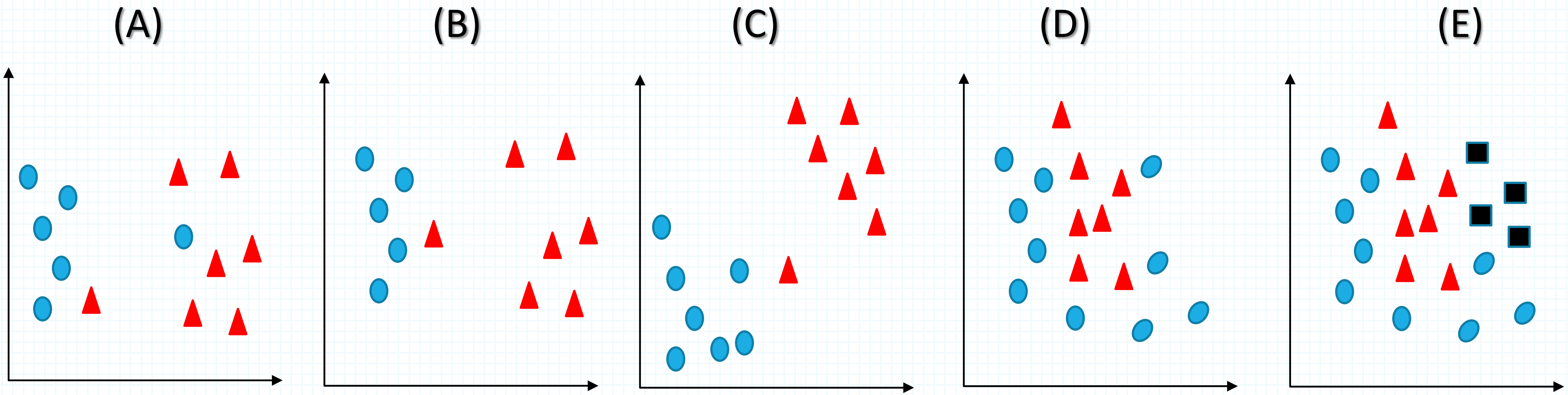
مقرر تعلم الآلة | كلية الهندسة المعلوماتية | المحاضرة 7

Support Vector Machine (SVM) 2

د. رياض سنبل

What if?

What are the problems of the current version for SVM?



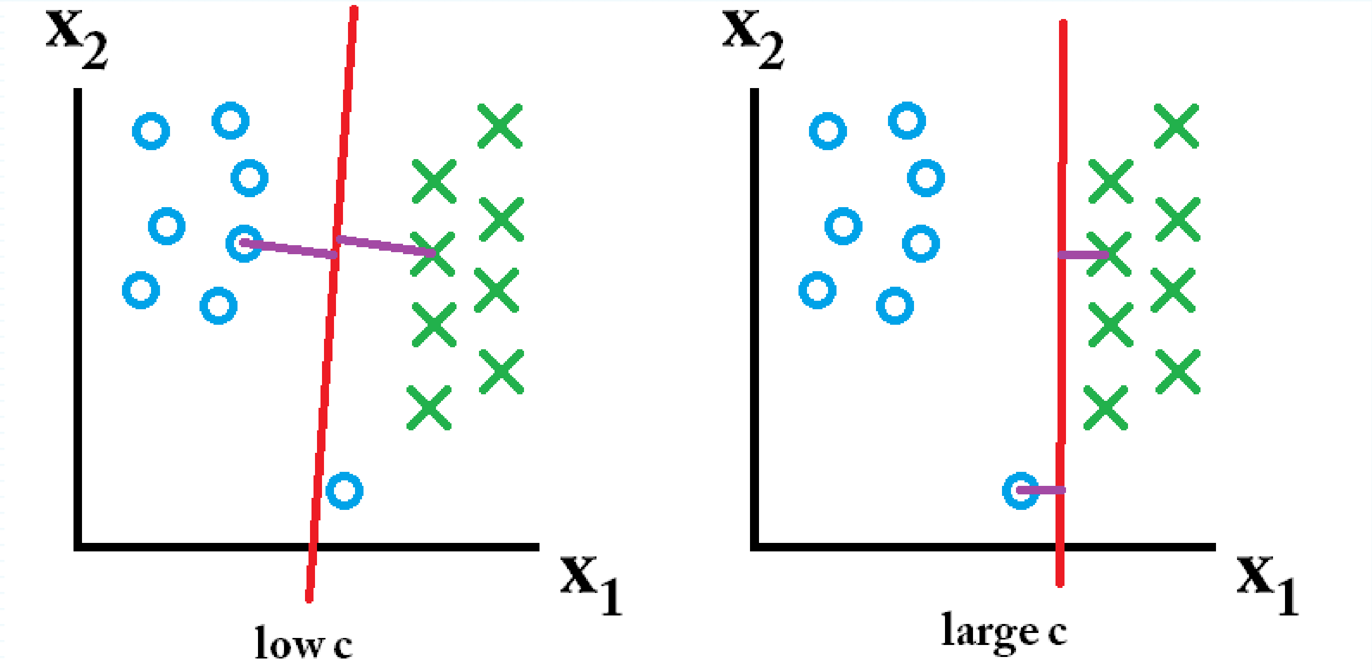
1st Improvement
Soft Margin SVM
(allows few misclassifications)

