Linear Machines & SVM

Linear Machines

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Define general
linear classifiers

Revisiting Logistic Regression

In Logistic Regression: given a training set of *n* labelled samples <$x^{(i)}$, $y^{(i)}$ >, we learn *P*(y|x) by assuming a logistic sigmoid function.

- ➔ We end up with a *linear classifier*.
- ➔ $g(x) = w^t x$ is called the *discriminant function*.

Linear Discriminant Functions

In general, taking a discriminative approach, we can *assume* some form for the discriminant function that defines the classifier.

➔ The learning task is to use the training samples to estimate the parameters of the classifier.

Linear Decision Boundaries

- **Linear discriminant functions give arise to liner decision boundaries**

  ➔ *linear classifiers* or *linear machines*

- **We will use both notations:**

$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} \quad \textit{or} \quad g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0$$

Linear Machine for C>2 Classes

- We can define C linear discriminant functions:

$$g_i(\mathbf{x}) = \mathbf{w}_i^t\mathbf{x}, \quad i = 1, 2, \ldots, C$$

- What is the decision rule for the classifier?

The Learning Task

- **Finding $\mathbf{w}_i$, $i = 1, 2, \ldots, C$**
- **Let's use the 2-class case as an example**
  - For n samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, of 2 classes ω_1 and ω_2 , **if** there exists a vector **w** such that $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x}$ classifies them all correctly ➔ Finding **w**
- **i.e., finding w such that**

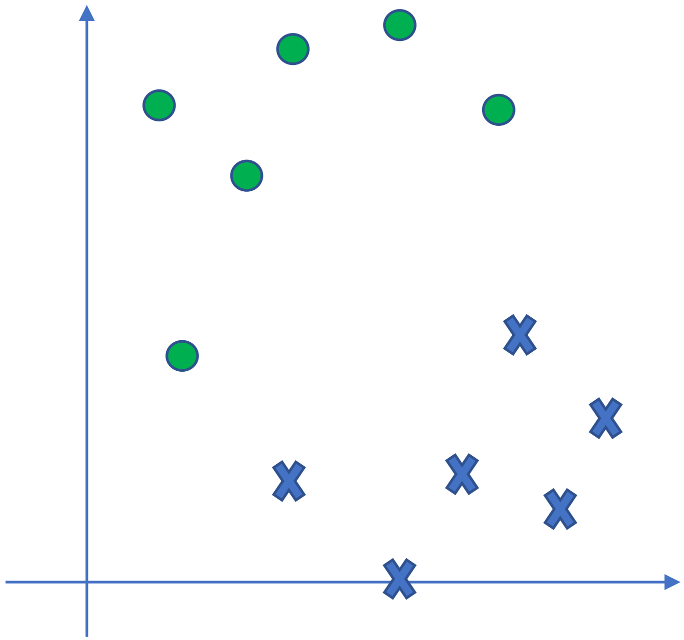
$\mathbf{w}^t\mathbf{x}_i \geq 0$ for samples of ω_1 and
$\mathbf{w}^t\mathbf{x}_i < 0$ for samples of ω_2,

Linear Separability

- **If we can find at least one vector w such that $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x}$ classifies all samples**
  - ➔ We say the samples are linearly separable.
- **An example of not linearly separable in 2-D:**

The Solution Region

- **There may be many different weight vectors that can all be valid solutions for a given training set**
  - ➔ The solution regions
- **If the solution vector is not unique, *Which one is the best?***



Solving for the Weight Vector

- **Consider the following approach: finding a solution vector which optimizes some objective function.**

➔ We may introduce additional constraints for a “good” solution”

➔ **Solving a constrained optimization problem.**

- **Theoretical: Lagrange or Karush-Kuhn-Tucker.**

- **In practice: e.g., gradient-descent-based search**

Gradient Descent Procedure

Basic idea:

- Define a cost function $J(\mathbf{w})$
- Starting from an initial weight vector $\mathbf{w}(0)$
- Update $\mathbf{w}$ by

$$\mathbf{w}(k+1) = \mathbf{w}(k) - \eta(k)\nabla J\big(\mathbf{w}(k)\big),$$

η>0 is the *learning rate.*

Linear Machines & SVM

The Concept of Margins

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

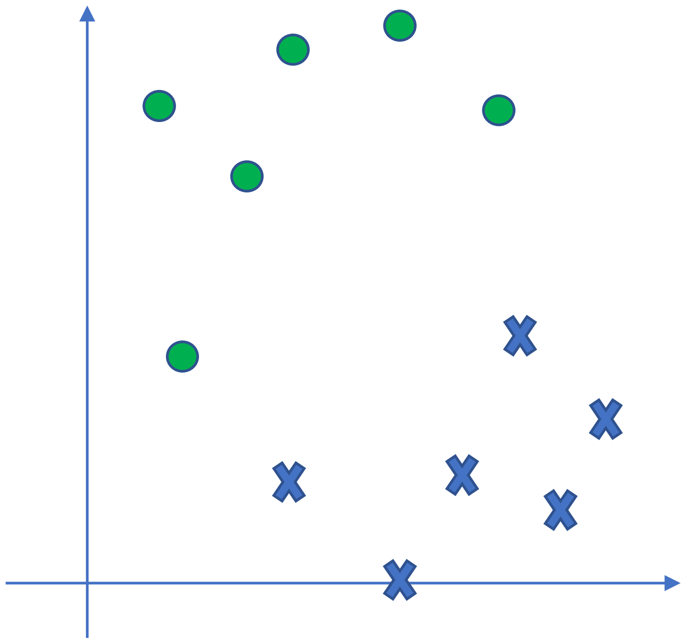
Objective

Illustrate Margins
in Classifier

Illustrating Linear Boundaries

- **The decision boundaries is given by the line $g(\mathbf{x}) = 0$.**
  - For appreciating a geometric interpretation, we will write w_0 explicitly, i.e., we have

