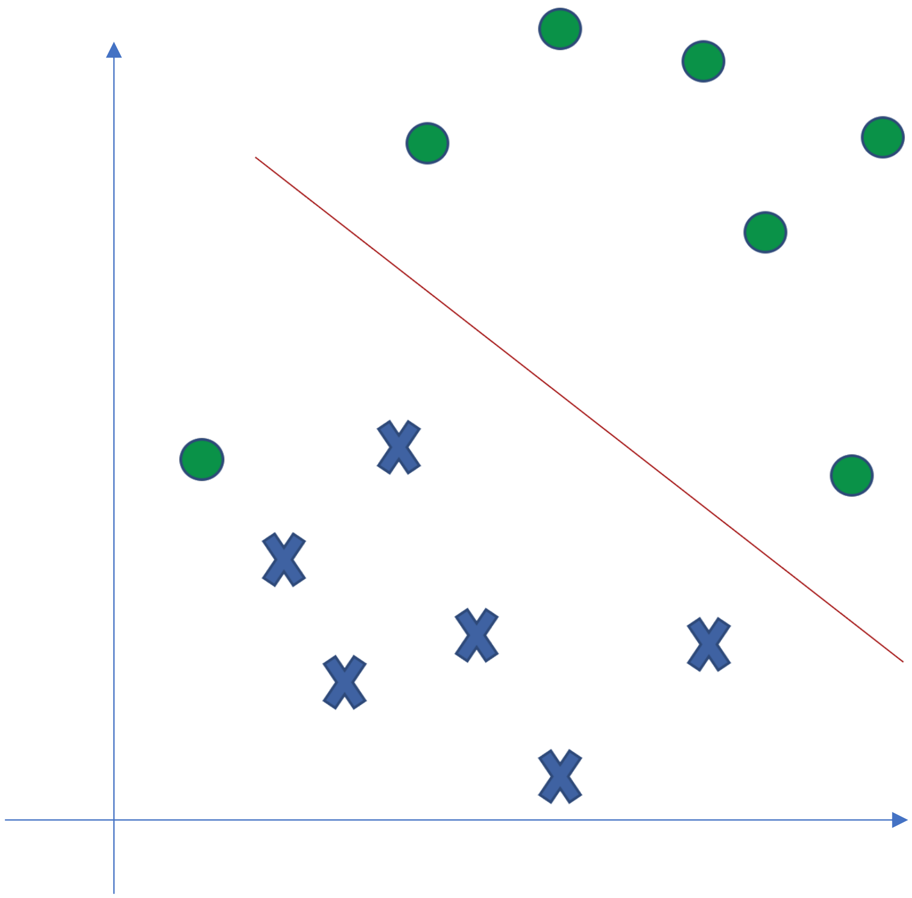
Linear Machines and SVM – Part 4: SVM for Non-linearly-separable Case

