SUPPORT VECTOR MACHINES

FA590 STATISTICAL LEARNING IN FINANCE

Dr. Zonghao Yang

Stevens Institute of Technology

Previous Lecture

- Understand the key concepts and techniques used in model selection, including cross-validation methods
- Gain insight into handling temporal data in financial modeling, particularly expanding-window and rolling-window validation approaches
- Explore hyperparameter tuning and feature selection strategies to optimize model performance

Learning Objectives

- Understand the mathematical foundations of support vector machines (SVMs), including linear classification and separating hyperplanes
- Formulate SVMs as optimization problems for both separable and non-separable cases
- Learn how to apply kernel transformations to extend SVMs for non-linear decision boundaries
- Compare SVMs with other classification methods, such as logistic regression, and understand the strengths and weaknesses of each approach

Part I

Review: Linear Algebra

Vectors and Vector Spaces

- **Vector** A vector is a quantity that has both magnitude and direction

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

  where $v_1, v_2, \ldots, v_n$ are the components of the vector.
- **Vector Space** A vector space (or linear space), $V \in \mathbb{R}^n$, is a collection of vectors that can be added together and multiplied by scalars, while satisfying the following properties:
  - Addition $\mathbf{u} + \mathbf{v} \in V$ for any $\mathbf{u}, \mathbf{v} \in V$
  - Scalar Multiplication $\alpha\mathbf{v} \in V$ for any $\alpha \in \mathbb{R}, \mathbf{v} \in V$
  - Zero Vector There exists a zero vector $\mathbf{0} \in V$ such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$ for all $\mathbf{v} \in V$

Dot Product (Inner Product)

- **Definition** The dot product of two vectors $\mathbf{u} = (u_1, u_2, \ldots, u_n)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ is defined as

$$\mathbf{u} \cdot \mathbf{v} = u_1 v_1 + u_2 v_2 + \cdots + u_n v_n$$

- **Norm** The norm (or magnitude) of a vector is given by

$$\|\mathbf{u}\| = \sqrt{\mathbf{u} \cdot \mathbf{u}} = \sqrt{u_1^2 + u_2^2 + \ldots + u_n^2}$$

- The dot product measures the cosine of the angle θ between the two vectors, scaled by their magnitudes

$$\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\| \|\mathbf{v}\| \cos\theta$$

- **Orthogonality** Two vectors are orthogonal (perpendicular) if their dot product is zero

$$\mathbf{u} \cdot \mathbf{v} = 0$$

HYPERPLANES

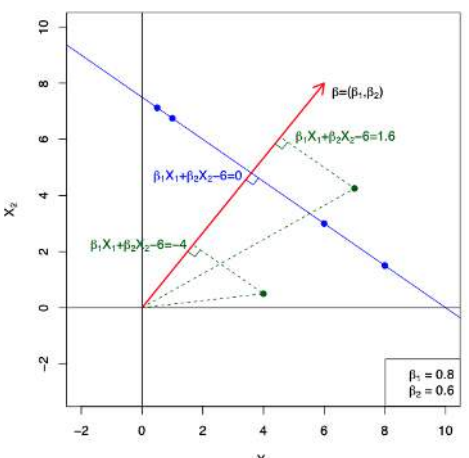
- **Hyperplanes** In a p-dimensional space, a hyperplane is a flat affine subspace of dimension $p - 1$
  - In a two-dimensional space, a hyperplane is a flat one-dimensional subspace, i.e., a line
  - In three dimensions, a hyperplane is a flat two-dimensional subspace, i.e., a plane
  - When $p > 3$, hyperplanes become difficult to visualize
- A hyperplane in a p-dimensional space is defined by a linear equation

$$\beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + b = 0$$

- Geometric Interpretation
  - The hyperplane divides the space into two halves
  - The normal vector $\beta = [\beta_1, \ldots, \beta_p]$ is perpendicular to the hyperplane

Hyperplanes in 2D Space

- The hyperplane: $\beta_1 x_1 + \beta_2 x_2 - 6 = 0$
- The normal vector: $\beta = (\beta_1, \beta_2) = (0.8, 0.6)$
- Points on the blue line satisfy the hyperplane equation and have a dot product of 6 with the normal vector
- The green dashed lines show the projection of different points onto the normal vector, indicating the perpendicular distance from the hyperplane
- Positive and negative values of $\beta_1 x_1 + \beta_2 x_2 - 6$ correspond to points lying on either side of the hyperplane



