Statistical Methods in Artificial Intelligence
CSE471 - Monsoon 2016 : Lecture 03

Avinash Sharma

CVIT, IIIT Hyderabad

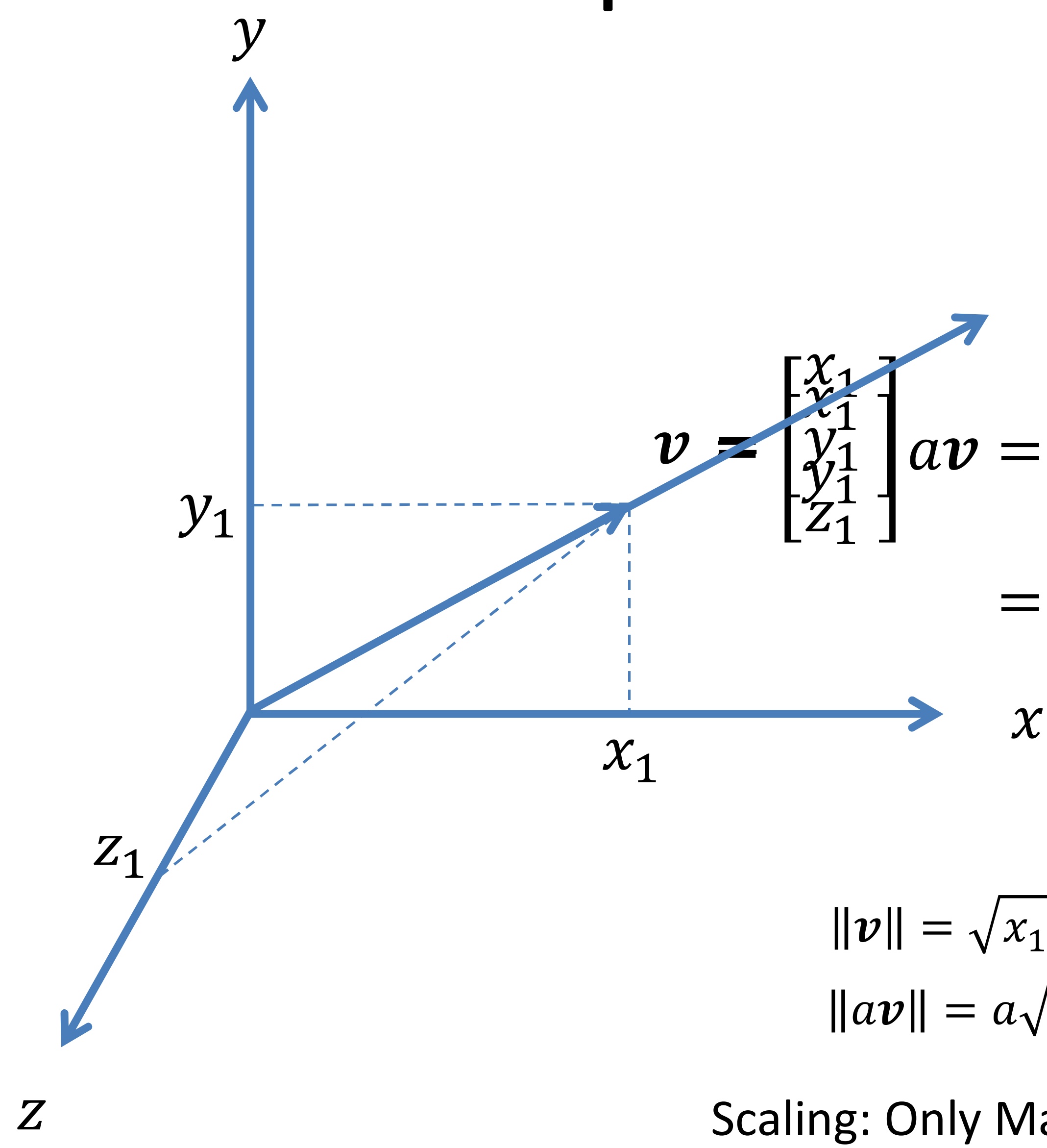
Lecture 03: Plan

- Basic Vector Operations
- Linear Discriminant Functions (LDFs)
- The Perceptron
- Generalized LDFs
- The Two-Category Linearly Separable Case
- Next Class
  - Learning LDF: Basic Gradient Descend
  - Perceptron Criterion Function

Basic Vector Operations

- Vector
- Vector Operations
  - Scaling
  - Transpose
  - Addition
  - Subtraction
  - **Dot Product**
- Equation of a Plane

Vector Operations



$$\boldsymbol{v} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad a\boldsymbol{v} = \begin{bmatrix} ax_1 \\ ay_1 \\ az_1 \end{bmatrix}$$