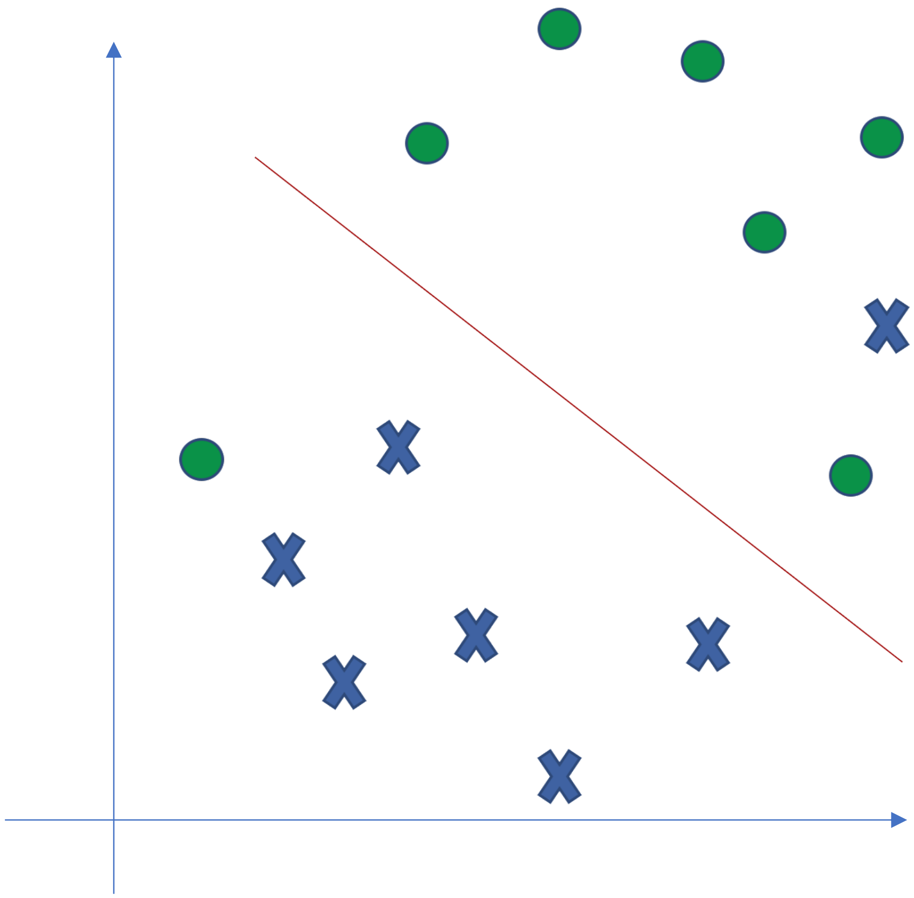
Ira A. Fulton Schools of Engineering
Arizona State University

Linear Separability Violated



Some samples will always be misclassified no matter what $\{\mathbf{w}, b\}$ is used.

Examining Misclassified Samples



- They will violate the constraints:

$\mathbf{w}^t\mathbf{x}^{(i)} + b \geq 1$ for $y^{(i)} = +1$

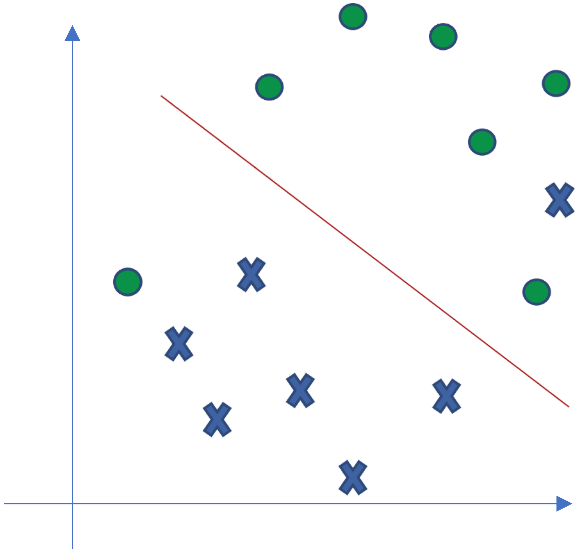
$\mathbf{w}^t\mathbf{x}^{(i)} + b \leq -1$ for $y^{(i)} = -1$

Relaxing the Constraints

- Introducing *non-negative* slack variables ξ_i

$$\mathbf{w}^t\mathbf{x}^{(i)} + b \geq 1 - \xi_i \quad \text{for } y^{(i)} = +1$$

$$\mathbf{w}^t\mathbf{x}^{(i)} + b \leq -1 + \xi_i \quad \text{for } y^{(i)} = -1$$



- For an error to occur, the corresponding ξ_i must exceed unity.
  - *Hinge loss* or *soft margin*.

➔ $\sum_i \xi_i$ provides an upper bound on the number of training errors.

