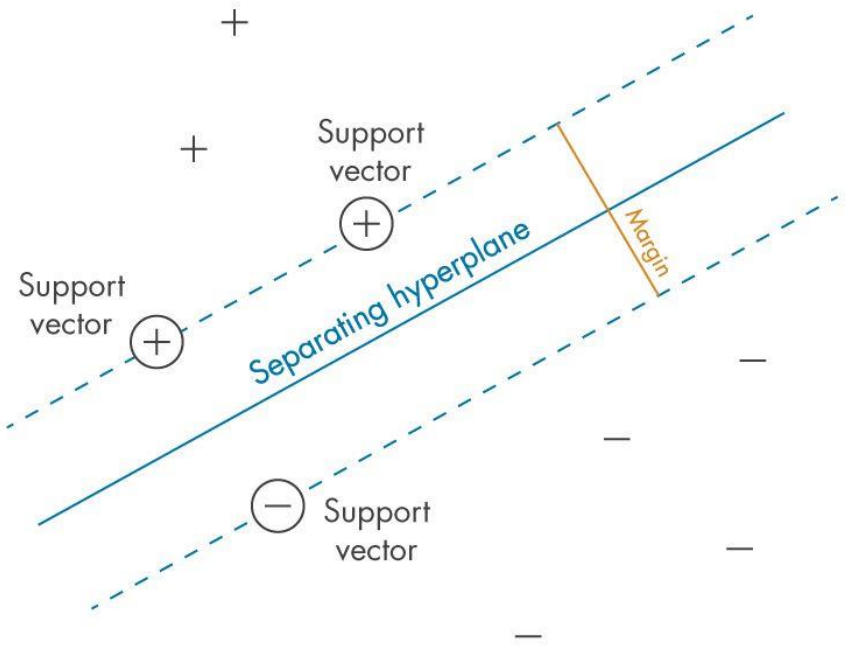
# Lecture 2.7

- Support Vector Machine

# Support Vector Machine

- **Support Vector Machine (SVM)** is a supervised machine learning algorithm widely used for classification and regression tasks
- The main goal of SVM is to find a **hyperplane** that best **separates data into classes** in a way that the margin between the nearest points (**support vectors**) of each class is maximized



# Linear SVM

- For simplicity, we begin with a linearly separable dataset
- Hyperplane Equation: In an n-dimensional space, a hyperplane is defined as

$$w^T x + b = 0$$

Where,

$w$ is the weight vector (normal to the hyperplane)

$x$ is the input feature vector

$b$ is the bias (offset)

- Classification rule

$$y_i = \begin{cases} +1 & w^T x_i + b \geq +1 \\ -1 & w^T x_i + b < -1 \end{cases}$$

