Business Anaytics

M. Tech QROR – 2nd yr (2024)

Classification - SVM

(Predictive Analytics)

Dr. Prasun Das

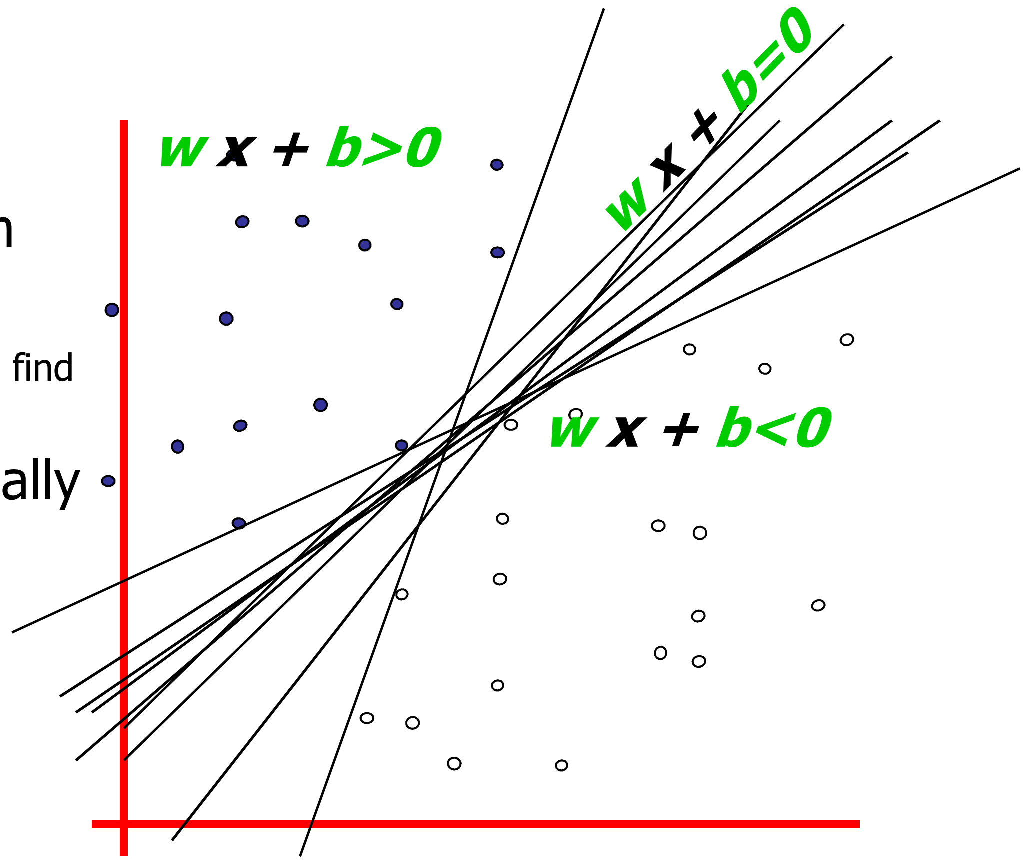
SQC & OR Unit

Indian Statistical Institute

e-mail: prasun@isical.ac.in

What is a good Decision Boundary?

- Consider a 2-class, linearly separable classification problem
- Many decision boundaries!
  - The Perceptron algorithm can be used to find such a boundary
- Are all decision boundaries equally good? – which is the best ?