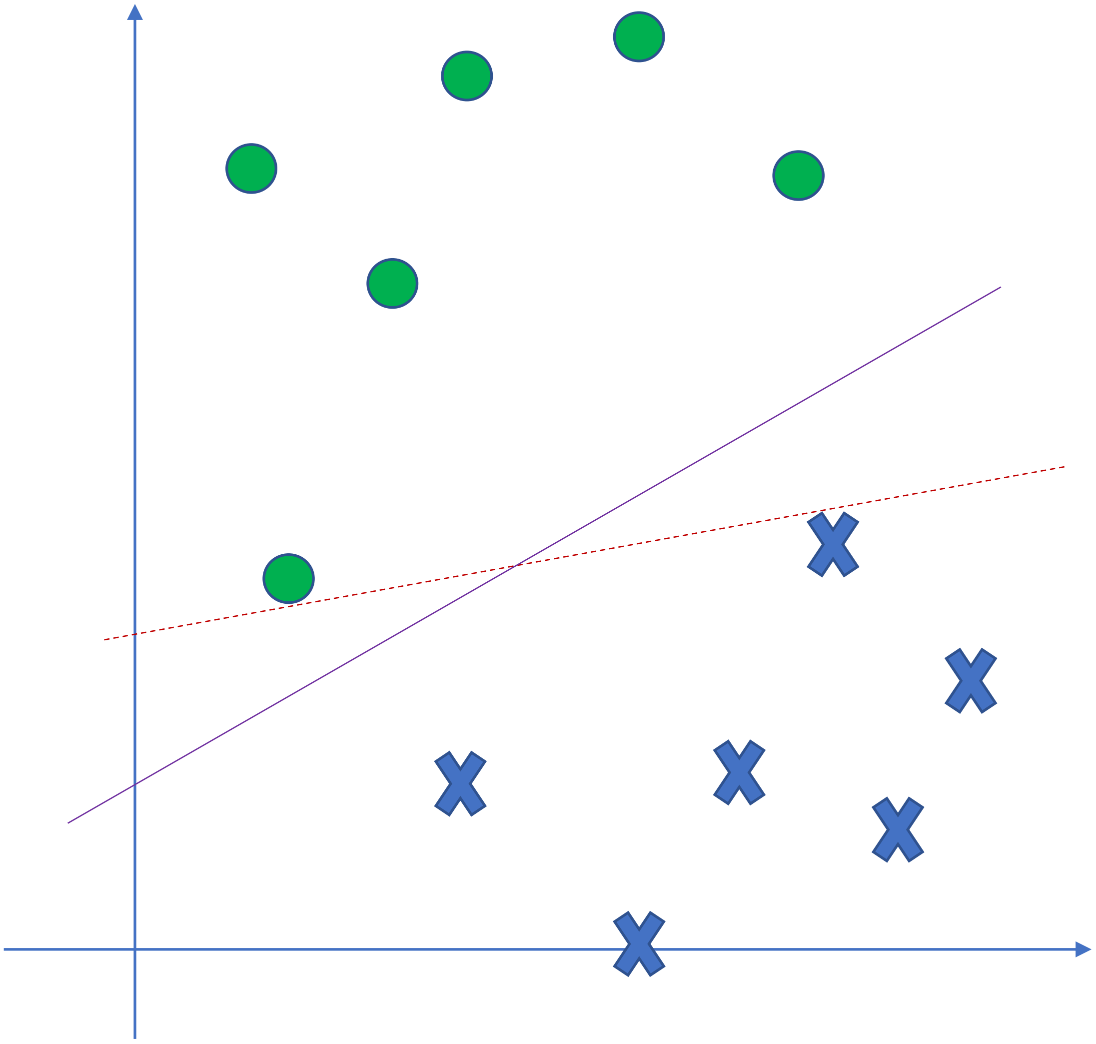
$$g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0$$



