Chapter 3: Kernel & Non-Parametric Methods

Introduction

In machine learning, not all patterns can be captured using simple linear models. To address complex, non-linear relationships in data, we often turn to **kernel methods** and **non-parametric models**. These methods do not assume a fixed form for the model but allow the complexity to grow with the data, making them powerful tools for flexible and accurate learning.

This chapter introduces the theoretical foundations and practical applications of **kernel-based learning algorithms** and **non-parametric techniques** such as **k-Nearest Neighbors (k-NN)**, **Parzen Windows**, and **Decision Trees**. You'll also explore **support vector machines with kernel tricks**, and how these approaches handle non-linearity and high-dimensional data.

3.1 Kernel Methods: Motivation and Basics

3.1.1 Limitations of Linear Models

- Linear models cannot capture non-linear decision boundaries.
- Feature transformation helps but can be computationally expensive and ad-hoc.

3.1.2 Kernel Trick

- A **kernel function** implicitly maps input features to a high-dimensional space without explicitly computing the transformation.
- The kernel trick allows dot products in high-dimensional feature spaces to be computed efficiently:

  $K(x, x') = \langle \phi(x), \phi(x') \rangle$

3.1.3 Common Kernels

- **Linear Kernel**: $K(x, x') = x^T x'$
- **Polynomial Kernel**: $K(x, x') = (x^T x' + c)^d$
- **RBF (Gaussian) Kernel**: $K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$
- **Sigmoid Kernel**: $K(x, x') = \tanh(\alpha x^T x' + c)$

3.2 Support Vector Machines (SVM) with Kernels

3.2.1 SVM Recap

- SVM seeks to find a hyperplane that maximizes the margin between classes.

3.2.2 SVM with Kernels

- Apply kernel trick to handle non-linear separations.
- Dual formulation:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

3.2.3 Soft Margin and C Parameter

- Allows misclassification.
- Balances margin maximization and classification error.

3.2.4 Advantages and Challenges

- **Advantages**:
    - Effective in high-dimensional spaces.
    - Robust to overfitting (with proper kernel and parameters).
- **Challenges**:
    - Choice of kernel and tuning parameters.
    - Computational cost for large datasets.

3.3 Non-Parametric Methods: Overview

3.3.1 Parametric vs Non-Parametric

Parametric	Non-Parametric
Fixed number of parameters	Flexible, grows with data
Example: Linear regression	Example: k-NN, Parzen windows

3.4 k-Nearest Neighbors (k-NN)

3.4.1 Basic Idea

- Given a new point, find the **k** closest points in the training set.
- Assign label based on majority (classification) or average (regression).

3.4.2 Distance Metrics

- Euclidean: $\sqrt{\sum_i (x_i - y_i)^2}$
- Manhattan: $\sum_i |x_i - y_i|$
- Minkowski: Generalized distance metric.

3.4.3 Pros and Cons

- **Pros**:
  - Simple, intuitive.
  - No training phase.
- **Cons**:
  - Computationally expensive at prediction time.
  - Sensitive to irrelevant features and scaling.

3.5 Parzen Windows and Kernel Density Estimation (KDE)

3.5.1 Probability Density Estimation

- Estimate underlying probability density from data.

3.5.2 Parzen Window Method

- Place a window (kernel function) on each data point.
- Average all to get estimate:

  $\hat{p}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$

- h: bandwidth or smoothing parameter

3.5.3 Choice of Kernel

- Common choices:

- Gaussian
- Epanechnikov
- Uniform

3.5.4 Curse of Dimensionality

- In high dimensions, KDE becomes less effective due to data sparsity.

3.6 Decision Trees

3.6.1 Structure and Splitting

- Tree-like model of decisions.
- Splits data based on feature thresholds to reduce impurity.

3.6.2 Impurity Measures

- **Gini Index**:

  $G = 1 - \sum_{i=1}^{C} p_i^2$

- **Entropy**:

  $H = -\sum_{i=1}^{C} p_i \log_2 p_i$

3.6.3 Pruning and Overfitting

- Full trees overfit; pruning improves generalization.

3.6.4 Advantages

- Interpretable.
- Non-linear decision boundaries.
- Handles mixed data types.

3.7 Model Selection and Hyperparameter Tuning

3.7.1 Cross-Validation

- Split data into training and validation sets.
- Common: k-fold cross-validation.

3.7.2 Grid Search & Random Search

- Search for best hyperparameters (e.g., k in k-NN, σ in RBF).

3.7.3 Bias-Variance Trade-Off

- Non-parametric methods tend to have low bias, high variance.
- Regularization and model simplification help balance this.

3.8 Real-World Applications

- **SVM with Kernels**: Handwriting recognition, face detection.
- **k-NN**: Recommender systems, anomaly detection.
- **KDE**: Density-based anomaly detection, image processing.
- **Decision Trees**: Credit scoring, medical diagnosis, business decision support.

Summary

In this chapter, we explored advanced methods that go beyond simple linear models by allowing for complex, non-linear relationships. **Kernel methods** empower algorithms like SVM to operate in high-dimensional feature spaces efficiently. **Non-parametric models** like **k-NN**, **Parzen Windows**, and **Decision Trees** offer great flexibility as they adapt to the structure of the data without a fixed form. While these methods offer powerful modeling capabilities, they also require careful tuning and can be sensitive to high-dimensional data and noise.