Projection and Distance

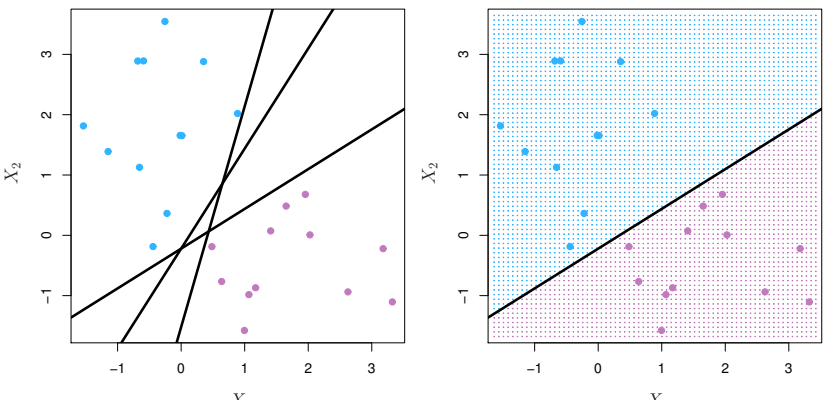
- **Projection of Vectors** The projection of vector $\mathbf{u}$ onto vector $\mathbf{v}$ is given by:

$$\text{proj}_{\mathbf{v}}\mathbf{u} = \left(\frac{\mathbf{u} \cdot \mathbf{v}}{\mathbf{v} \cdot \mathbf{v}}\right)\mathbf{v}$$

- **Distance from a Point to a Hyperplane** The distance d from a point $\mathbf{x_1}$ to a hyperplane $\beta \cdot \mathbf{x} + b = 0$ is:

$$d = \frac{|\beta \cdot \mathbf{x_1} + b|}{\|\beta\|}$$

Separating Hyperplanes



- Suppose $f(X) = \beta \cdot \mathbf{x} + b$, then $f(X) = 0$ defines a hyperplane
  - $f(X) > 0$ for points on one side of the hyperplane, and $f(X) < 0$ for points on the other
- If we code the colored points as $Y_i = +1$ for blue, say, and $Y_i = -1$ for mauve, then if $Y_i \cdot f(X_i) > 0$ for all i, $f(X) = 0$ defines a **separating hyperplane** , and the sample is **linearly separable**

Part II

SUPPORT VECTOR MACHINES

SVM Formulation

- Say the data sample S is linearly separable by some margin
- **Decision boundary**

$$g(\mathbf{x}) = \beta \cdot \mathbf{x} + b = 0$$

- **Linear classifier**

$$f(\mathbf{x}) = \text{sign}(g(\mathbf{x})) = \text{sign}(\beta \cdot \mathbf{x} + b)$$

- Idea: Find two parallel hyperplanes that correctly classify all the points and maximize the distance between them

SVM Formulation (Contd. 1)

- Decision boundary for the two hyperplanes

$$\beta \cdot \mathbf{x} + b = +1$$

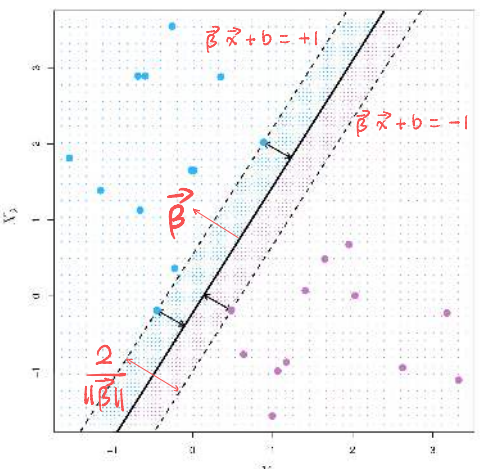
$$\beta \cdot \mathbf{x} + b = -1$$

- Training data is correctly classified if

$$\beta \cdot \mathbf{x}_i + b \geq +1 \quad \text{when } y_i = +1$$

$$\beta \cdot \mathbf{x}_i + b \leq -1 \quad \text{when } y_i = -1$$

- Equivalently: $y_i(\beta \cdot \mathbf{x}_i + b) \geq +1$ for all i
- What is the distance between the two hyperplanes?