$$= [ax_1 \quad ay_1 \quad az_1]^T$$

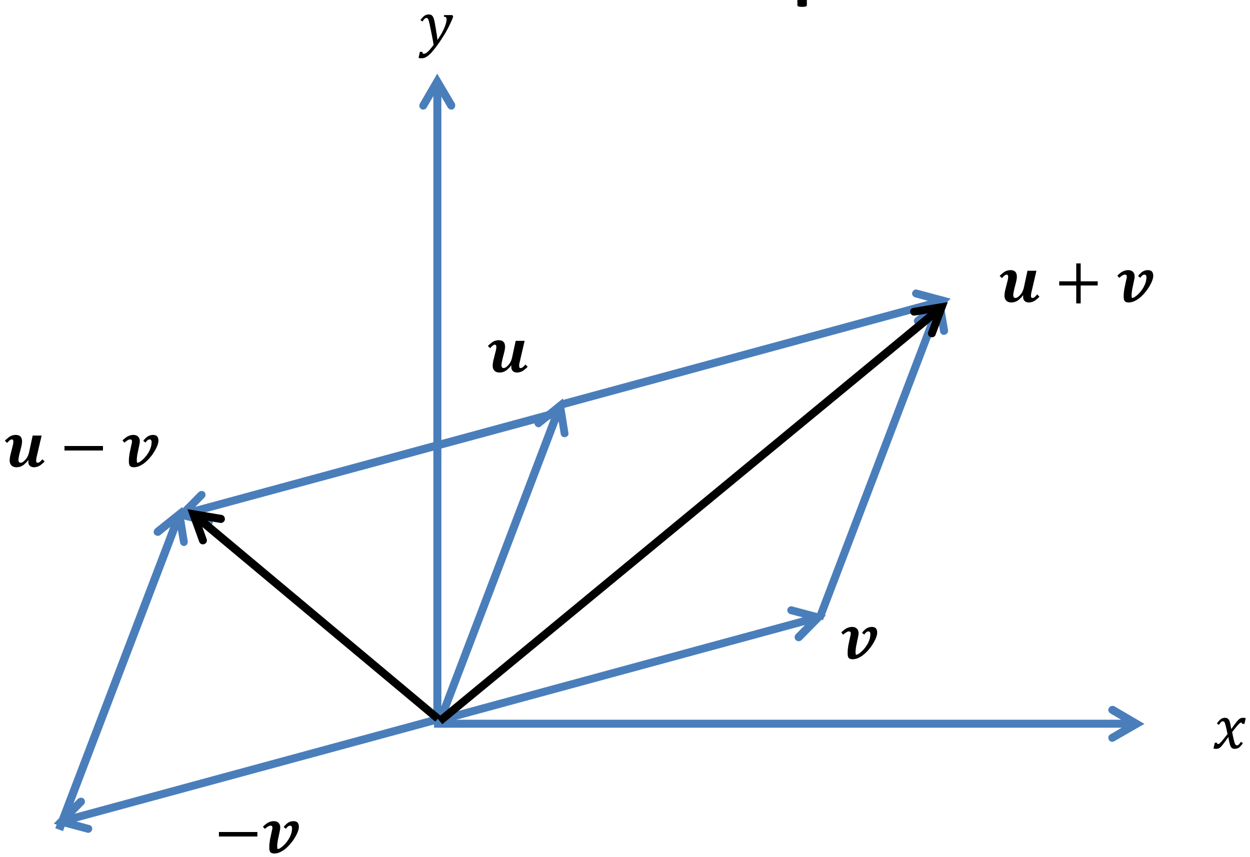
Transpose

$$\|\boldsymbol{v}\| = \sqrt{{x_1}^2 + {y_1}^2 + {z_1}^2}$$

$$\|a\boldsymbol{v}\| = a\sqrt{{x_1}^2 + {y_1}^2 + {z_1}^2}$$

Scaling: Only Magnitude Changes

Vector Operations



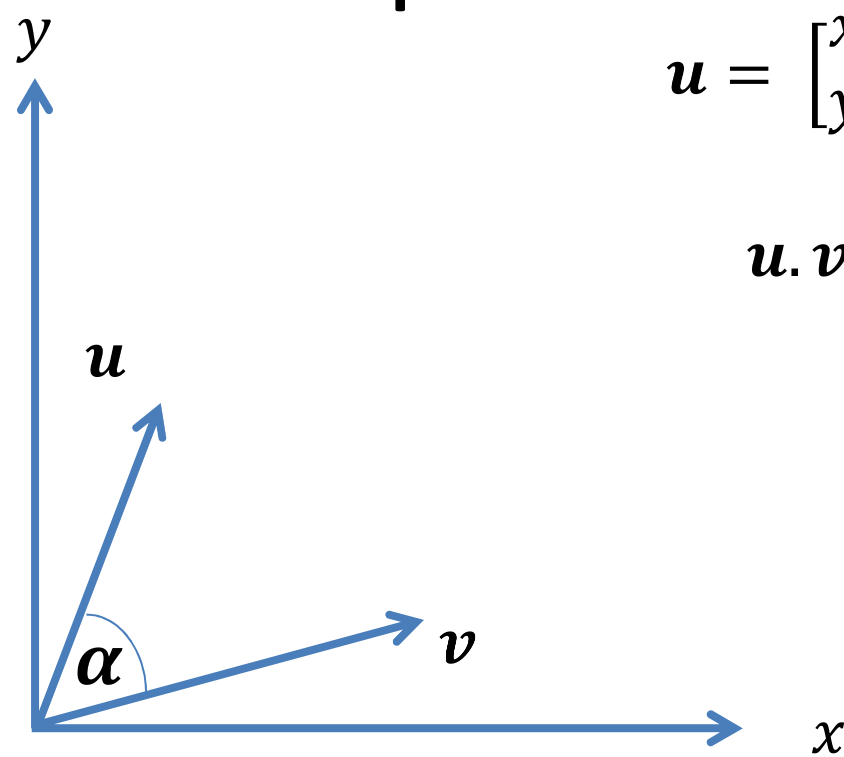
$$\boldsymbol{u} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$$

$$\boldsymbol{v} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\boldsymbol{u}+\boldsymbol{v} = \begin{bmatrix} x_1 + x_2 \\ y_1 + y_2 \end{bmatrix}$$

$$\boldsymbol{u}-\boldsymbol{v} = \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \end{bmatrix}$$