- **The normal vector of the decision line/plane is**

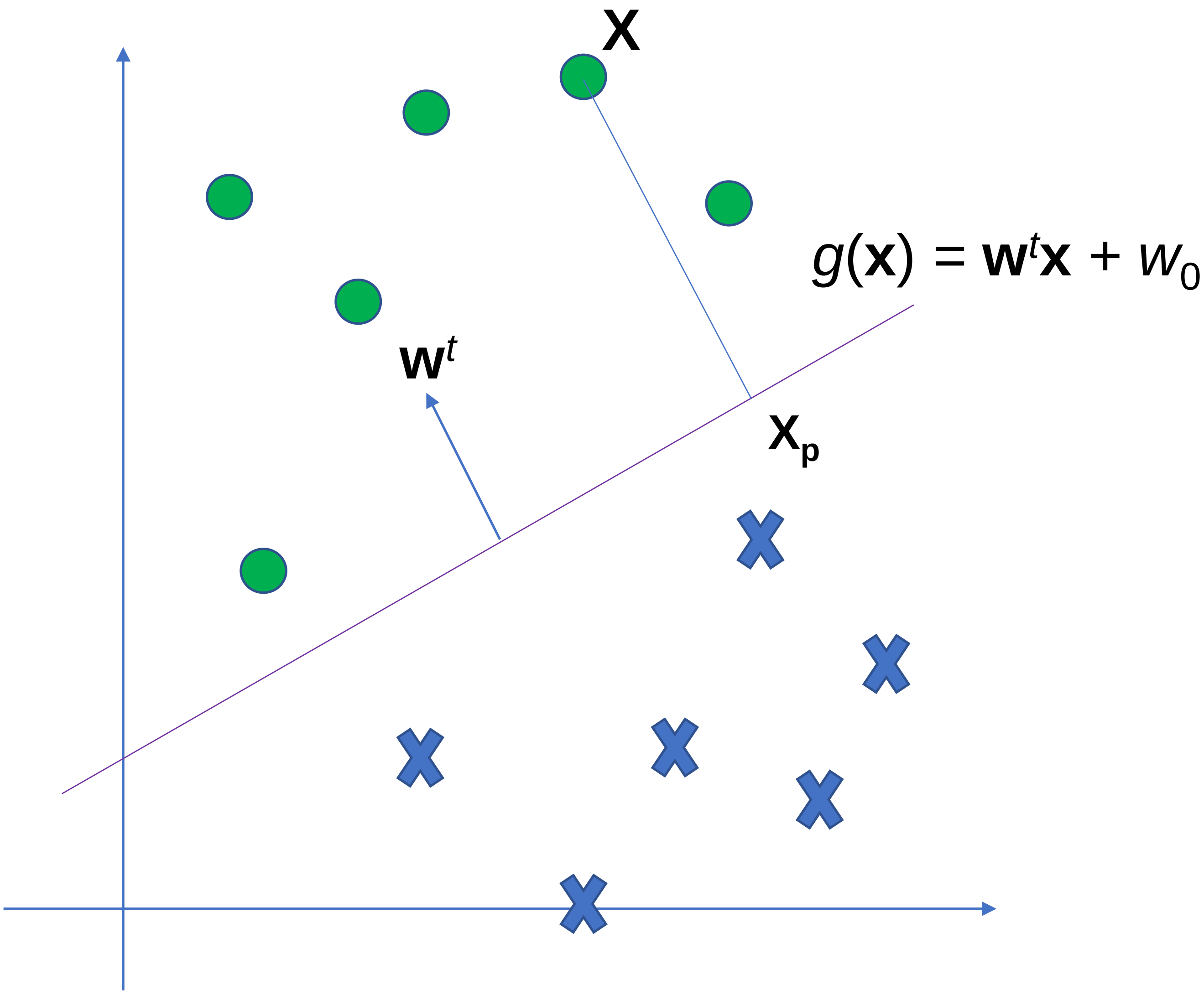
____

Which one is better?

