Support Vector Machines (SVM): Kernels

1. Introduction to Kernels

The kernel is a mathematical technique used in Support Vector Machines (SVM) to handle non-linear decision boundaries. It allows SVM to implicitly map data into a higher-dimensional feature space without explicitly computing the transformation.

Why Do We Need Kernels?

- In many real-world problems, the data is not linearly separable in the original feature space.
- Transforming the data into a higher-dimensional space can make it linearly separable.
- However, explicitly computing the transformation is computationally expensive or infeasible for high dimensions.
- Kernels solve this problem by implicitly performing the transformation through the **kernel trick**.

—

2. Mathematical Formulation of Kernels

Feature Mapping

Let $\phi(\mathbf{x})$ represent a transformation that maps the data from the original space $\mathbb{R}^d$ to a higher-dimensional space $\mathbb{R}^p$ where $p > d$:

$$\phi : \mathbf{x} \in \mathbb{R}^d \mapsto \phi(\mathbf{x}) \in \mathbb{R}^p$$

The decision boundary in the higher-dimensional space is:

$$\phi(\mathbf{w}) \cdot \phi(\mathbf{x}) + b = 0$$

Kernel Function

Instead of explicitly computing $\phi(\mathbf{x})$, the kernel function computes the inner product in the higher-dimensional space directly:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

This is known as the **kernel trick**.

—

3. SVM Dual Problem with Kernels

The dual formulation of SVM uses the kernel function. The objective function in the dual form becomes:

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

Subject to:

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

The kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ replaces the dot product $\mathbf{x}_i \cdot \mathbf{x}_j$ in the input space.

—

4. Common Kernel Functions

4.1 Linear Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

- Equivalent to standard linear SVM. - Suitable for linearly separable data.

4.2 Polynomial Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^d$$

- Maps data into a polynomial feature space. - d: Degree of the polynomial. - c: A constant controlling higher-order terms.

4.3 Radial Basis Function (RBF) Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right)$$

- Measures similarity between two points. - $\gamma > 0$: Determines the influence of individual data points.

4.4 Sigmoid Kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha(\mathbf{x}_i \cdot \mathbf{x}_j) + c)$$

- Mimics the behavior of neural networks. - α and c: Parameters to tune the kernel.

—

5. Properties of Kernels

- A kernel function must satisfy Mercer's condition, ensuring that it corresponds to a valid inner product in some feature space.
- Kernels enable non-linear decision boundaries without explicitly transforming data.

—

6. Choosing the Right Kernel

The choice of kernel depends on the problem:

- **Linear Kernel:** Use when the data is linearly separable.
- **Polynomial Kernel:** Use when interactions between features are polynomial in nature.
- **RBF Kernel:** Default choice for non-linear problems.
- **Sigmoid Kernel:** Use for problems where sigmoid activation is relevant.

—

7. Advantages of Kernels

- **Non-linear Decision Boundaries:** Kernels allow SVM to handle complex patterns.
- **Efficient Computation:** The kernel trick avoids explicitly mapping to high-dimensional spaces.
- **Versatility:** A wide range of kernel functions can be used for different types of data.

—

8. Conclusion

The kernel function is a fundamental concept in SVM that extends its applicability to non-linear problems. By implicitly transforming the feature space using the kernel trick, SVM can find optimal decision boundaries even in complex datasets. The choice of kernel function and its parameters significantly impacts the performance of the model.