Vector Operations



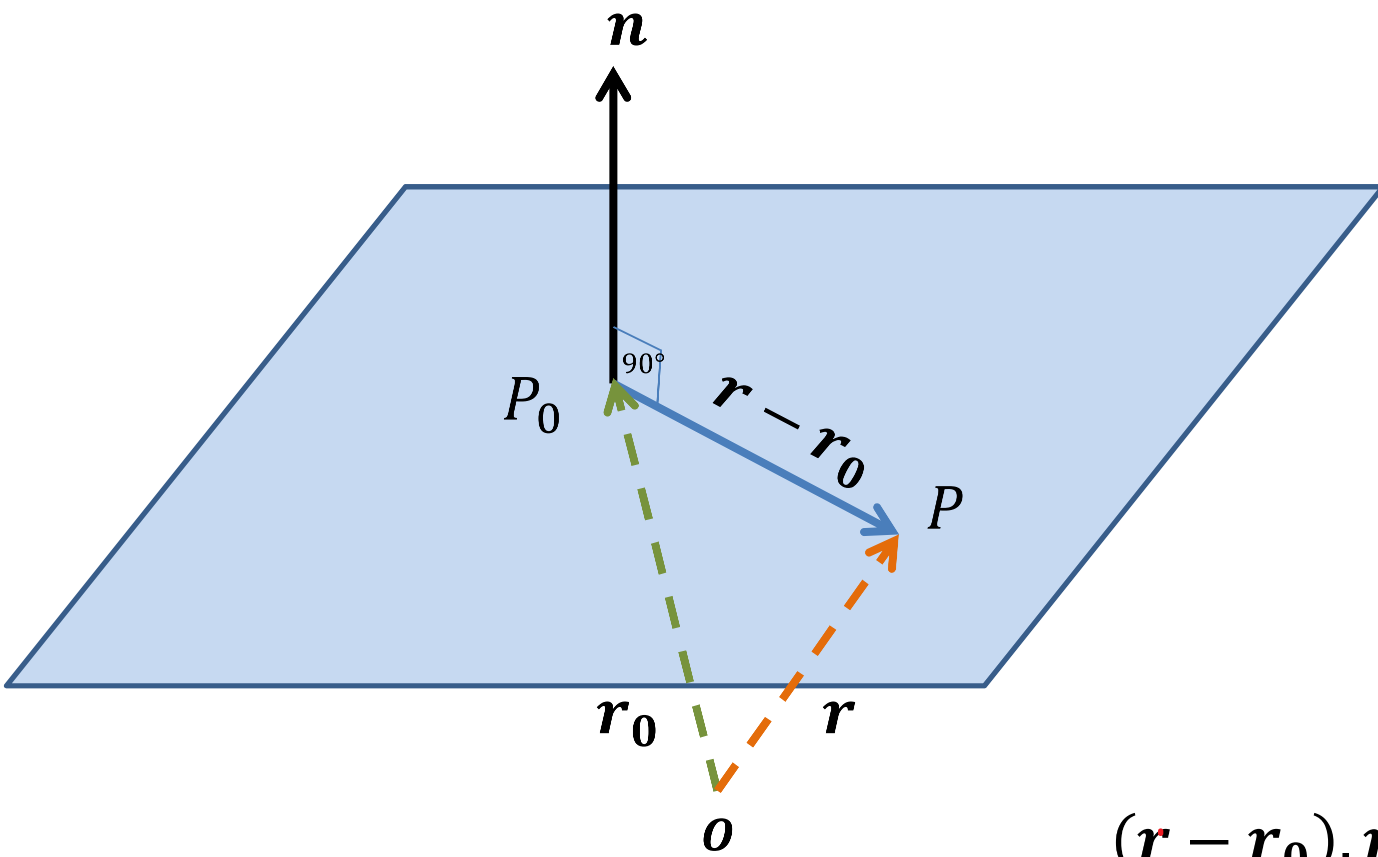
$$\boldsymbol{u} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad \boldsymbol{v} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\frac{\boldsymbol{u}^T \boldsymbol{v}}{\|\boldsymbol{u}\| \, \|\boldsymbol{v}\|} = \cos \boldsymbol{\alpha}$$

$$\begin{aligned} \boldsymbol{u}.\boldsymbol{v} &= x_1 x_2 + y_1 y_2 \\ &= \boldsymbol{u}^T \boldsymbol{v} \\ &= \|\boldsymbol{u}\| \, \|\boldsymbol{v}\| \cos \boldsymbol{\alpha} \end{aligned}$$

- Dot Product (*Inner Product*) of two vectors is a ***scalar***.
- Dot product if two perpendicular vectors is 0