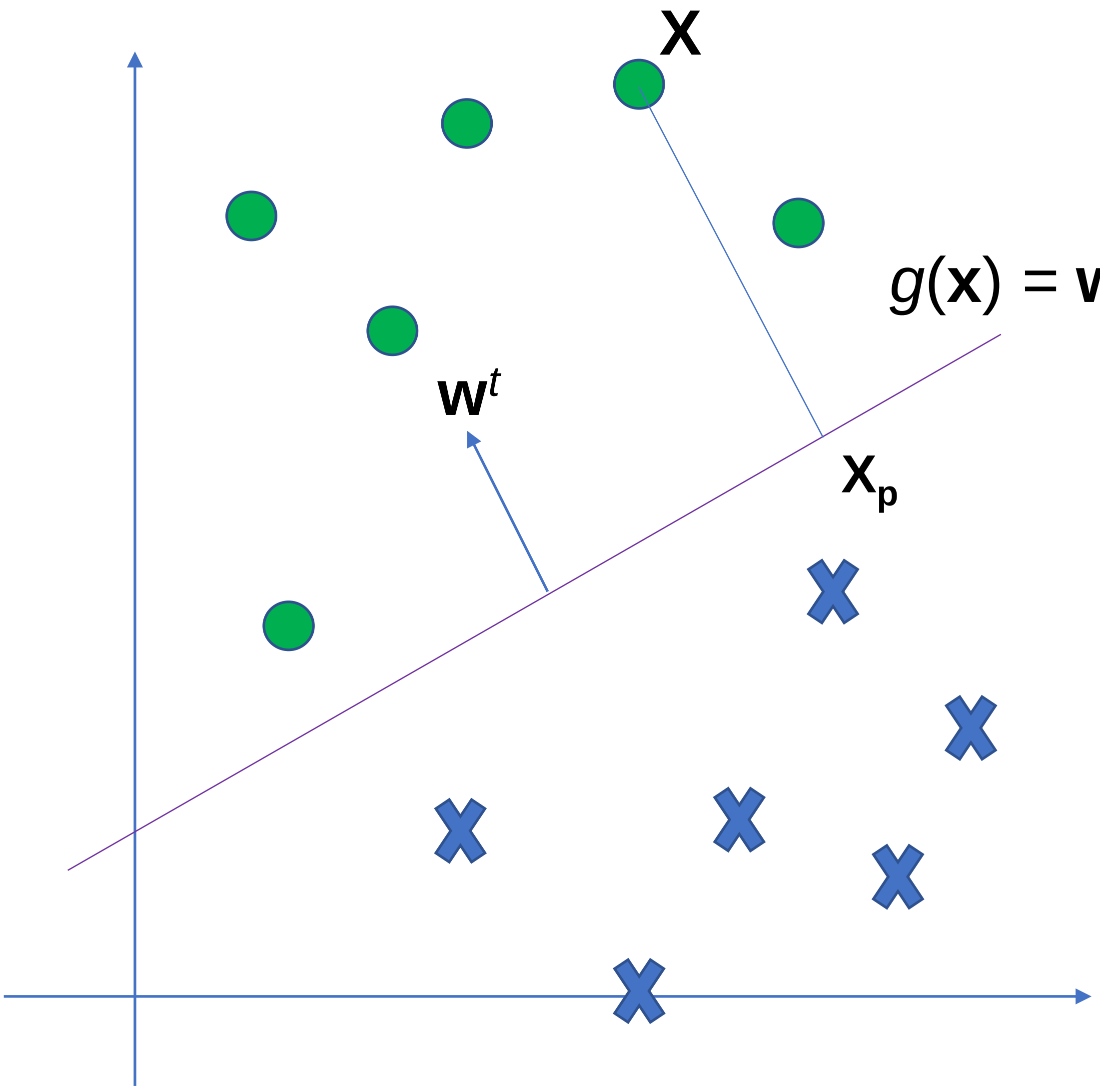


➔ Consider the distances of the samples to the decision plane.

Distance to the Decision Plane



Distance to the Decision Plane

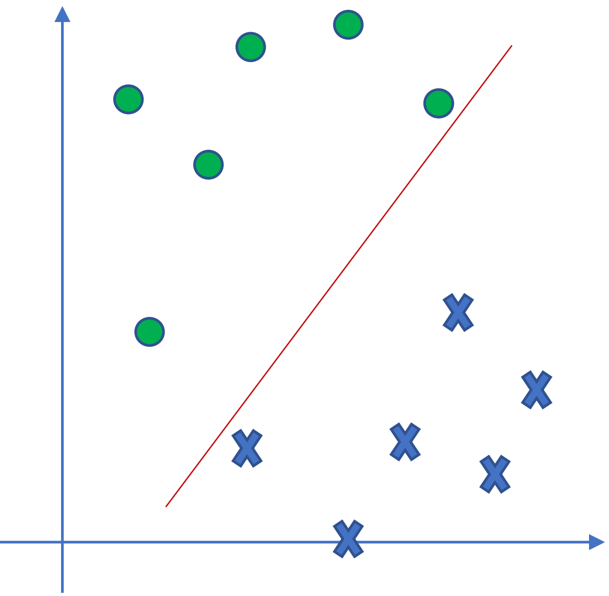


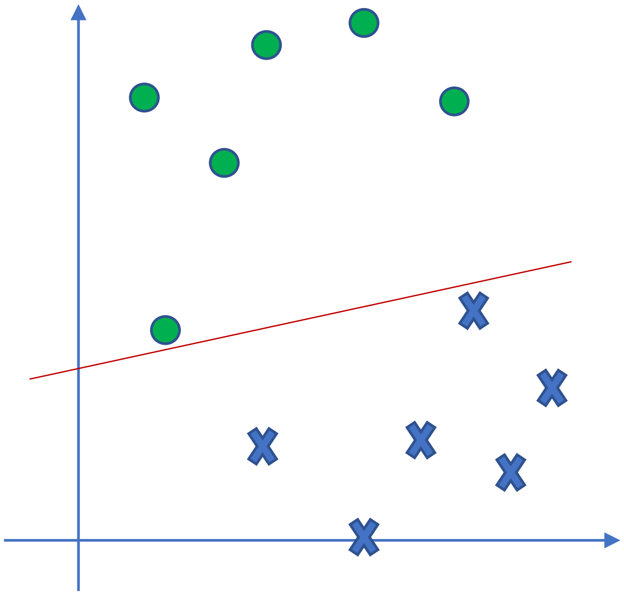
g(x) gives an algebraic measure of the distance from x to the decision plane.

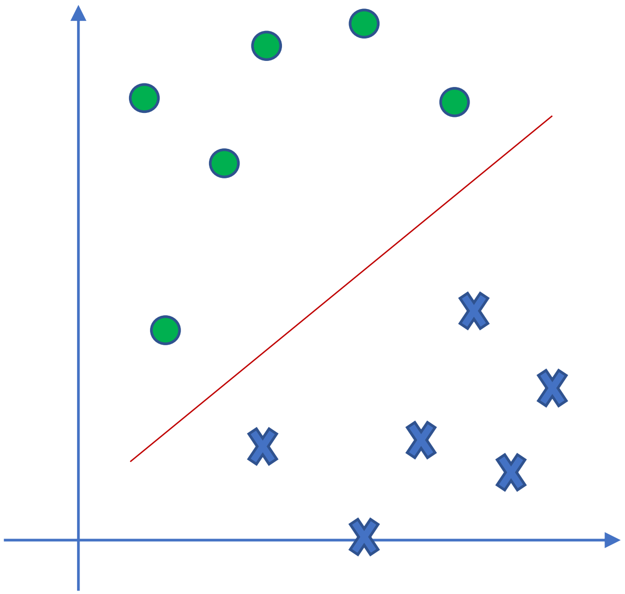
The Concept of Margins

- **Let g(x) = 0 be a decision plane**
  - The **margin** of a sample **x** (w.r.t. the decision plane) is the distance from **x** to the plane.
  - For a given set of samples S, the margin (w.r.t a decision plane) is the smallest margin over all **x** in S.
- **For a given set, a classifier that gives rise to a larger margin will be better.**