Updating the Formulation

$$\{\mathbf{w}^*, b^*\} = \underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||^2 + C\left(\sum_{\mathrm{i}} \xi_i\right)$$

Subject to

$\mathbf{w}^t\mathbf{x}^{(i)} + b \geq 1 - \xi_i$ for $y^{(i)} = +1$

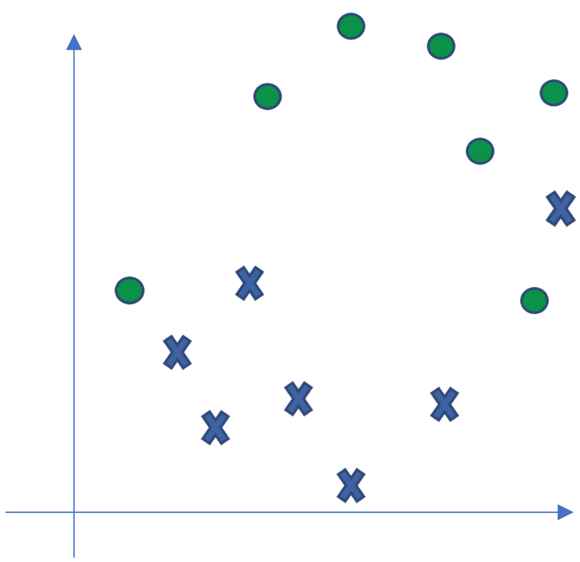
$\mathbf{w}^t\mathbf{x}^{(i)} + b \leq -1 + \xi_i$ for $y^{(i)} = -1$

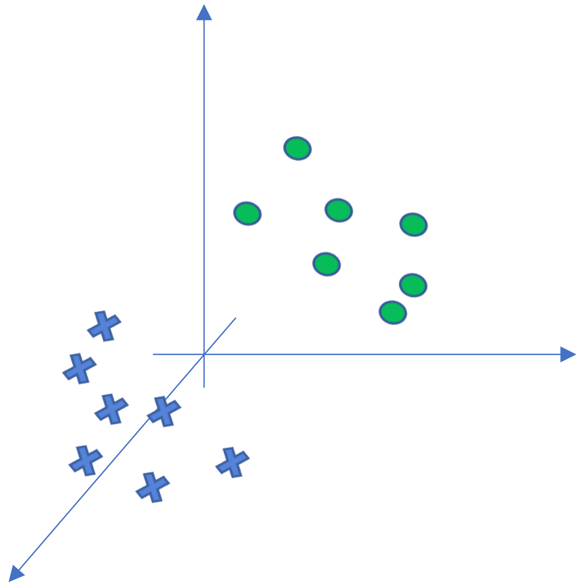
$\xi_i \geq 0, \forall i$

- C is a parameter to control how much penalty is assigned to errors.

Are non-linear decision boundaries possible?

- Transform data to higher dimensions using a mapping
  - More freedom to position the samples
  - May make the samples linearly separable
  - Run linear SVM in the new space ➔ may be equivalent to non-linear boundaries in the original space





- What mapping to use?

The Kernel Trick

Revisit the Lagrange Dual Formulation for SVM

$$L_D(\mathbf{w}, b, \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)} \cdot \mathbf{x}^{(j)}$$

Introduce a **kernel** function

$$L_D(\mathbf{w}, b, \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y^{(i)} y^{(j)} K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$$

The Kernel Trick (cont'd)

Mercer's Theorem: for a symmetric, non-negative definite kernel function satisfying some minor conditions, there exists a mapping Φ(x) such that

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \Phi(\mathbf{x}^{(i)}) \cdot \Phi(\mathbf{x}^{(j)})$$

→ Using a kernel function in L_D can effectively defines an implicit mapping to a higher-dimensional space, where linear SVM was run.

→ The decision boundaries in the original space can be highly non-linear.

Common Kernel Functions

- Polynomials of degree *d*

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle^{d}$$

- Polynomials of degree up to *d*

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \left(\left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle + 1\right)^{d}$$

- Gaussian kernels

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \exp\left(-\frac{\left\|\mathbf{x}^{(i)} - \mathbf{x}^{(j)}\right\|^2}{2\sigma^2}\right)$$

- Sigmoid kernel

$$K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \tanh\left(\eta\left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle + \nu\right)$$