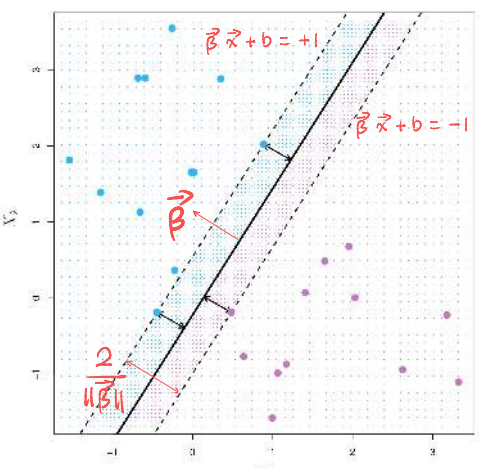
SVM Formulation (Contd. 2)

- Decision boundary for the two hyperplanes

$$\beta \cdot \mathbf{x} + b = +1$$

$$\beta \cdot \mathbf{x} + b = -1$$

- The distance between the two hyperplanes: $\frac{2}{\|\beta\|}$

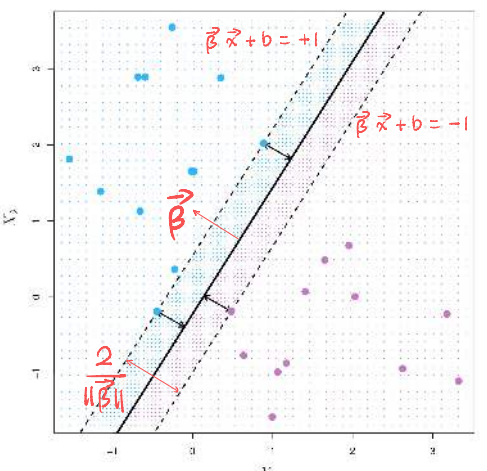


SVM Formulation (Contd. 3)

- SVM as an optimization problem:

  Maximize the distance: $\frac{2}{\|\beta\|}$
  Such that: $y_i(\beta \cdot \mathbf{x}_i + b) \geq +1$ (for all i)
- Equivalently:

  Minimize: $\frac{1}{2}\|\beta\|^2$
  Such that: $y_i(\beta \cdot \mathbf{x}_i + b) \geq +1$ (for all i)
- The optimization problem is a convex quadratic program, which can be solved efficiently
- The data point that lies closest to the decision boundary is the **support vector**



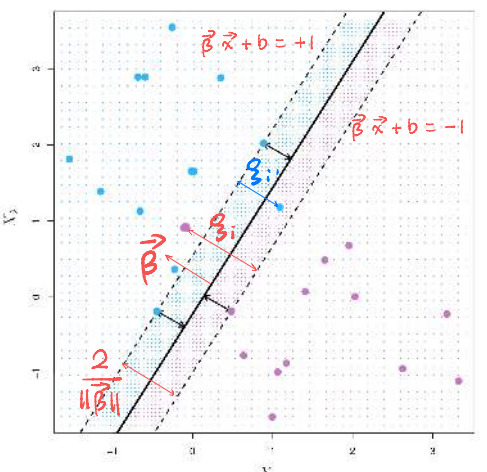
SVM Formulation (Non-Separable Case)

- Idea: Introduce a **slack** (ξ) for misclassified points, and minimize the slack
- SVM standard (primal) form (with slack):

  Minimize: $\frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$
  Such that:

$$y_i(\beta \cdot \mathbf{x}_i + b) \geq +1 - \xi_i$$
$$\xi_i \geq 0$$

  (for all i)

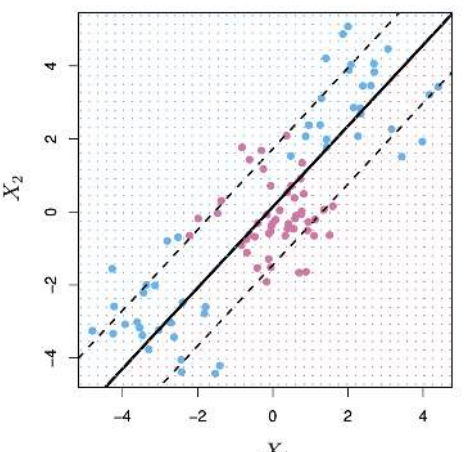


Part III

Generalization of SVM

Linear Boundary Can Fail

- Sometimes a linear boundary simply won't work, no matter what value of ξ
- What to do: **Feature transformation** , also known as **kernel transformation**