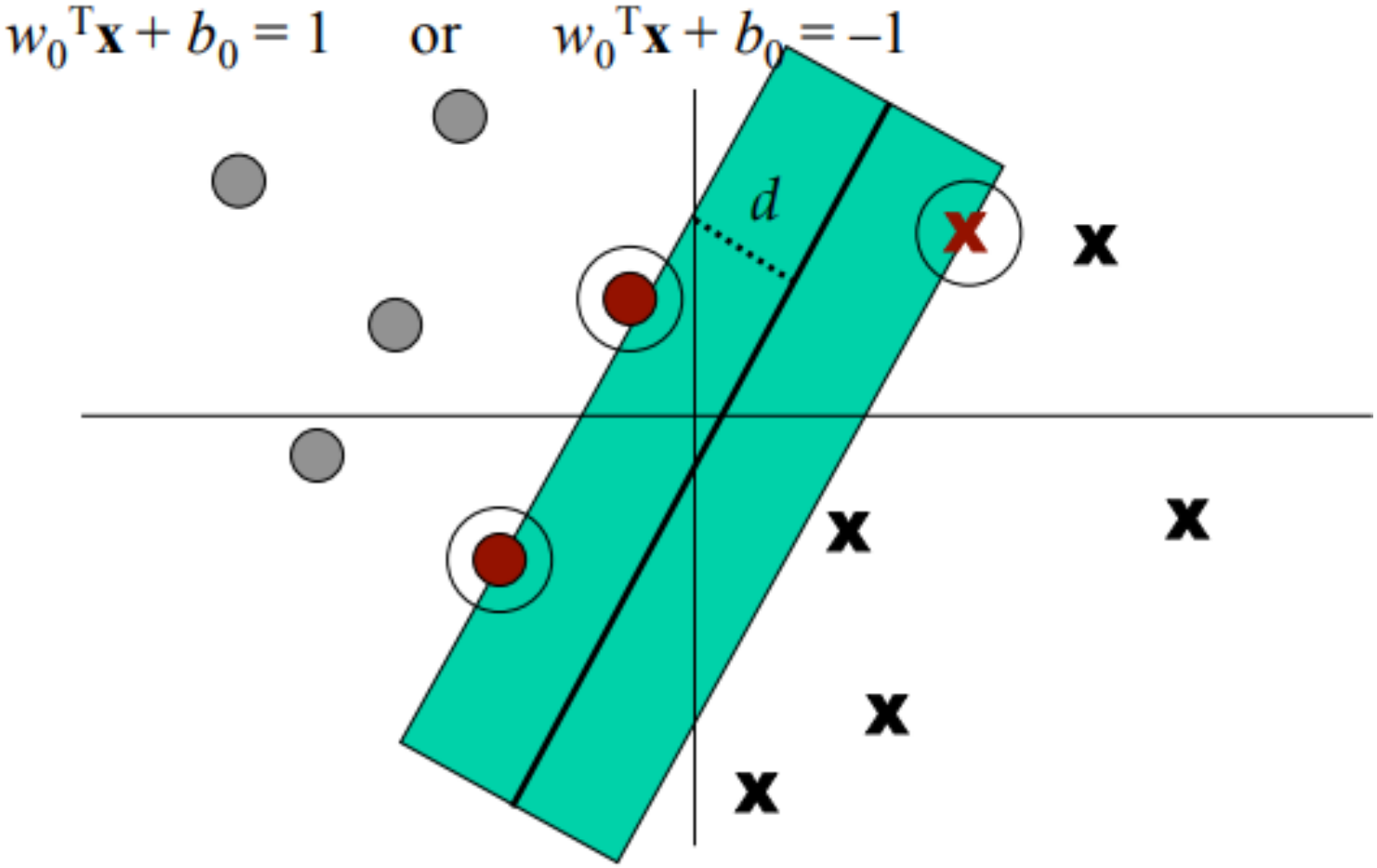
- Multiplying all the $x_i$ with $y_i$, we get

$$y_i(w^T x_i + b) \geq 1$$

- For support vectors

$$y_i(w^T x_i + b) = 1$$

# Linear SVM Diagram



# Definition of Margin

- The margin is the perpendicular distance from any point $x_i$ to the hyperplane
- This distance is given by

$$Distance = \frac{|w^T x_i + b|}{\|w\|}$$

- Substituting $y_i(w^T x_i + b) = 1$ into the distance formula

$$Distance = \frac{1}{\|w\|}$$

- The total margin is the distance between the two parallel hyperplane $w^T x + b = 1$ and $w^T x + b = -1$:

$$Distance = \frac{2}{\|w\|}$$

# SVM Optimization

- The one way to optimize is to maximize the margin $\frac{2}{\|w\|}$
- However, we minimize convex quadratic function $\frac{1}{2}\|w\|^2$ which is equivalent to maximize $\frac{2}{\|w\|}$
- Therefore, we use formulation of:

$$\text{Minimize: } \frac{1}{2}\|w\|^2$$

$$\text{Subject to:}$$

$$y_i(w^T x_i + b) \geq 1$$

# Lagrangian Formulation

- To incorporate the constraints $y_i(w^T x_i + b) \geq 1$ into the optimization problem, we introduce non-negative Lagrange multipliers $\alpha_i \geq 0$
- The Lagrangian is:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$$

