# Support Vector Machines

## Regression and Classification

The main idea behind Support Vector Machines is to find an optimal hyperplane that best separates or fits the data.

**For Classification (SVC):**

- The hyperplane is a decision boundary that separates data points of different classes.
- "Optimal" means it has the **maximum margin** (distance) from the nearest data points of any class.

**For Regression (SVR):**

- The hyperplane is a function that best fits the data.
- "Optimal" means it has as many data points as possible within an $\epsilon$-insensitive tube, balancing model complexity and prediction error.

### The Kernel Trick

For non-linear data, SVMs use the **kernel trick** to map data into a high-dimensional feature space where a linear hyperplane can be found. All computations are done using a kernel function, avoiding explicit mapping.

# Goal of Support Vector Regression (SVR)

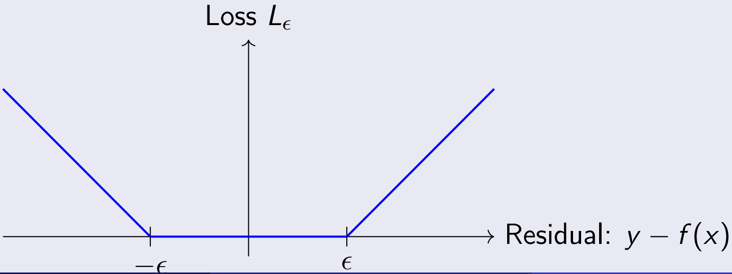
## Fitting an $\epsilon$-Insensitive Tube

Unlike traditional regression (e.g., Least Squares) which tries to minimize error for all points, SVR aims to find a function $f(x) = \mathrm{w}^T\mathrm{x} + b$ such that most data points $(x_i, y_i)$ lie within an $\epsilon$-tube.

### The $\epsilon$-Insensitive Loss Function

Errors are only penalized if a data point's residual, $|y - f(x)|$, is greater than $\epsilon$.

$$L_\epsilon(y, f(x)) = \max(0, |y - f(x)| - \epsilon)$$



# SVR: The Primal Formulation

## Minimizing Complexity and Error

We want to minimize the norm of w to control model complexity. To handle points outside the $\epsilon$-tube, we introduce non-negative slack variables $\xi_i$ (for points above the tube) and $\xi_i^*$ (for points below).

### Primal Optimization Problem

$$\min_{\mathrm{w}, b, \xi, \xi^*} \frac{1}{2}||\mathrm{w}||^2 + C \sum_{i=1}^{n} (\xi_i + \xi_i^*)$$

subject to:

$$\begin{aligned} y_i - (\mathrm{w}^T \mathrm{x}_i + b) &\leq \epsilon + \xi_i \\ (\mathrm{w}^T \mathrm{x}_i + b) - y_i &\leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0 \quad \text{for } i = 1, \ldots, n \end{aligned}$$

The constant $C > 0$ is a regularization parameter that controls the trade-off between the flatness of $f(x)$ and the tolerance for errors.

# SVR: The Dual Formulation

## Derivation via Lagrange Multipliers

We introduce Lagrange multipliers $\alpha_i, \alpha_i^*, \mu_i, \mu_i^* \geq 0$ and form the Lagrangian. Taking derivatives with respect to the primal variables $(\mathrm{w}, b, \xi_i, \xi_i^*)$ and setting to zero gives:

$$\frac{\partial \mathcal{L}}{\partial \mathrm{w}} = 0 \quad \Longrightarrow \quad \mathrm{w} = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \mathrm{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad \Longrightarrow \quad C - \alpha_i - \mu_i = 0 \quad \Longrightarrow \quad \alpha_i \leq C$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i^*} = 0 \quad \Longrightarrow \quad C - \alpha_i^* - \mu_i^* = 0 \quad \Longrightarrow \quad \alpha_i^* \leq C$$

### The Dual Optimization Problem

Substituting these back into the Lagrangian yields the dual problem to be

# SVR, Representer Theorem, and Kernels I

The dual formulation reveals two key insights:

## 1. Connection to Representer Theorem

The expression for the optimal weight vector, $\mathbf{w} = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \mathbf{x}_i$, shows that $\mathbf{w}$ is a linear combination of the input data points. This is exactly what the **Representer Theorem** states: the solution to this type of regularized problem lies in the span of the training data.

## 2. The Kernel Trick in Action

The dual objective function depends only on the dot product $\mathbf{x}_i^T \mathbf{x}_j$. This allows us to apply the kernel trick by replacing it with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$.

### Prediction for a New Point z

The prediction function becomes:

$$f(\mathrm{z}) = \mathrm{w}^T \mathrm{z} + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) \mathrm{x}_i^T \mathrm{z} + b = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) K(\mathrm{x}_i, \mathrm{z}) + b$$

The data points with non-zero coefficients $(\alpha_i - \alpha_i^*)$ are the **Support Vectors**.