Equation of a Plane



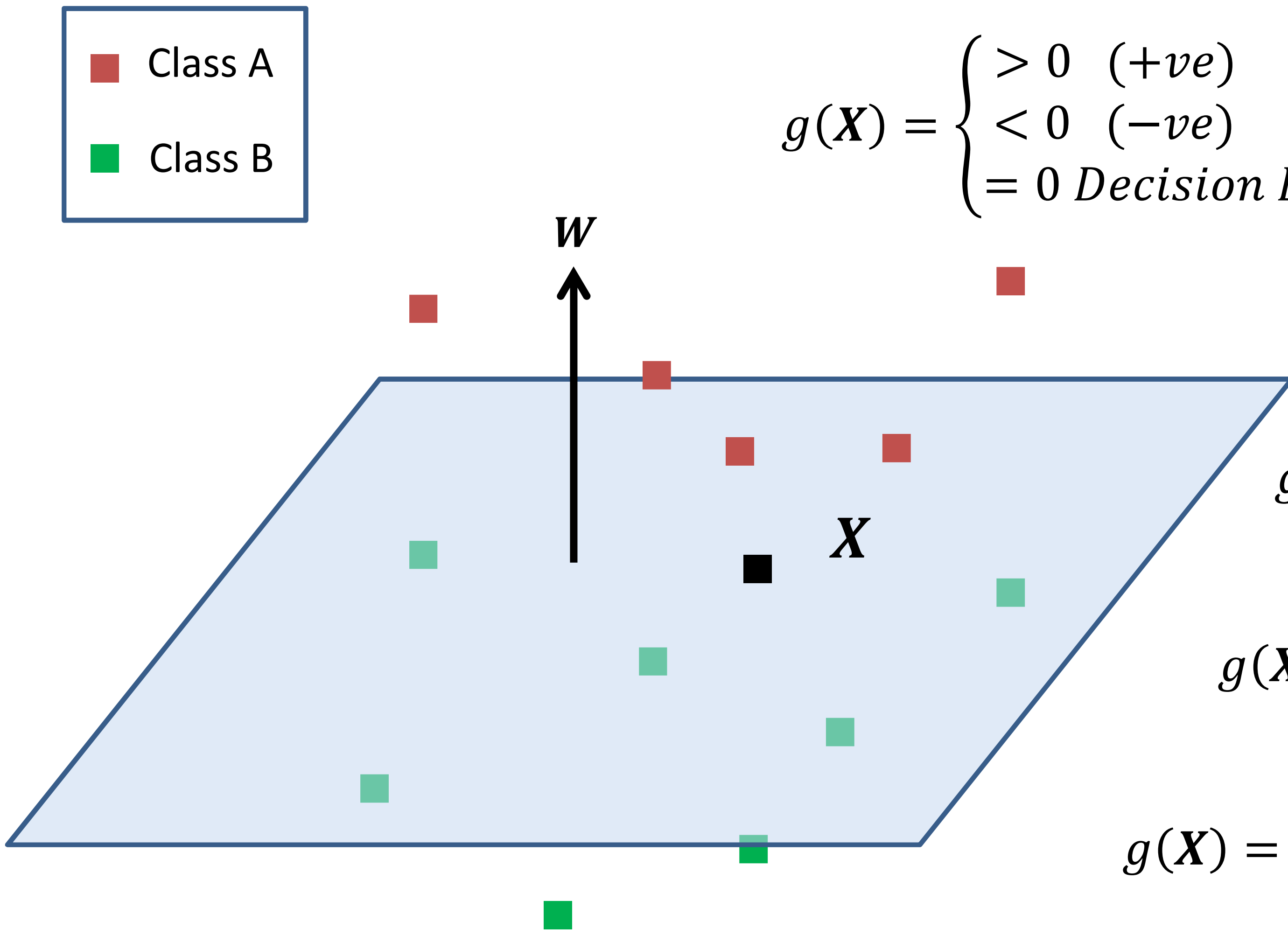
$$(\boldsymbol{r}-\boldsymbol{r_0}).\boldsymbol{n}=\boldsymbol{0}$$

Linear Discriminant Functions

- Assumes a 2-class classification setup
- Decision boundary is represented explicitly in terms of components of $\boldsymbol{X}$.
- Aim is to seek parameters of a linear discriminant function which minimize the training error.
- Why Linear ?
  - Simplest possible
  - Generalized

Linear Discriminant Functions

$$g(\boldsymbol{X}) = \begin{cases} > 0 \quad (+ve) \qquad class\ A \\ < 0 \quad (-ve) \qquad class\ B \\ = 0 \ Decision\ Boundary \end{cases}$$



$$g(\boldsymbol{X}) = 0$$

$$g(\boldsymbol{X}) = \boldsymbol{W}^T\boldsymbol{X}$$

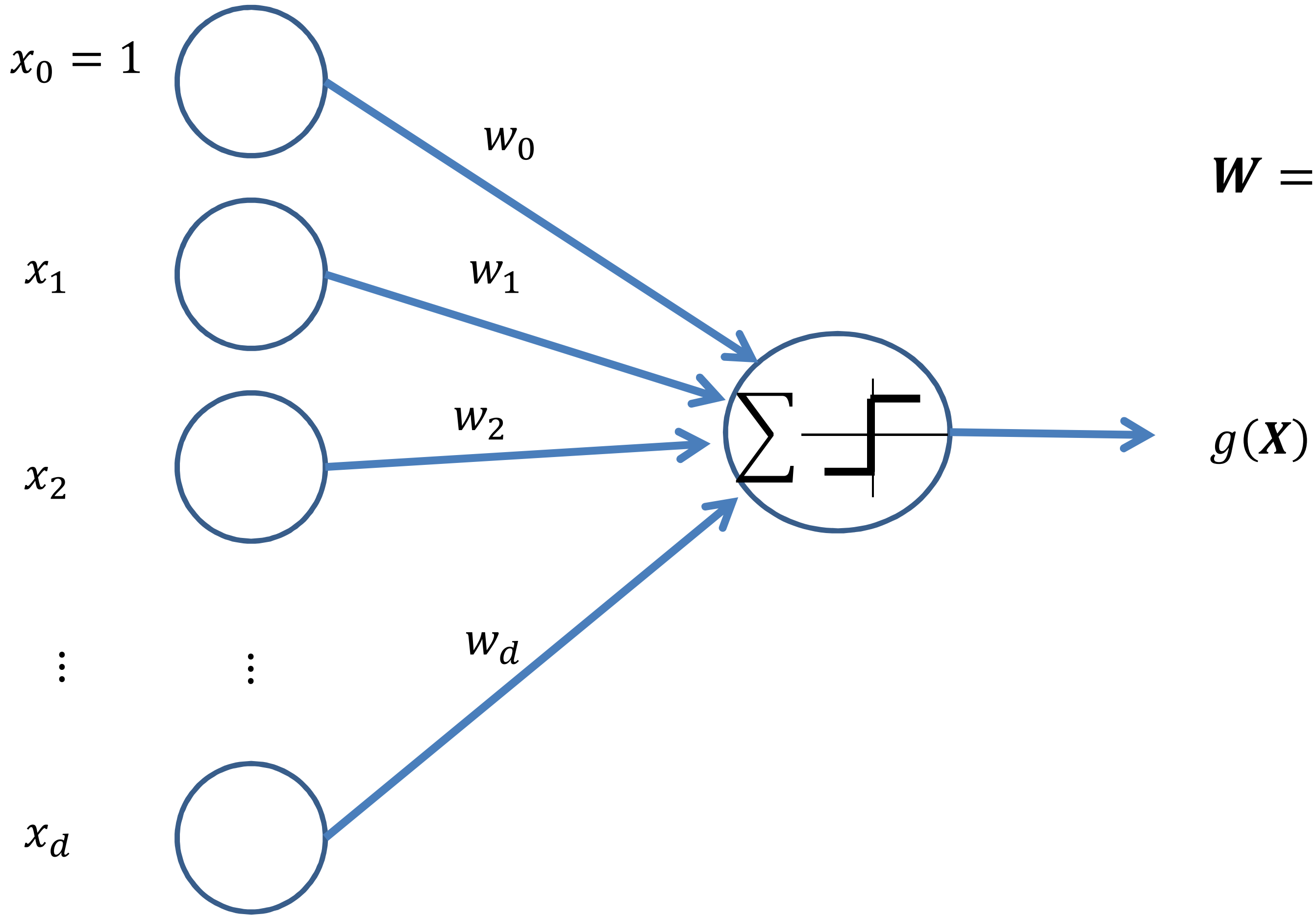
$$g(\boldsymbol{X}) = \boldsymbol{W}^T\boldsymbol{X} + w_0$$