- The Karush-Kuhn-Tucker (KKT) conditions must hold for the solution to be optimal
- The conditions are:

$$i. \quad \frac{\delta L}{\delta w} = 0$$

$$ii. \quad \frac{\delta L}{\delta b} = 0$$

$$iii. \quad \frac{\delta L}{\delta \alpha} \geq 0$$

# Derive Dual Formulation

*i.* $\frac{\delta L}{\delta w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \Rightarrow \boldsymbol{w = \sum_{i=1}^{n} \alpha_i y_i x_i}$

*ii.* $\frac{\delta L}{\delta b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \boldsymbol{\sum_{i=1}^{n} \alpha_i y_i = 0}$

Now,

$$\|w\|^2 = \left(\sum_{i=1}^{n} \alpha_i y_i x_i\right)^T \left(\sum_{j=1}^{n} \alpha_j y_j x_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

Substituting (i) and (ii) into $L(w, b, \alpha)$

$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1]$ we get

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{n} \alpha_i$$

# Dual Optimization Problem

Substitute these conditions back into the Lagrangian to derive the dual formulation

Maximize: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$

Subject to:

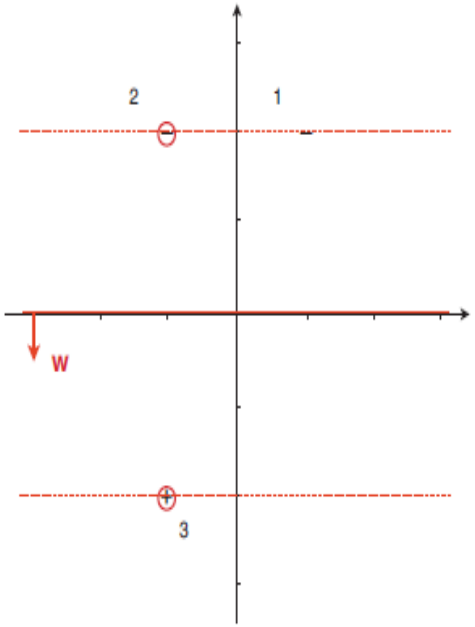
$$\sum_{i=1}^{n} \alpha_i y_i = 0 \, where \, \alpha_i \geq 0$$

Once the dual problem is solved, we get the optimal $\alpha_i$, which are used to compute $w$ and $b$

# Non-linear SVM (Kernel Trick)

- For non-linear data, SVM maps the data into a higher-dimensional space using a kernel function $K(x_i, x_j)$ without explicitly computing the transformation
- Common Kernel Functions:
  - Linear Lernel: $K(x_i, x_j) = x_i^T x_j$
  - Polynomial Kernel: $K(x_i, x_j) = (x_i^T x_j + 1)^d$
  - Gaussian (RBF) Kernel: $K(xi, xj) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$
  - Sigmoid Kernel: $K(xi, xj) = \tanh(\kappa x_i^T x_j + c)$

# Example 1




**Example 7.5 (Two maximum-margin classifiers and their support vectors).** Let the data points and labels be as follows (see Figure 7.8 (left)):

$$\mathbf{X} = \begin{pmatrix} 1 & 2 \\ -1 & 2 \\ -1 & -2 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} -1 \\ -1 \\ +1 \end{pmatrix} \quad \mathbf{X}' = \begin{pmatrix} -1 & -2 \\ 1 & -2 \\ -1 & -2 \end{pmatrix}$$