C Hyper-parameter

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

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

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

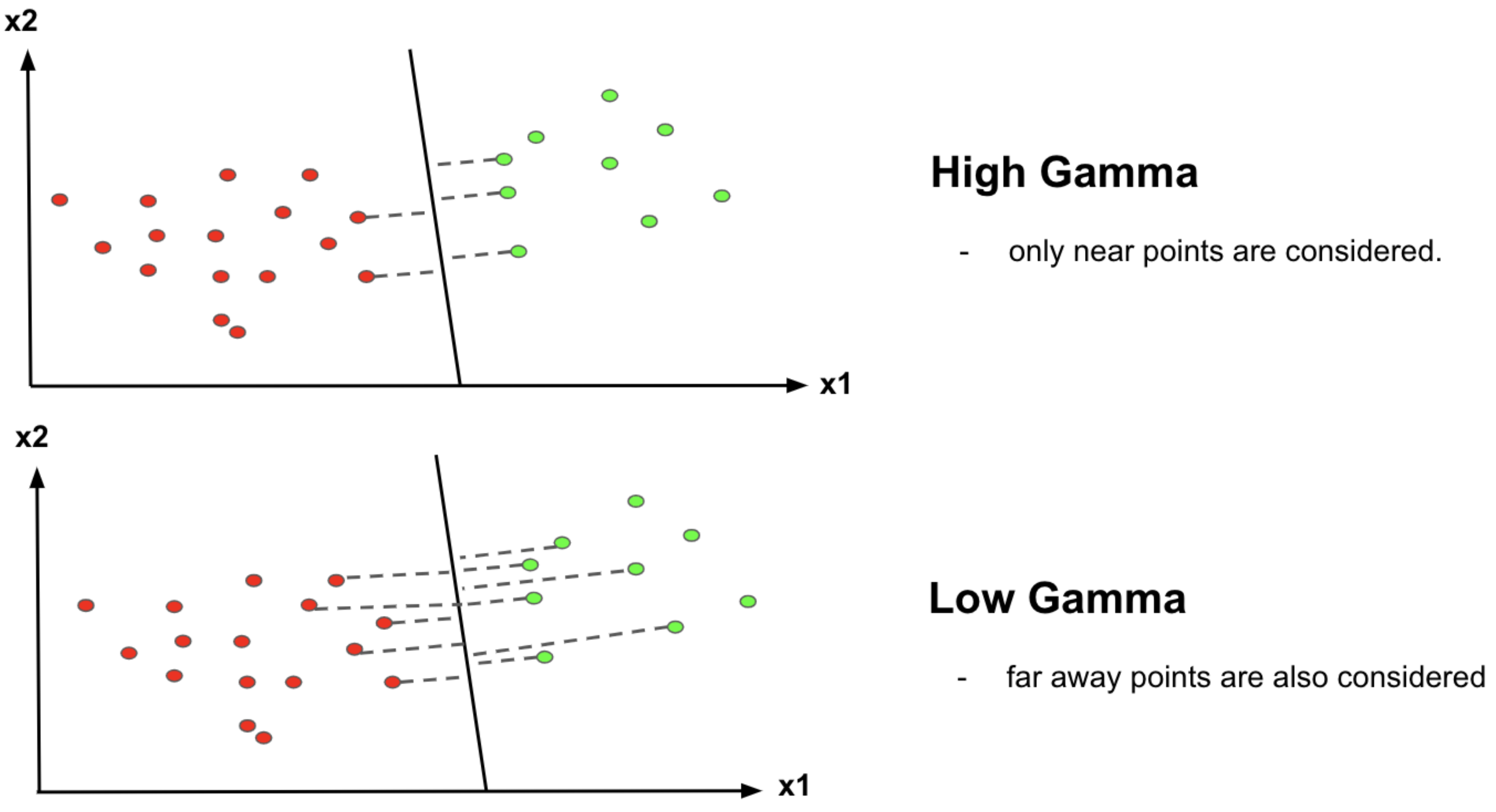


2nd Improvement

Consider more points to get the decision boundary.