# SVR in Practice: A Non-Linear Example I

We generate noisy sinusoidal data and fit an SVR model using a non-linear kernel (Radial Basis Function - RBF):

$$K(\mathrm{x}, \mathrm{z}) = \exp(-\gamma \|\mathrm{x} - \mathrm{z}\|_2^2)$$

The figure shows:

- The original noisy data points.
- The SVR prediction curve, which captures the underlying non-linear trend.
- The $\epsilon$-tube around the prediction.
- The **Support Vectors** (points on or outside the tube) that define the model.

# SVR in Practice: A Non-Linear Example II

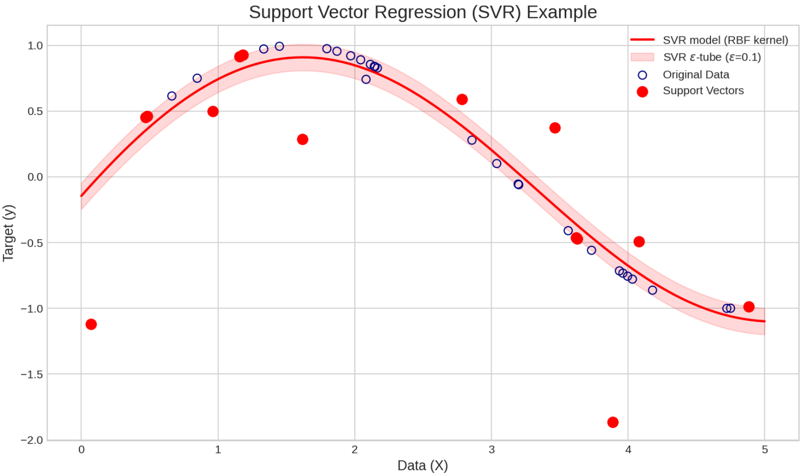



Figure: SVR with an RBF kernel fitting non-linear data.

# Goal of Support Vector Classification (SVC)

## Finding the Maximum-Margin Hyperplane

For a binary classification problem with labels $y_i \in \{-1, 1\}$, SVC aims to find a hyperplane $(\mathrm{w}, b)$ that separates the two classes with the largest possible margin.

### The Margin

The margin is the distance between the two parallel hyperplanes $\mathrm{w}^T\mathrm{x} + b = 1$ and $\mathrm{w}^T\mathrm{x} + b = -1$. This distance can be shown to be $\frac{2}{||\mathrm{w}||}$. Maximizing the margin is therefore equivalent to minimizing $||\mathrm{w}||^2$.

# The Hinge Loss Function I

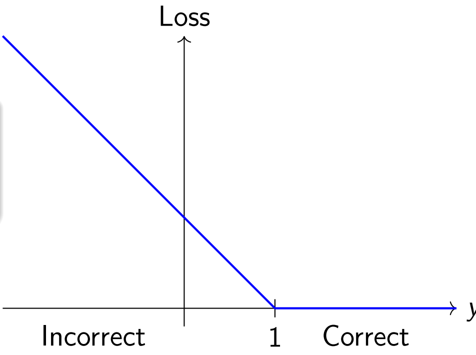
For SVC, the decision function is $f(x) = \mathrm{w}^T\mathrm{x} + b$. The product $y_i f(x_i)$ measures how correctly and confidently a point is classified.

- If $y_i f(x_i) \geq 1$: The point is correctly classified and outside the margin. No penalty.
- If $y_i f(x_i) < 1$: The point is inside the margin or misclassified. It incurs a linear penalty.

# The Hinge Loss Function II

**Hinge Loss Formula**

$$L(y, f(x)) = \max(0, 1 - yf(x))$$



To handle non-separable data, we introduce slack variables $\xi_i \geq 0$ (soft-margin SVC). The term $\sum \xi_i$ is an upper bound on the number of misclassifications and is equivalent to using the Hinge Loss.

## Primal Optimization Problem

$$\min_{\mathrm{w},b,\xi} \frac{1}{2}||\mathrm{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to:

$$y_i(\mathrm{w}^T\mathrm{x}_i + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0$$

## Dual Optimization Problem

Following a similar derivation with Lagrange multipliers, we arrive at the dual problem:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathrm{x}_i, \mathrm{x}_j)$$

subject to:

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C$$

The prediction for a new point z is $\mathrm{sign}(\sum_{i=1}^{n} \alpha_i y_i K(\mathrm{x}_i, \mathrm{z}) + b)$.

# SVR vs. SVC: Key Differences

## Regression vs. Classification

While both are SVMs, their objectives and mechanics are fundamentally different.

### Support Vector Regression (SVR)

- **Goal:** Fit a function to data.
- **Loss:** Uses $\epsilon$-**insensitive loss**. Errors are ignored if they are within the $\epsilon$-tube.
- **Constraints:** Two-sided constraints to keep data points inside the tube.
- **Support Vectors:** Points on the margin or outside the $\epsilon$-tube.

### Support Vector Classification (SVC)

- **Goal:** Find a separating boundary.
- **Loss:** Uses **Hinge Loss**, which penalizes points inside the margin or on the wrong side of the hyperplane.
- **Constraints:** One-sided constraint to ensure points are correctly classified with a margin.
- **Support Vectors:** Points on the margin or misclassified points.