The matrix $\mathbf{X}'$ on the right incorporates the class labels; i.e., the rows are $y_i \mathbf{x}_i$. The Gram matrix is (without and with class labels):

$$\mathbf{X}\mathbf{X}^{\mathrm{T}} = \begin{pmatrix} 5 & 3 & -5 \\ 3 & 5 & -3 \\ -5 & -3 & 5 \end{pmatrix} \quad \mathbf{X}'\mathbf{X}'^{\mathrm{T}} = \begin{pmatrix} 5 & 3 & 5 \\ 3 & 5 & 3 \\ 5 & 3 & 5 \end{pmatrix}$$

The dual optimisation problem is thus

$$\arg\max_{\alpha_1,\alpha_2,\alpha_3} -\frac{1}{2}\left(5\alpha_1^2 + 3\alpha_1\alpha_2 + 5\alpha_1\alpha_3 + 3\alpha_2\alpha_1 + 5\alpha_2^2 + 3\alpha_2\alpha_3 + 5\alpha_3\alpha_1 + 3\alpha_3\alpha_2 + 5\alpha_3^2\right) + \alpha_1 + \alpha_2 + \alpha_3$$

$$= \arg\max_{\alpha_1,\alpha_2,\alpha_3} -\frac{1}{2}\left(5\alpha_1^2 + 6\alpha_1\alpha_2 + 10\alpha_1\alpha_3 + 5\alpha_2^2 + 6\alpha_2\alpha_3 + 5\alpha_3^2\right) + \alpha_1 + \alpha_2 + \alpha_3$$

subject to $\alpha_1 \geq 0, \alpha_2 \geq 0, \alpha_3 \geq 0$ and $-\alpha_1 - \alpha_2 + \alpha_3 = 0$. While in practice such problems are solved by dedicated quadratic optimisation solvers, here we will show how to solve this toy problem by hand.

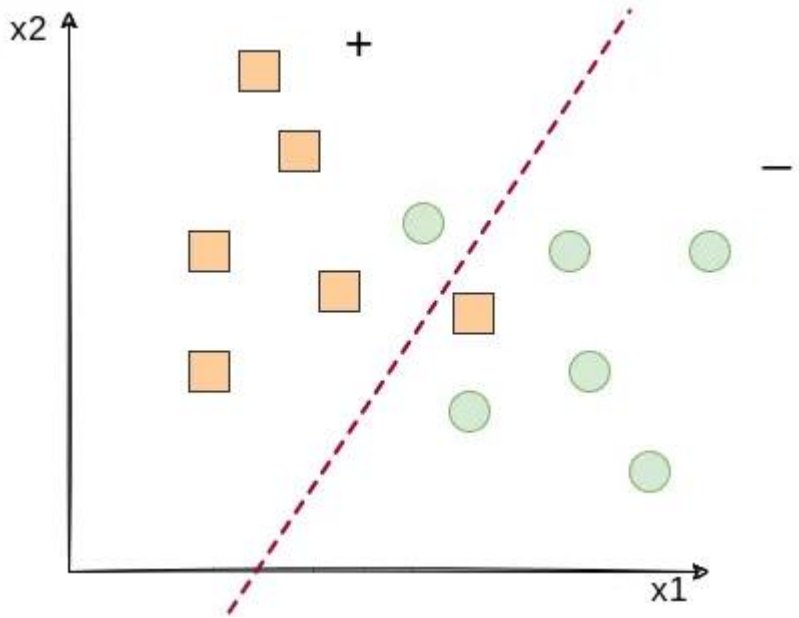
Using the equality constraint we can eliminate one of the variables, say $\alpha_3$, and simplify the objective function to

$$\arg\max_{\alpha_1,\alpha_2,\alpha_3} -\frac{1}{2}\left(5\alpha_1^2+6\alpha_1\alpha_2+10\alpha_1(\alpha_1+\alpha_2)+5\alpha_2^2+6\alpha_2(\alpha_1+\alpha_2)+5(\alpha_1+\alpha_2)^2\right)$$
$$+2\alpha_1+2\alpha_2$$
$$=\arg\max_{\alpha_1,\alpha_2,\alpha_3} -\frac{1}{2}\left(20\alpha_1^2+32\alpha_1\alpha_2+16\alpha_2^2\right)+2\alpha_1+2\alpha_2$$