Gamma Hyper-Parameters

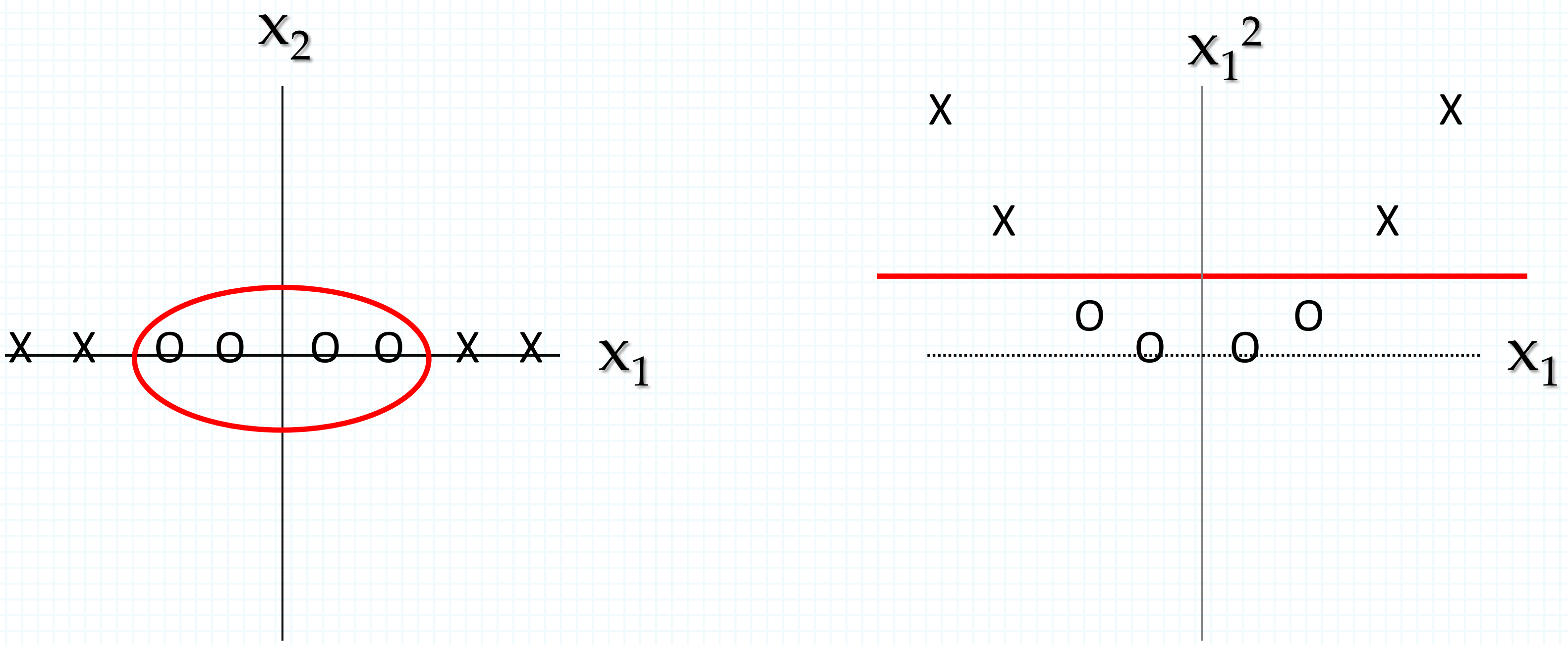
- It defines how far influences the calculation of plausible line of separation.
- when gamma is higher, nearby points will have high influence; low gamma means far away points also be considered to get the decision boundary.