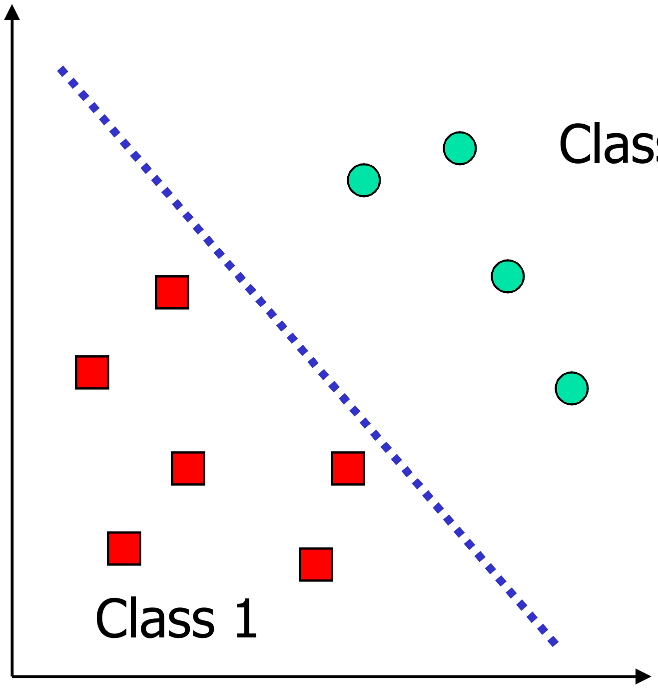


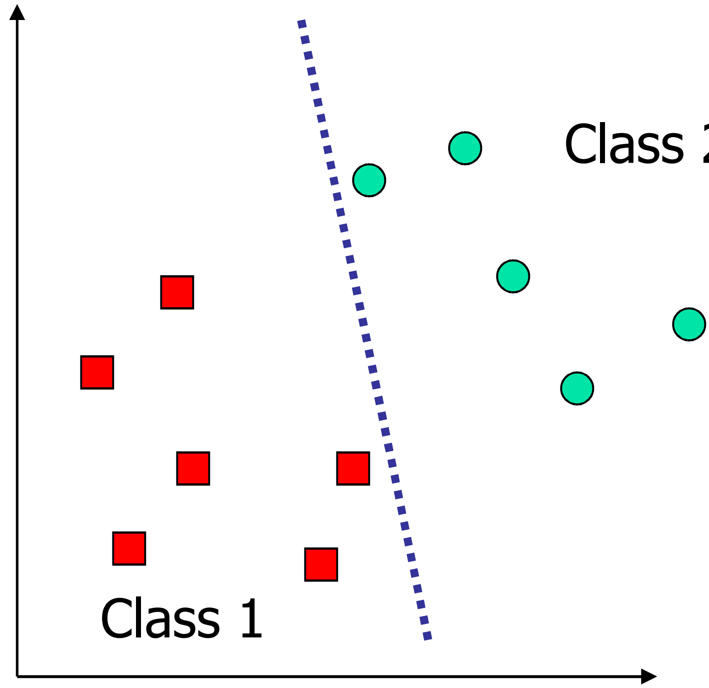
$$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}\,\mathbf{x} + b)$$