Example: Nonlinear Decision Boundary

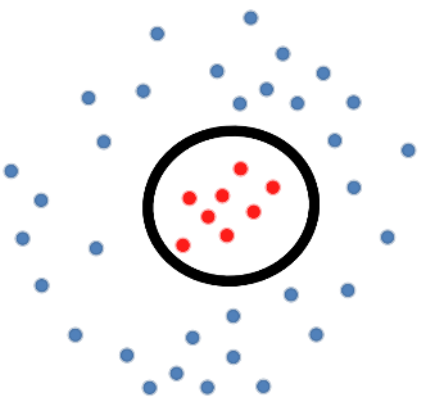
- The following training data is separable via a circular decision boundary:

$$\begin{aligned} g(\mathbf{x}) &= \beta_1 x_1^2 + \beta_2 x_2^2 + b \\ &= \beta_1 \mathcal{X}_1 + \beta_2 \mathcal{X}_2 + b \end{aligned}$$

- For example: Circle of radius r, i.e.,

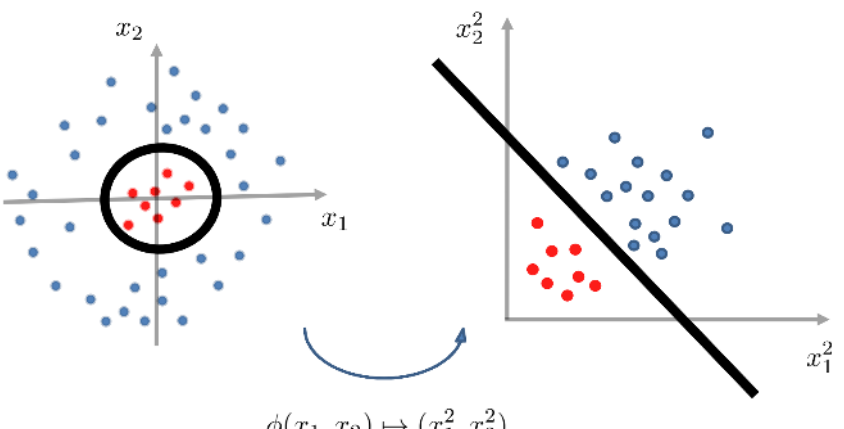
$$\beta_1 = \beta_2 = 1, b = -r^2$$

- The decision boundary is not linear in x_1 and x_2, but linear in $\mathcal{X}_1$ and $\mathcal{X}_2$



EXAMPLE: NONLINEAR DECISION BOUNDARY (CONTD.)

- Feature transformation on our data: $\phi(x_1, x_2) \mapsto (x_1^2, x_2^2)$
- Then g becomes linear in ϕ-transformed feature space!



Feature Transform for Quadratic Boundaries

- Generic quadratic boundary ($\mathbb{R}^2$ case): $\phi(x_1, x_2) \mapsto (x_1^2, x_2^2, x_1x_2, x_1, x_2, 1)$

$$\begin{aligned} g(\vec{x}) &= w_1x_1^2 + w_2x_2^2 + w_3x_1x_2 + w_4x_1 + w_5x_2 + w_0 \\ &= \sum_{p+q\leq 2} w^{p,q}x_1^p x_2^q \end{aligned}$$

- Generic quadratic boundary ($\mathbb{R}^d$ case):
  $\phi(x_1, \ldots, x_d) \mapsto (x_1^2, x_2^2, \ldots, x_d^2, x_1x_2, \ldots, x_{d-1}x_d, x_1, x_2, \ldots, x_d, 1)$

$$g(\vec{x}) = \sum_{i,j=1}^{d} \sum_{p+q\leq 2} w_{i,j}^{p,q} x_i^p x_j^q$$

- **Theorem** Given a data sample S, there exists a feature transform such that for any labelling of S is linearly separable in the transformed space

SVMs: MORE THAN 2 CLASSES?

- **OVA** (One versus All): Fit K different 2-class SVM classifiers $\hat{f}_k(\mathbf{x})$, $k = 1, \ldots, K$; each class versus the rest. Classify $\mathbf{x}^*$ to the class for which $\hat{f}_k(\mathbf{x}^*)$ is largest
- **OVO** (One versus One): Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_{k\ell}(\mathbf{x})$. Classify $\mathbf{x}^*$ to the class that wins the most pairwise competitions
- If K is not too large, use OVO

SVM versus Logistic Regression

- Only logistic regression can provide a prediction of probability
- I personally prefer the data-driven way to select from SVM and logistic regression
  - Choose the model that performs the best on the validation set