3rd Improvement
How to make a plane curved?

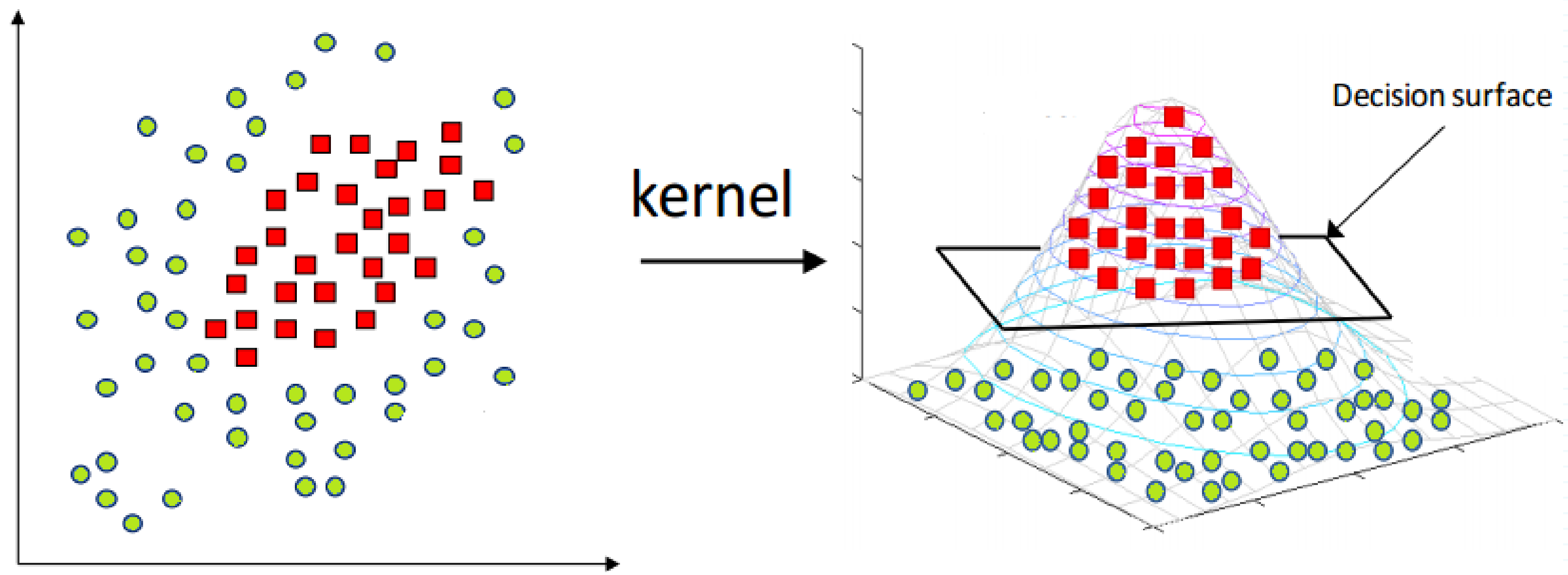
When Linear Separators Fail

- The most interesting feature of SVM is that it can even work with a non-linear dataset.
- We use “Kernel Trick” which makes it easier to classifies the points.



The Kernel Trick

- try converting this lower dimension space to a higher dimension space using some quadratic functions which will allow us to find a decision boundary that clearly divides the data points.
- These functions which help us do this are called Kernels and which kernel to use is purely determined by hyperparameter tuning.