• denotes +1

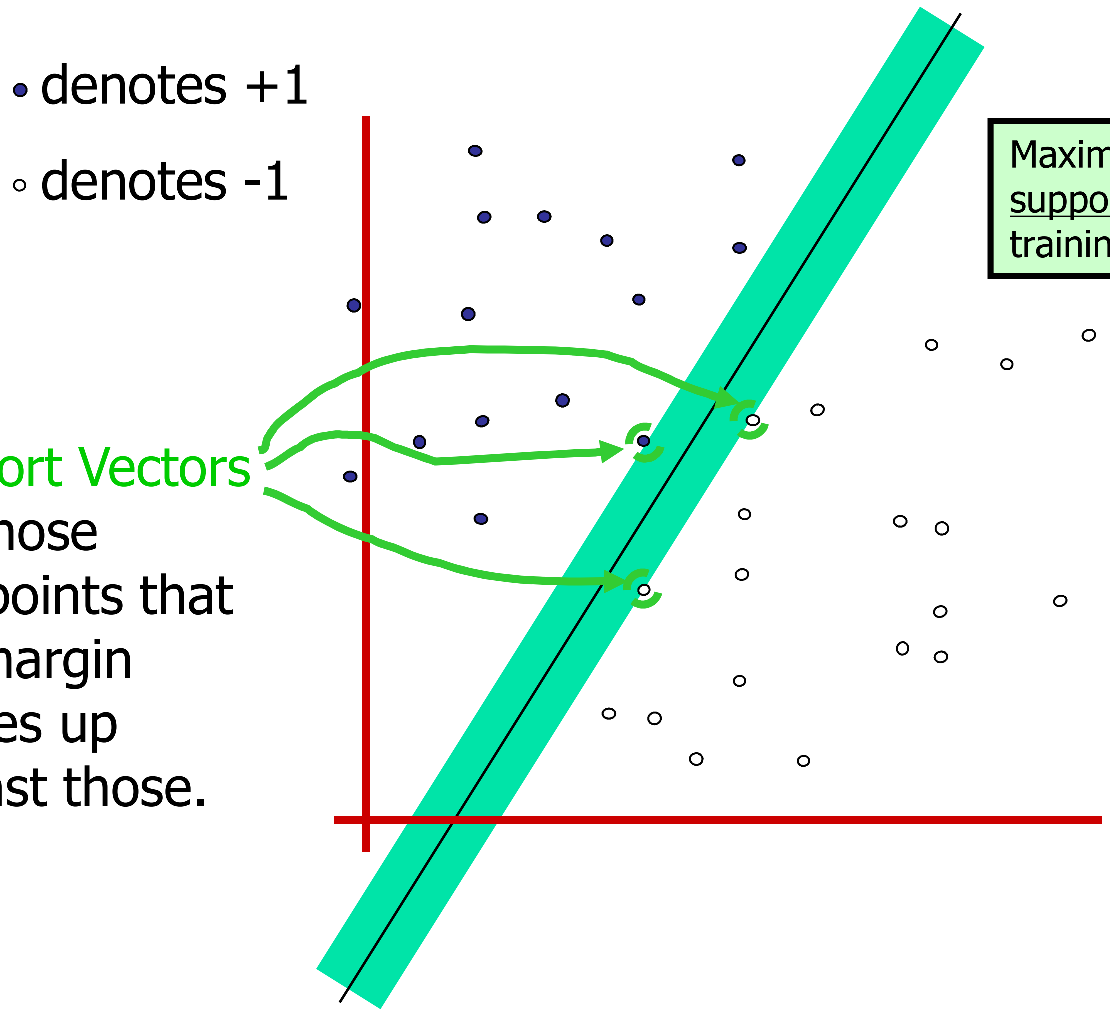
◦ denotes -1

Examples of Bad Decision Boundaries





Margin: the width that the boundary could be increased by before hitting a datapoint.



Support Vectors are those datapoints that the margin pushes up against those.