Setting partial derivatives to 0 we obtain $-20\alpha_1-16\alpha_2+2=0$ and $-16\alpha_1-16\alpha_2+2=0$ (notice that, because the objective function is quadratic, these equations are guaranteed to be linear). We therefore obtain the solution $\alpha_1 = 0$ and $\alpha_2 = \alpha_3 = 1/8$. We then have $\mathbf{w} = 1/8(\mathbf{x}_3 - \mathbf{x}_2) = \begin{pmatrix} 0 \\ -1/2 \end{pmatrix}$, resulting in a margin of $1/\|\mathbf{w}\| = 2$. Finally, $t$ can be obtained from any support vector, say $\mathbf{x}_2$, since $y_2(\mathbf{w}\cdot\mathbf{x}_2 - t) = 1$; this gives $-1\cdot(-1-t)=1$, hence $t=0$. The resulting maximum-margin classifier is depicted in Figure 7.8 (left). Notice that the first example $\mathbf{x}_1$ is not a support vector, even though it is on the margin: this is because removing it will not affect the decision boundary

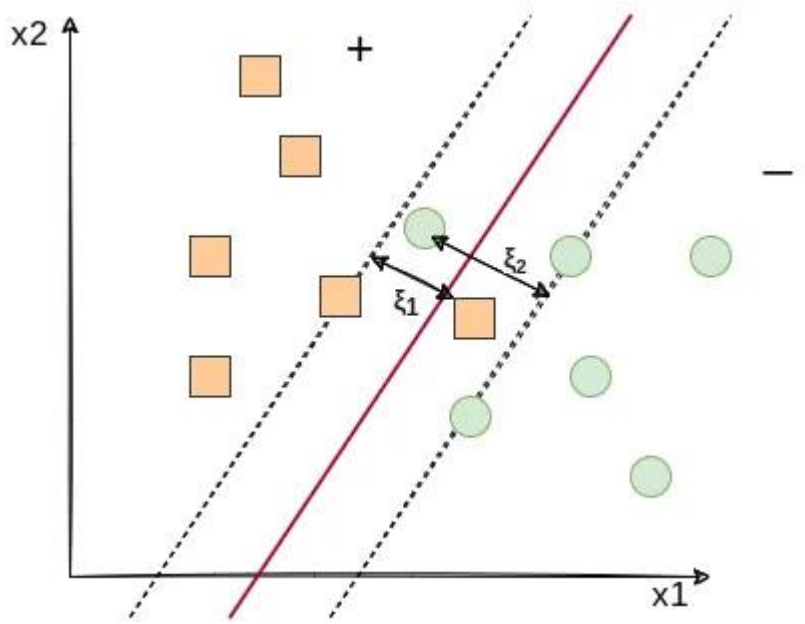
# Soft-Margin SVM

- If there are some data points placed in the opposite directions of the true classes the primal problem will be infeasible; hence, it has no solution



# Slack Variable

- To tackle these problems, we introduce **a slack variable $\xi$**
- Consider the data points outside the lines defining the margin
- $\xi$ is the distance from these data points to the lines defining the margin



# Soft Margin SVM Optimization

$$Min\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

Subject to:

$$y_i(w^T x_i + b) \geq 1 - \xi_i$$

Where,

$$\xi_i \geq 0$$

# Lagrangian Multiplier Application

$$L(w, b, \alpha, \xi): \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{n}\alpha_i[y_i(w^T x_i + b) - 1]$$

# Lagrangian Formulation

- The Karush-Kuhn-Tucker (KKT) conditions must hold for the solution to be optimal
- The conditions are:

$$i. \quad \frac{\delta L}{\delta w} = 0$$

$$ii. \quad \frac{\delta L}{\delta b} = 0$$

$$iii. \quad \frac{\delta L}{\delta \xi} = 0$$