Kernel functions: Polynomial kernel

- Following is the formula for the polynomial kernel:

$$f(X1, X2) = (X1^T . X2 + 1)^d$$

- Here d is the degree of the polynomial, which we need to specify manually.
- Suppose we have two features X1 and X2 and output variable as Y, so using polynomial kernel we can write it as:

$$X1^T . X2 = \begin{bmatrix} X1 \\ X2 \end{bmatrix} . \begin{bmatrix} X1 & X2 \end{bmatrix}$$

$$= \begin{bmatrix} X1^2 & X1 . X2 \\ X1 . X2 & X2^2 \end{bmatrix}$$

- So we basically need to find $X1^2$, $X2^2$ and X1.X2, and now we can see that 2 dimensions got converted into 5 dimensions.

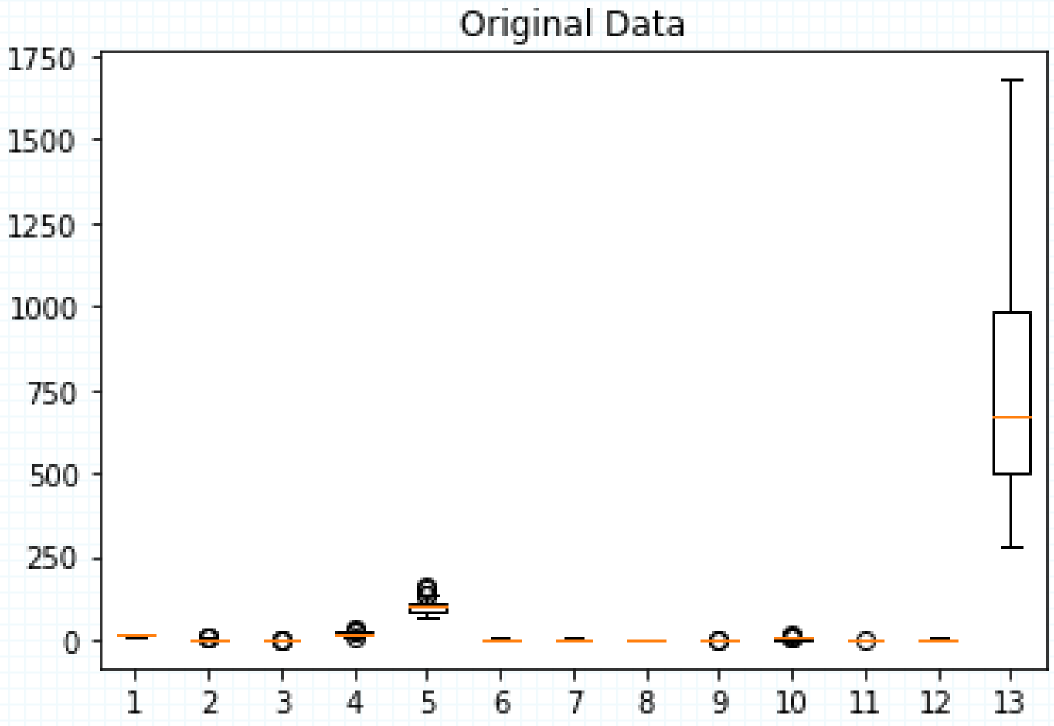
Other commonly used kernels

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma \langle x, x' \rangle + r)^d$, where d is specified by parameter `degree`, r by `coef0`.
- rbf: $\exp(-\gamma \|x - x'\|^2)$, where γ is specified by parameter `gamma`, must be greater than 0.
- sigmoid $\tanh(\gamma \langle x, x' \rangle + r)$, where r is specified by `coef0`.

4^{th} Improvement
Feature Scaling

Why feature scaling?

- Why?
- Any suggestion?



Feature scaling

- **Feature scaling** is mapping the feature values of a dataset into the same range.
- The two most widely adopted approaches for feature scaling are normalization and standardization.
- Normalization maps the values into the [0, 1] interval:

$$z = \frac{x - min(x)}{max(x) - min(x)}$$

- Standardization shifts the feature values to have a mean of zero, then maps them into a range such that they have a standard deviation of 1:

$$z = \frac{x - \mu}{\sigma}$$

  - It centers the data, and it's more flexible to new values that are not yet seen in the dataset. That's why we prefer standardization in general.

Feature scaling