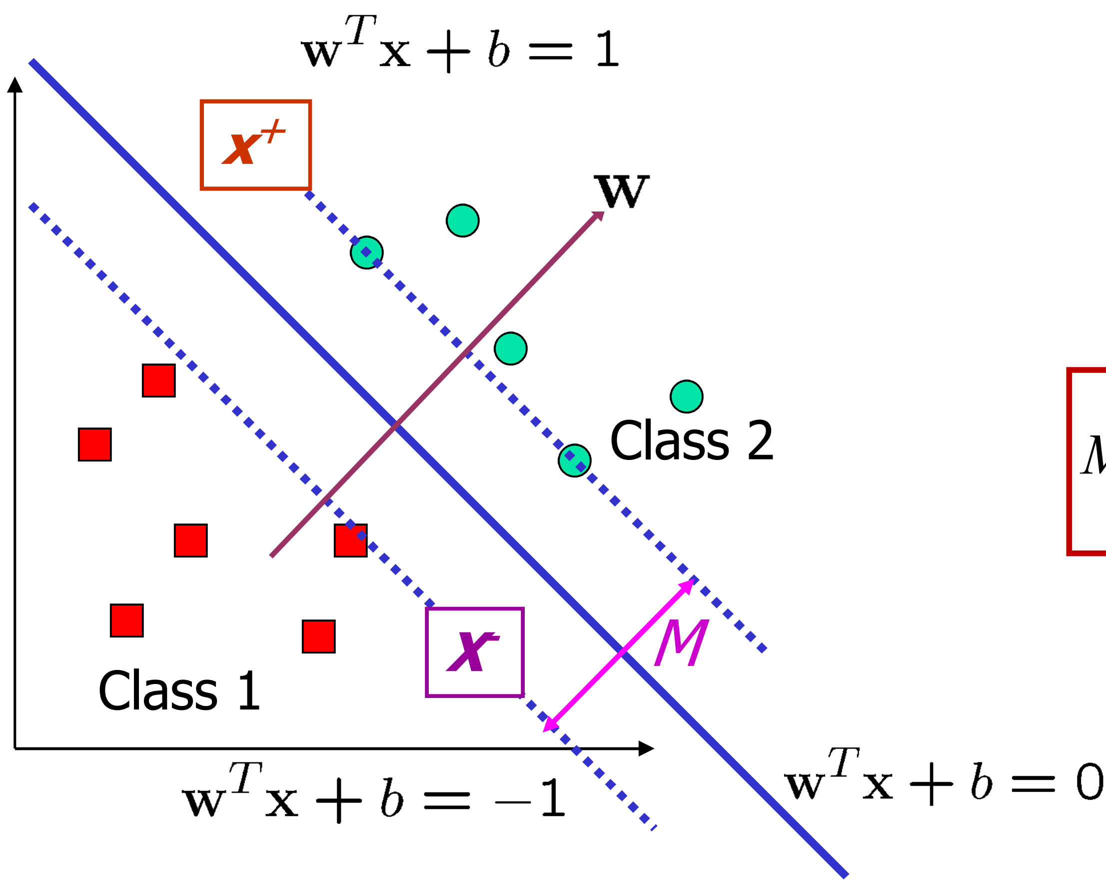
Maximizing the Margin implies that only support vectors are important; other training examples are ignorable.

The maximum margin linear classifier is the linear classifier with the maximum margin.

This is the simplest kind of SVM (Called an Linear SVM)

Large-margin Decision Boundary

- The decision boundary should be as far away from the data of both classes as possible - We should maximize the margin, *M*



What we know:

- $\boldsymbol{w} \,.\, \boldsymbol{x}^{+} + b = +1$
- $\boldsymbol{w} \,.\, \boldsymbol{x}^{-} + b = -1$
- $\boldsymbol{w} \,.\, (\boldsymbol{x}^{+} - \boldsymbol{x}^{-}) = 2$

$$\boxed{M = \frac{(x^{+} - x^{-}) \cdot w}{|w|} = \frac{2}{|w|}}$$

Linear SVM: Finding the Decision Boundary

- Goal: **1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \text{if } y_i = +1$$

$$wx_i + b \leq 1 \quad \text{if } y_i = -1$$

$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$

2) Maximize the Margin: $M = \frac{2}{|w|}$

Minimize $\frac{1}{2} w^t w$

Finding the Decision Boundary

- **Formulate a Constrained QPP Problem and solve for *w* and *b***
- Minimize $\Phi(w) = \frac{1}{2} w^t w$ or, Minimize $\frac{1}{2}||\mathbf{w}||^2$

subject to $y_i(wx_i + b) \geq 1 \quad \forall i$

  - Need to optimize a *quadratic* function subject to *linear* constraints.
  - The solution involves constructing a ***dual problem*** where a ***Lagrange multiplier*** α_i is associated with every constraint in the primary problem:

  - The Lagrangian of this optimization problem is

$$\mathcal{L} = \frac{1}{2}||\mathbf{w}||^2 - \sum_i \alpha_i \left(y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1 \right) \qquad \alpha_i \geq 0 \quad \forall i$$

The Dual Problem

- By setting the derivative of the Lagrangian to be zero, the optimization problem can be written in terms of α_i (the dual problem)

$$\max.\ W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

$$\text{subject to } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Decision variables: $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n$

- This is a quadratic programming (QP) problem- A global maximum of α_i can always be found.