Use Margins to Compare Solutions







- Max margin
- SVM

Linear Machines & SVM

Linear SVM: Linearly Separable Case

Ira A. Fulton Schools of Engineering
Arizona State University

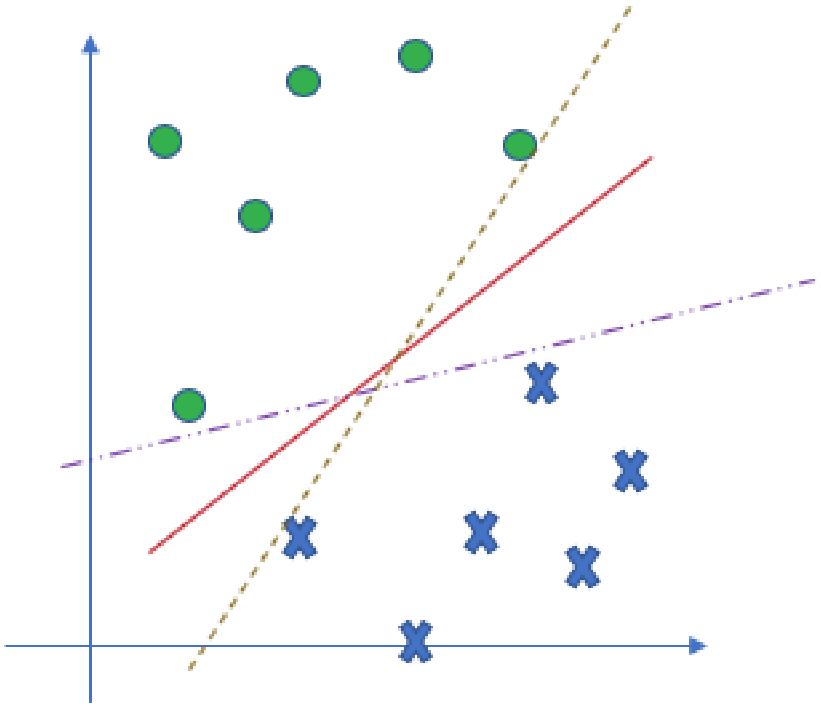
Objective

Objective

Construct SVM for Linearly Separable Data

Key Idea of Support Vector Machines

- For a given set, a classifier that gives rise to a larger margin will be better.
- SVM: To find the decision boundary such that the margin is maximized.



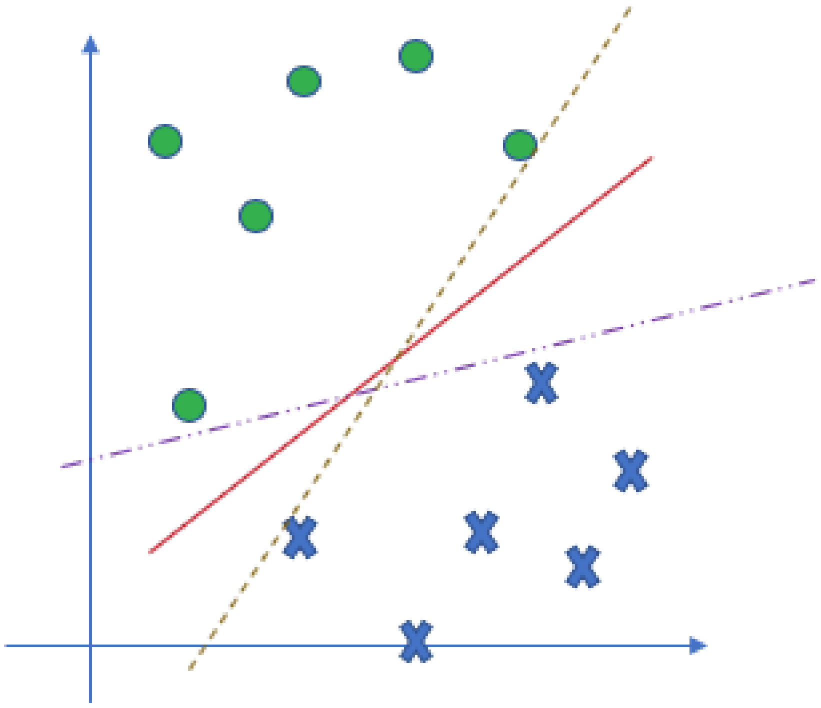
Formulating the Problem

- **Given labeled training data:**

  $<\mathbf{x}^{(i)}, y^{(i)}>, y^{(i)} \in \{-1,1\}, \mathbf{x}^{(i)} \in \boldsymbol{R}^d,$
  $i=1,\ldots,n,$

- **Assuming the points are linearly separable, let's write a separating hyperplane as:**