The perceptron

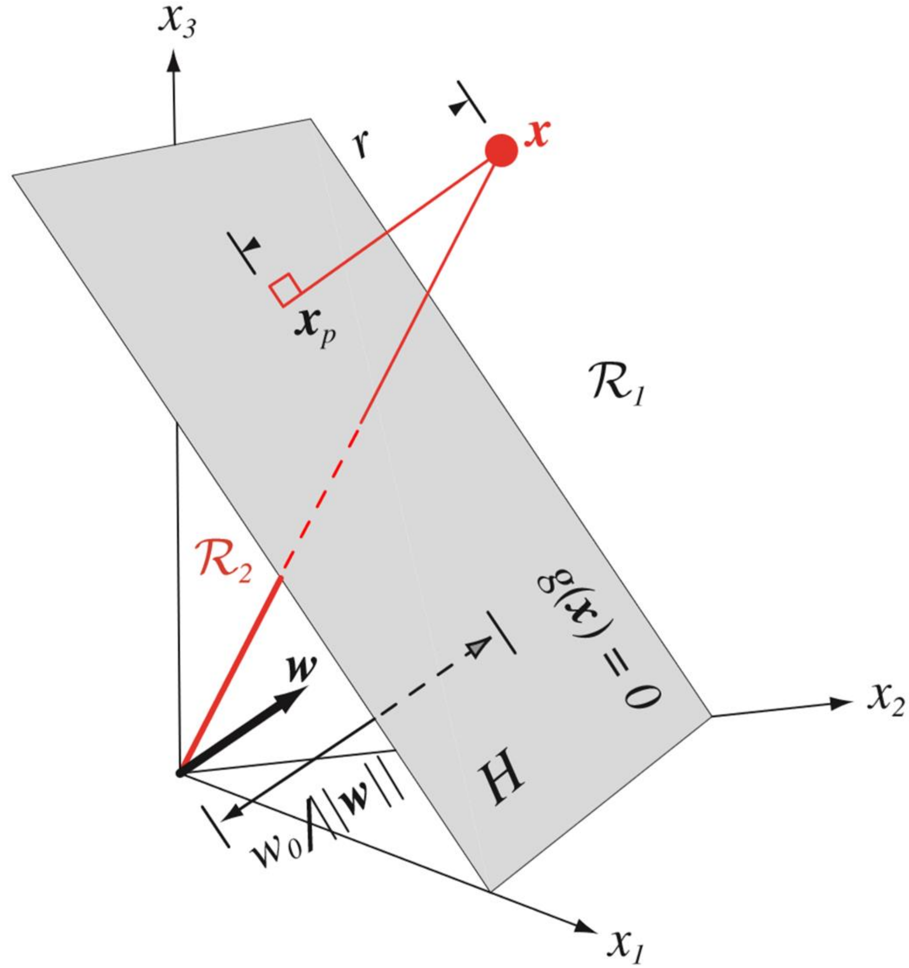
$$g(\boldsymbol{X}) = \boldsymbol{W}^T\boldsymbol{X} + w_0$$

$$\boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

$$\boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$



Perceptron Decision Boundary



Perceptron Summary

- Decision boundary surface (hyperplane) $g(\boldsymbol{X}) = 0$ divides feature space into two regions.
- Orientation of the boundary surface is decided by the normal vector $\boldsymbol{w}$.
- Location of the boundary surface is determined by the bias term w_0 .
- $g(\boldsymbol{X})$ is proportional to distance of $\boldsymbol{X}$ from the boundary surface.
- $g(\boldsymbol{X}) > 0$ positive side and $g(\boldsymbol{X}) < 0$ negative side.

Generalized LDFs

- Linear:

$$g(\boldsymbol{X}) = w_0 + \boldsymbol{W}^T \boldsymbol{X} \qquad \boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$g(\boldsymbol{X}) = w_0 + \sum_{i=1}^{d} w_i x_i$$

- Non Linear

$$g(\boldsymbol{X}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=1}^{d} w_{ij} x_i x_j$$

(Quadratic)

Generalized LDFs

- Linear

$$g(\boldsymbol{X}) = w_0 + \boldsymbol{W}^T\boldsymbol{X} \qquad \boldsymbol{X} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix} \quad \boldsymbol{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$g(\boldsymbol{X}) = \sum_{i=0}^{d} w_i x_i \qquad x_0{=}1$$