- ***w*** can be recovered by

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

The Optimization Problem Solution

- The solution has the form:

$$\boxed{\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x}_i \,, \quad b = y_k - \boldsymbol{w}^T \boldsymbol{x}_k \text{ for any } \boldsymbol{x}_k \text{ such that } \alpha_k \neq 0}$$

- Many of the α_i are zero. **Each non-zero α_i indicates that corresponding x_i is a support vector (SV).** The decision boundary is determined only by the SVs.

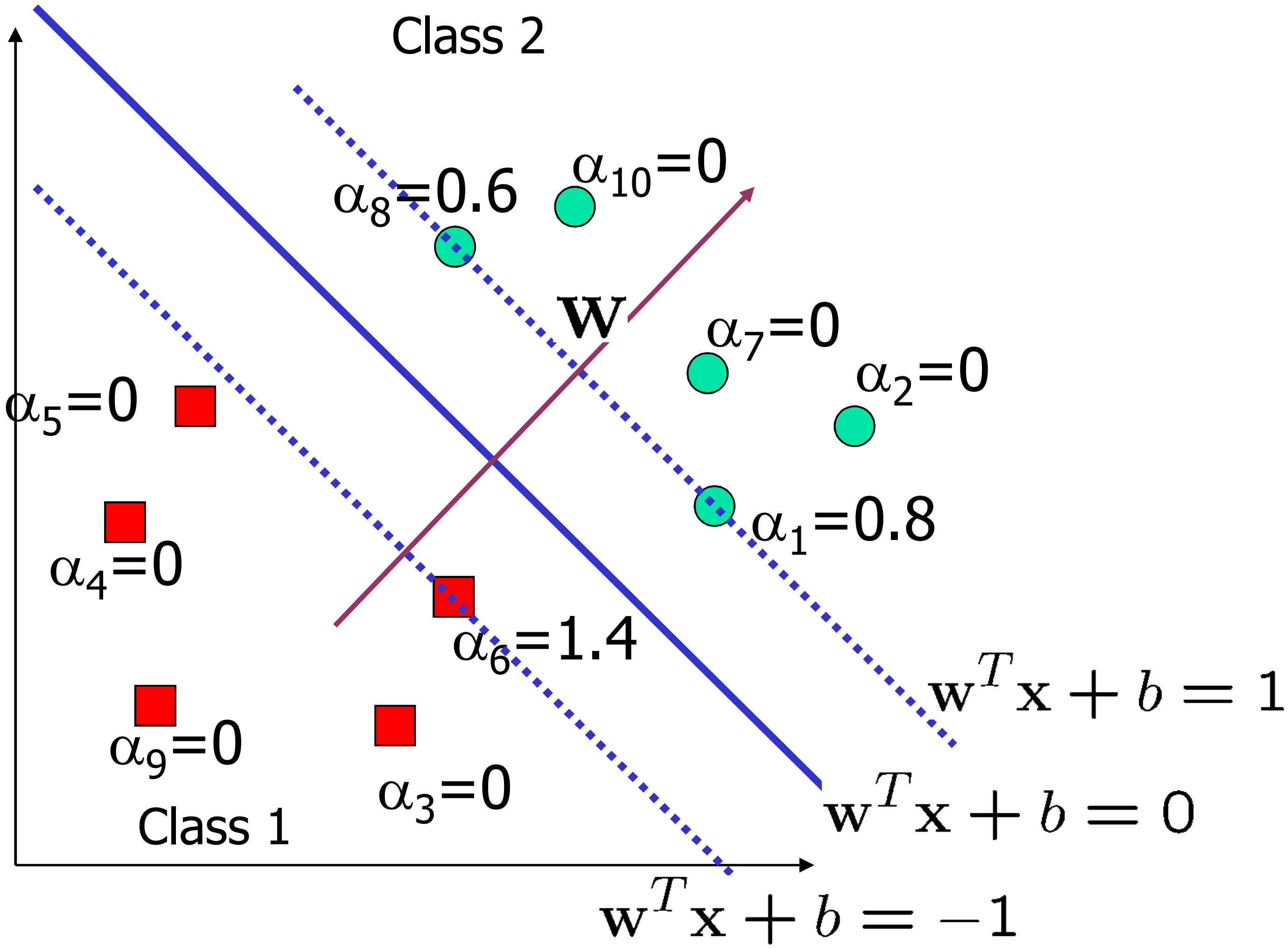
- Let t_j (j=1, ..., s) be the indices of the s support vectors. We can write