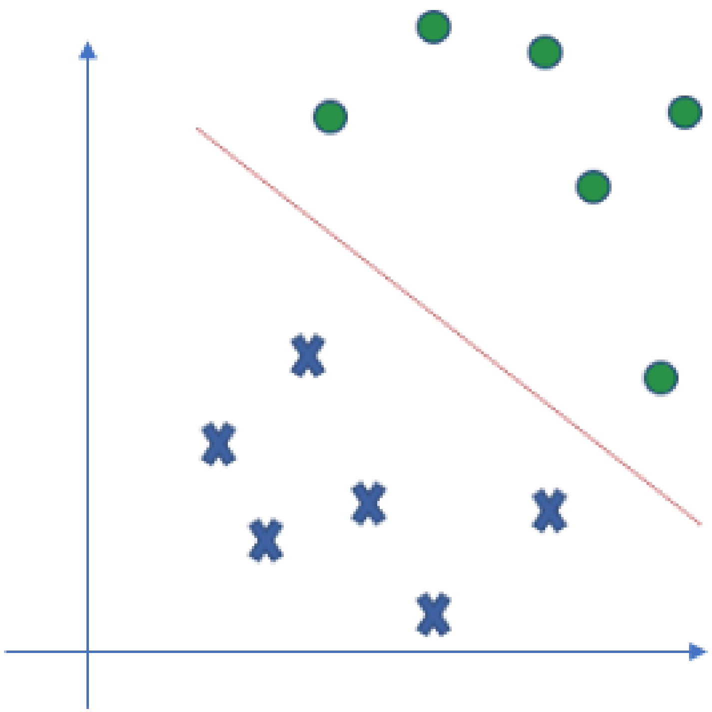
  H: $\mathbf{w}^t\mathbf{x} + b = 0$



Formulating the Problem (cont'd)

- Let d_+ (d_-) be the shortest distance from the separating hyperplane to the *closest* positive (negative) examples.
- These defines planes H_1 and H_2.
- We can let $d_+ = d_- = d$

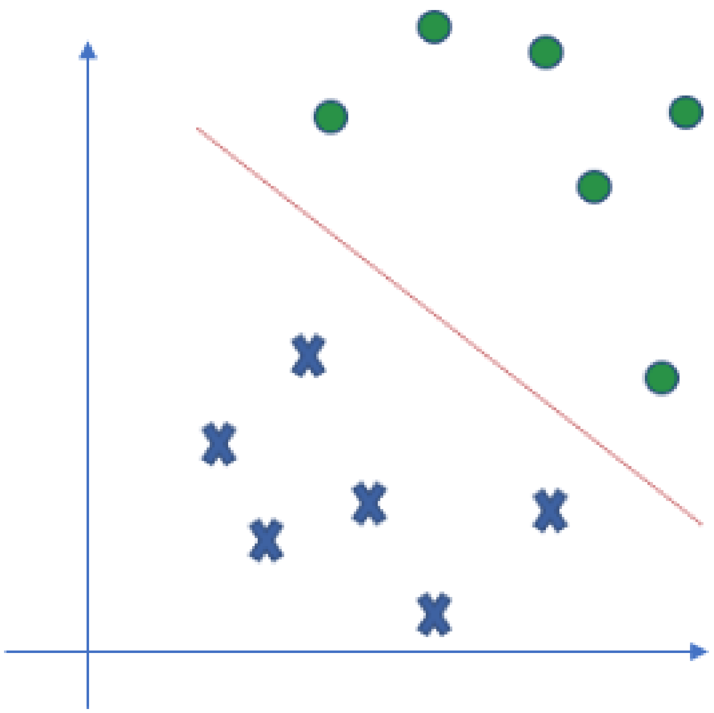
  ➔ Find a solution maximizing 2d.



Formulating the Margin

- **Given separating plane H: $w^t x + b = 0$ and distance d, what are the equations for H_1 and H_2?**

- **Consider the plane H* given by $w^t x + b = ||w||d$**
  - Check its orientation
  - Check its distance to H



Formulating the Margin (cont'd)

- H_1 is given by $\mathbf{w}^t\mathbf{x} + b = ||\mathbf{w}||d$
- Similarly, H_2 is given by $\mathbf{w}^t\mathbf{x} + b = -||\mathbf{w}||d$
- Note: for any plane equation, $\mathbf{w}^t\mathbf{x} + b = 0$, $\{\mathbf{w}, b\}$ is defined only up to an unknow scale:
  - $\{s\mathbf{w}, sb\}$ is also a valid solution to the equation, for any constant s.

Formulating the Margin (cont'd)

- We can have the canonical formulation for all the planes as

$$H: \mathbf{w}^t\mathbf{x} + b = 0$$

$$H_1: \mathbf{w}^t\mathbf{x} + b = 1$$

$$H_2: \mathbf{w}^t\mathbf{x} + b = -1$$

- The region between H_1 and H_2 is also called the margin, and its width is $\frac{2}{||\mathbf{w}||}$

Formulating SVM

$$\{\mathbf{w}^*, b^*\} = \underset{\mathbf{w},b}{\operatorname{argmin}} \|\mathbf{w}\| \;\; or \;\; \{\mathbf{w}^*, b^*\} = \underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

Subject to

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

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

The constraints can be combined into:

$$y^{(i)}(\mathbf{w}^t\mathbf{x}^{(i)} + b) - 1 \geq 0 \quad \forall i$$

➔ A nonlinear (quadratic) optimization problem with linear inequality constraints.

How to solve SVM? (Outline)

- **Reformulate the problem using Lagrange multipliers *α***
  - Lagrangian Primal Problem
  - Lagrangian Dual Problem

- **The Karush-Kuhn-Tucker Conditions**
  - *Necessary and sufficient* for $\mathbf{w}$, b, α.
  - Solving the SVM problem ➔ finding a solution to the KKT conditions.