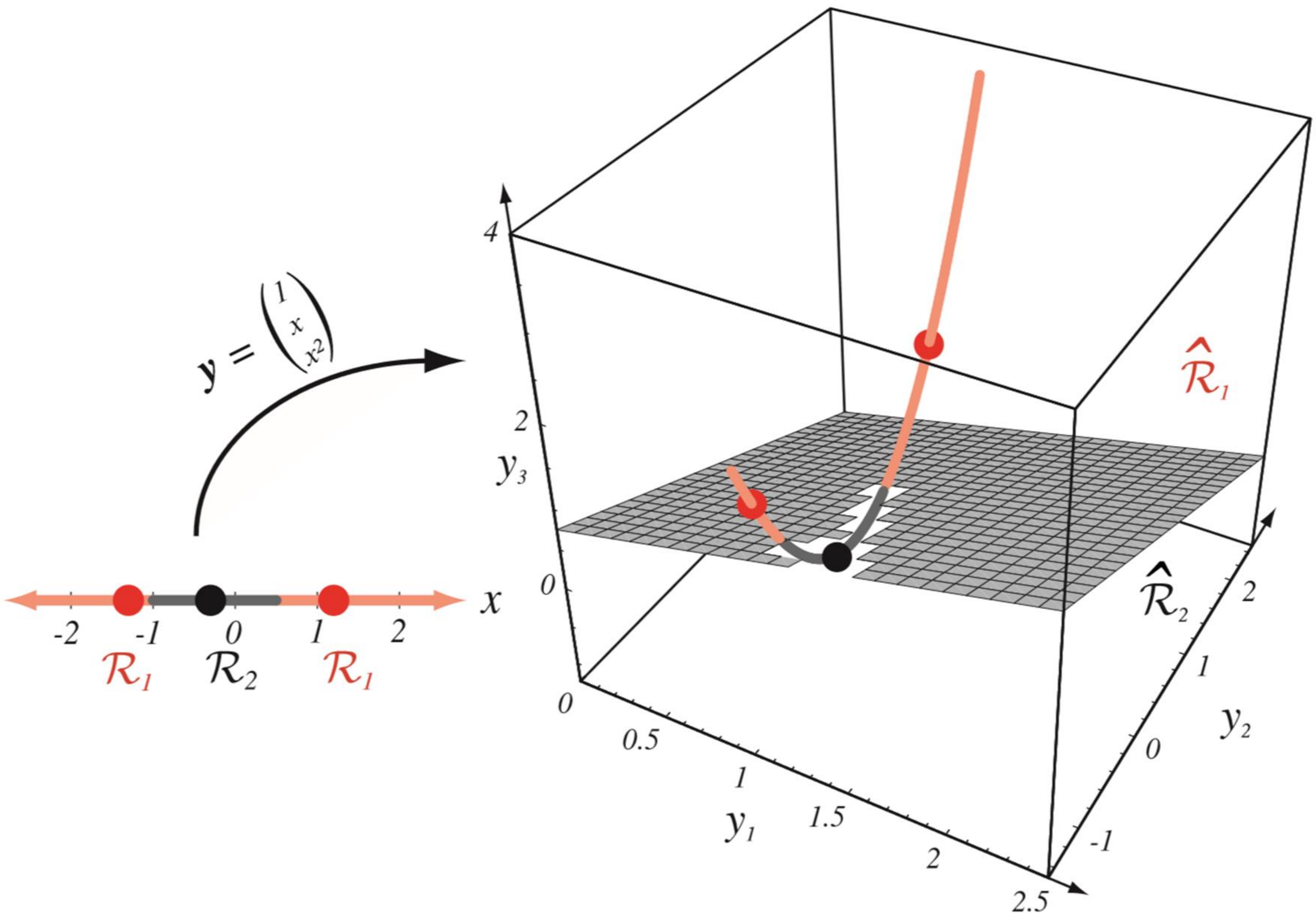
$$g(\boldsymbol{X}) = \boldsymbol{a}^T\boldsymbol{Y} \qquad \boldsymbol{Y} = \begin{bmatrix} x_0 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} x_0 \\ \boldsymbol{X} \end{bmatrix} \quad \boldsymbol{a} = \begin{bmatrix} w_0 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \boldsymbol{W} \end{bmatrix}$$

- Non Linear

$$\boldsymbol{Y} = \varphi(\boldsymbol{X})$$

$$g(\boldsymbol{X}) = \boldsymbol{a}^T\boldsymbol{Y} = \sum_{i=1}^{\hat{d}} a_i y_i \qquad \boldsymbol{a} = \begin{bmatrix} a_1 \\ \vdots \\ a_{\hat{d}} \end{bmatrix}$$