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

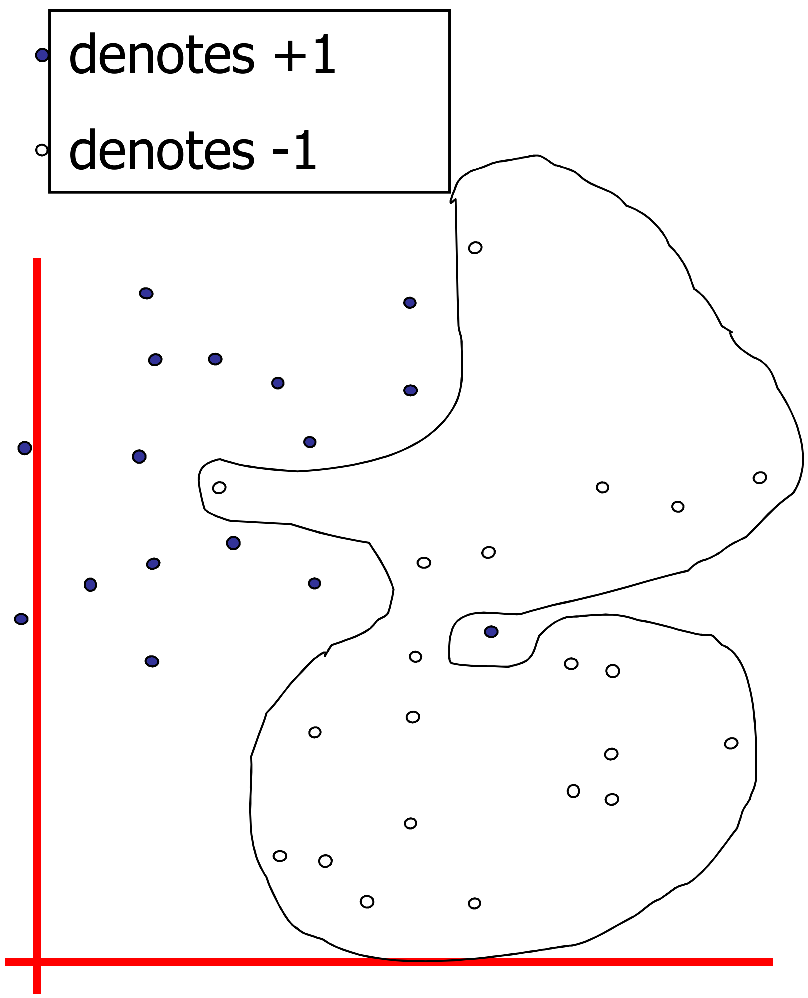
- For testing with a new datapoint $\boldsymbol{z}$, compute: $f(\boldsymbol{z})$ = $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (\mathbf{x}_{t_j}^T \mathbf{z}) + b$
- Notice that it relies on an *inner product* between the test point $\boldsymbol{z}$ and the support vectors $\boldsymbol{x}_{t_j}$.

- Classify **z** as class -1 if $f(\boldsymbol{z})$ is positive, and class-2 otherwise.

A Geometrical Interpretation



Dataset with noise



- **Hard Margin: So far we require all data points be classified correctly**

  - No training error

- **What if the training set is noisy?**

  - Solution 1: use very powerful kernels: Soft Margin Classifier

OVERFITTING!