SVM: Lagrangian Primal Formulation

Define

$$L_P(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_i \alpha_i [y^{(i)} (\mathbf{w}^t \mathbf{x}^{(i)} + b) - 1]$$

then the SVM solution should satisfy

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0, \qquad \frac{\partial L_P}{\partial \mathrm{b}} = 0,$$

$$\alpha_i \geq 0,$$

$$\alpha_i [y^{(i)} (\mathbf{w}^t \mathbf{x}^{(i)} + b) - 1]=0$$

$\Rightarrow$

The final w is given by

$$\mathbf{w} = \sum_i \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

and b is given by

$$y^{(k)} - \mathbf{w}^t \mathbf{x}^{(k)}$$

for any k such that $\alpha_k > 0$

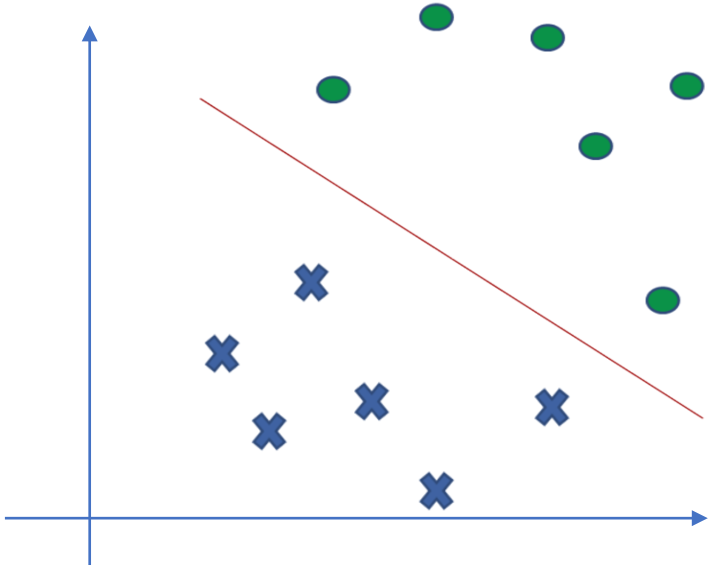
SVM: Lagrangian Dual Formulation

- The objective function is

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

- The solution is the same as before. But there is an important observation.

- Points for which $\alpha_i > 0$ are called support vectors



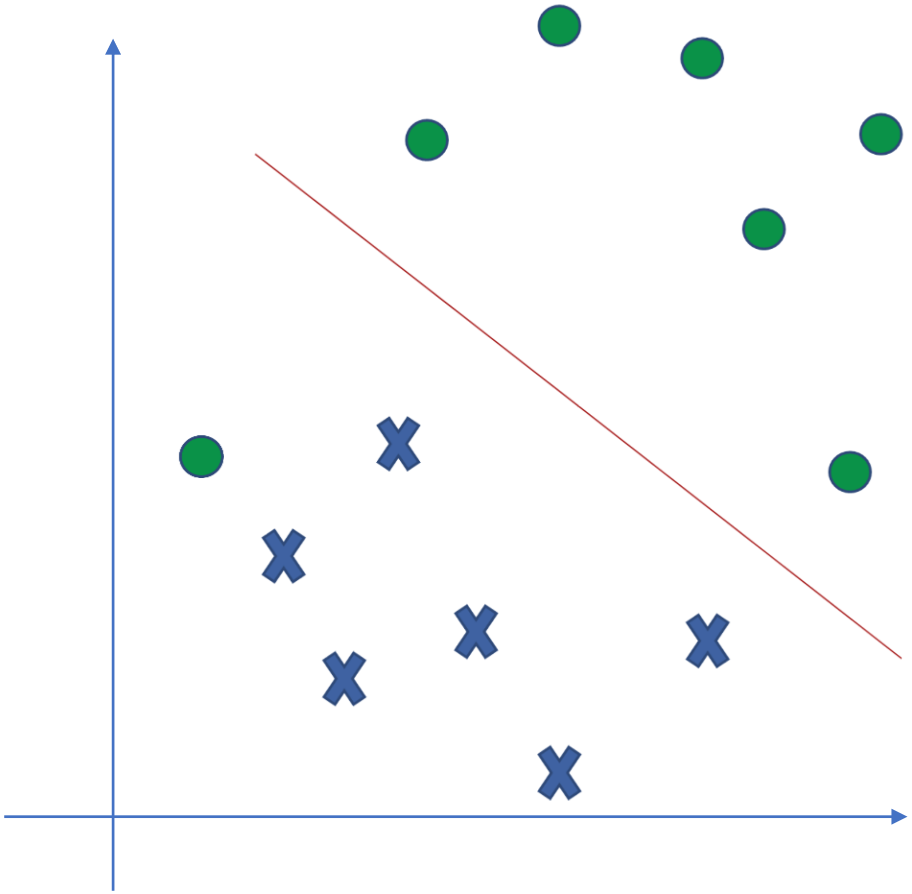
Linear Machines & SVM

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 {w,*b*} is used.