Generalized LDFs

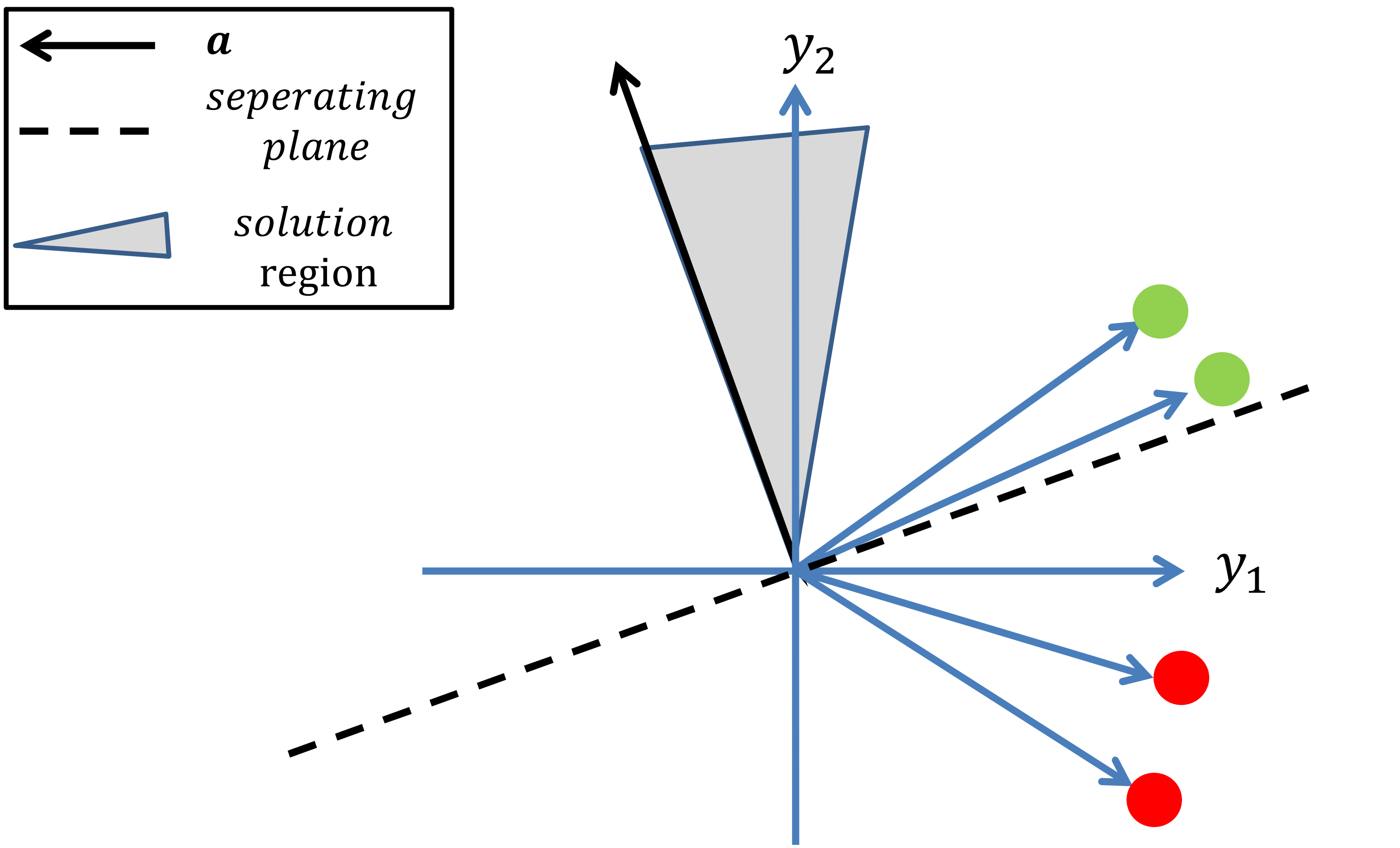


Generalized LDFs Summary

- φ can be any arbitrary mapping function that projects original data points $\boldsymbol{X} \in \mathbb{R}^{d}$ to points $\boldsymbol{Y} \in \mathbb{R}^{\hat{d}}$ where $\hat{d} \gg d$.
- The hyperplane decision surface $\widehat{H}$ passes through origin.
- **Advantage**: In the mapped higher dimensional space data might be linear separable.
- **Disadvantage**: The mapping is computationally intensive and learning the classification parameters can be non-trivial (Curse of Dimensionality).

Two-Category Linearly Separable Case

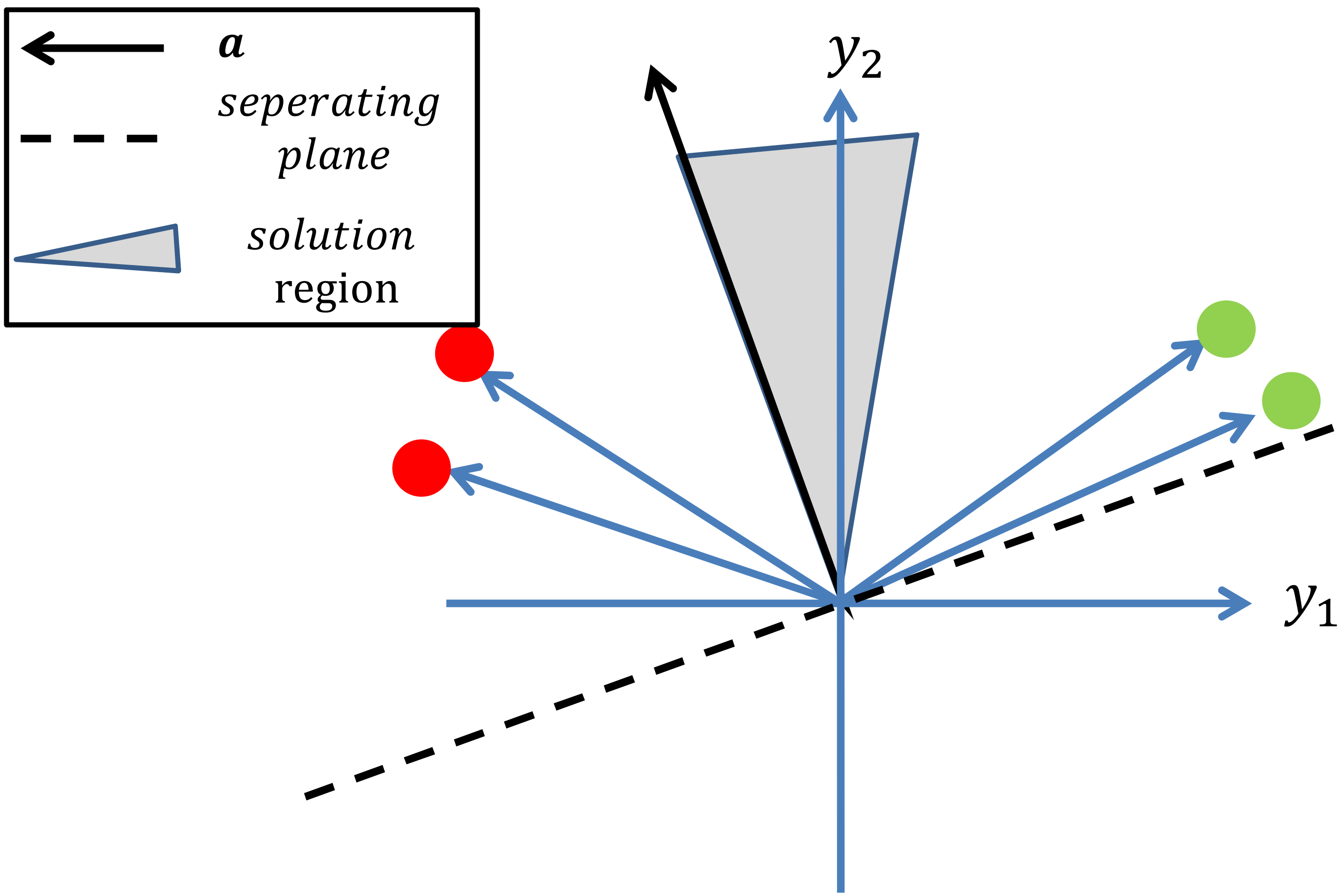
$$g(\boldsymbol{X}) = \boldsymbol{a}^T\boldsymbol{Y} = \sum_{i=1}^{\hat{d}} a_i y_i \quad = \begin{cases} > 0 \quad (+ve) \qquad class\ A \\ < 0 \quad (-ve) \qquad class\ B \\ = 0\ Decision\ Boundary \end{cases}$$