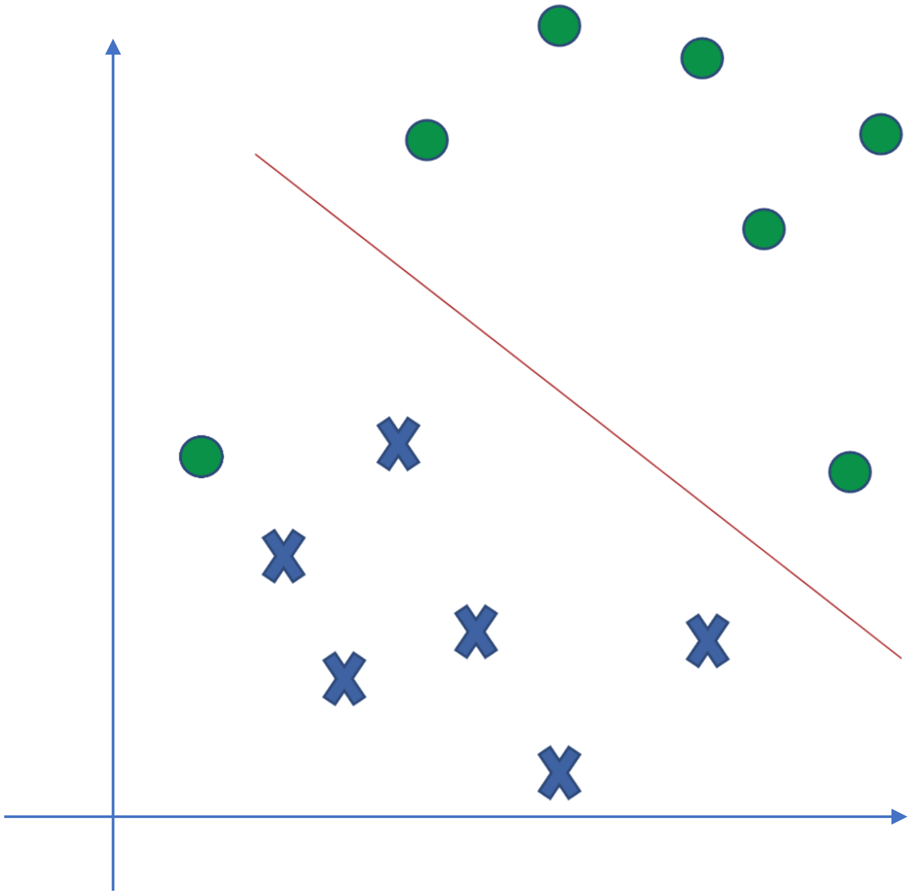


Examining Misclassified Samples

- **They will violate the constraints:**

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

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



Relaxing the Constraints

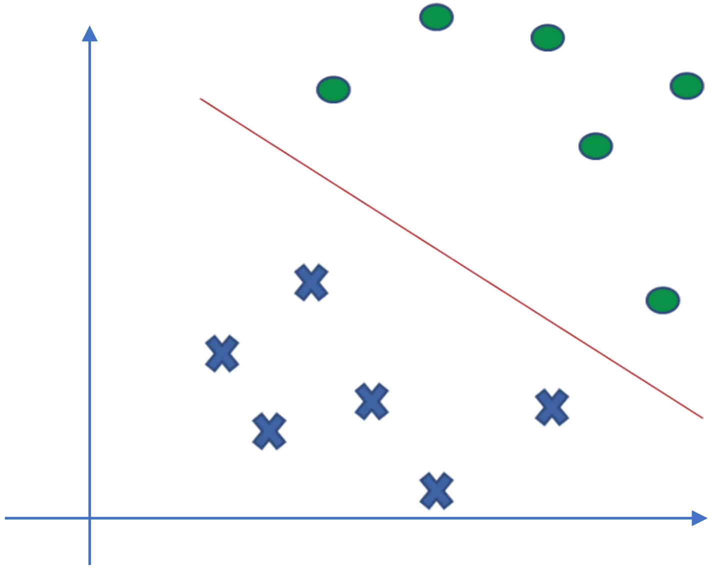
- **Introducing *non-negative* slack variables ξ_i**

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

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

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

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

Subject to

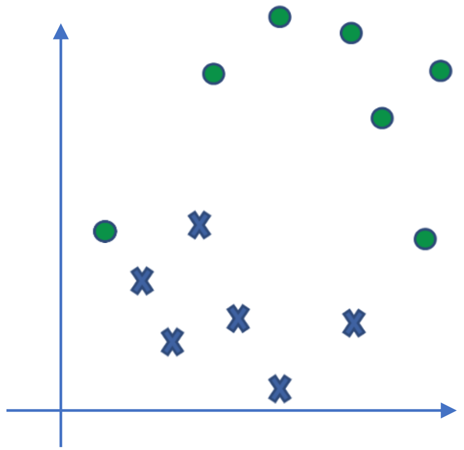
$$w^t x^{(i)} + b \geq 1 - \xi_i \quad \text{for } y^{(i)} = +1$$

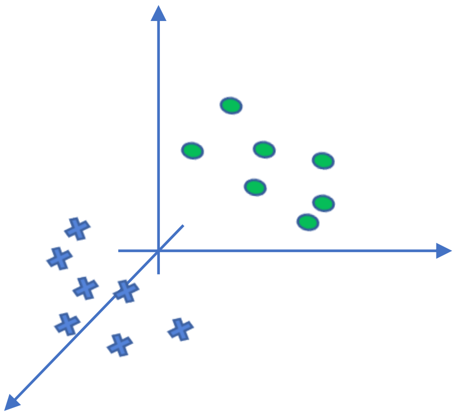
$$w^t x^{(i)} + b \leq -1 + \xi_i \quad \text{for } y^{(i)} = -1$$

$$\xi_i \geq 0, \forall i$$

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**

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