Two-Category Linearly Separable Case

$$g(\boldsymbol{X}) = \boldsymbol{a}^T \boldsymbol{Y} = \sum_{i=1}^{\hat{d}} a_i y_i > 0$$

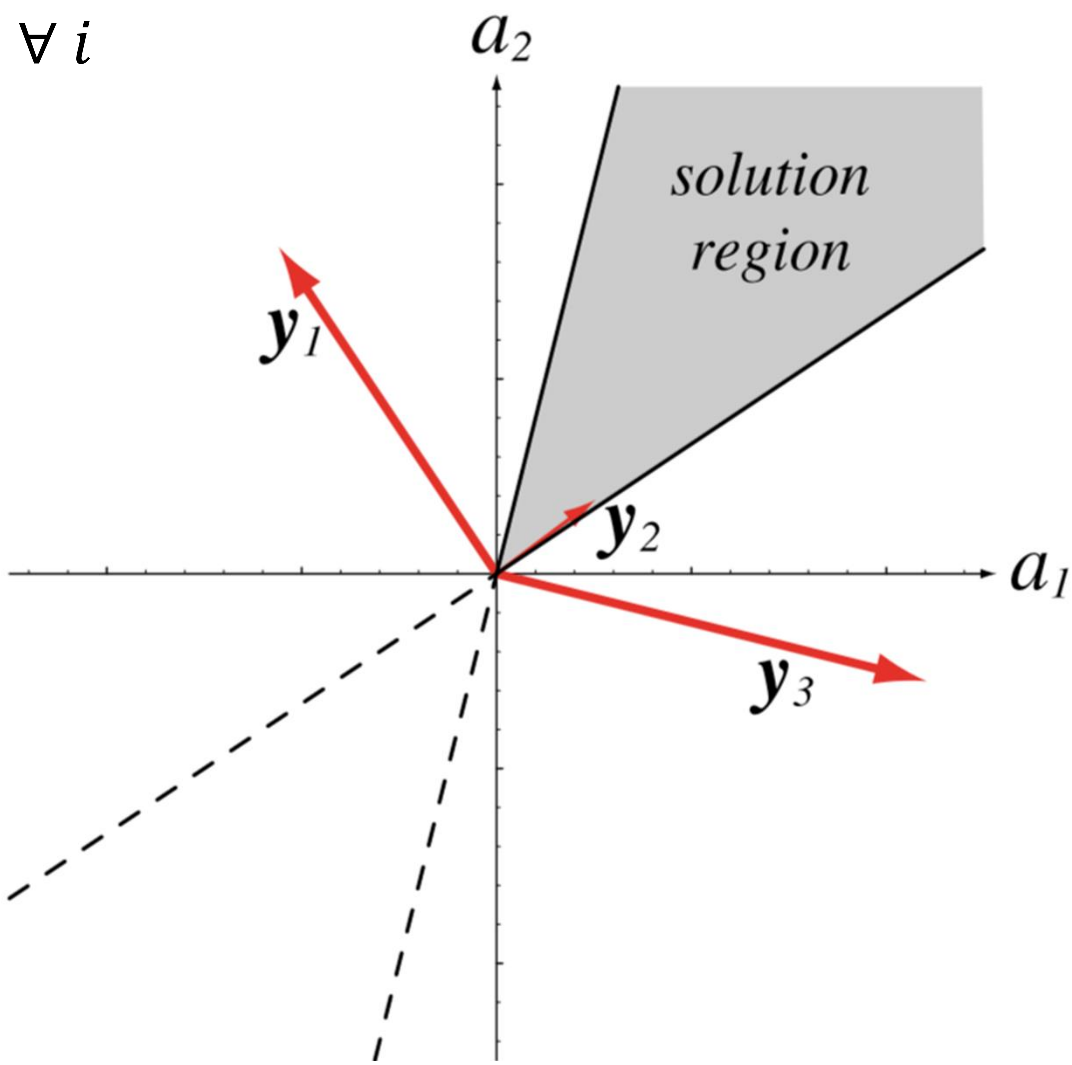
Normalized Case



Two-Category Linearly Separable Case

$$\mathbf{a}^T \mathbf{y}_i > 0 \quad \forall\, i$$

Data vector



Two-Category Linearly Separable Case

$\boldsymbol{a}^T \boldsymbol{y}_i \geq b > 0 \quad \forall\, i$