Machine Learning

Dr.Hajialiasgari

Tehran University
Of
Medical Science

January 21, 2025

1. Overview

2. Cross Validation

1 Overview

2 Cross Validation

Parametric vs. Non-Parametric Methods in Classification

1. Parametric Methods

- **Definition**: Assume a specific form (parametric model) for the decision boundary or the data distribution.
- **Examples**: Logistic regression, Naive Bayes, Linear Discriminant Analysis (LDA).
- **Advantages**:
  - Simple and computationally efficient.
  - Easy to interpret.
  - Work well with smaller datasets if assumptions hold.
- **Disadvantages**:
  - Relies heavily on the correctness of the assumed model.
  - May perform poorly if the true relationship deviates from the assumed model.

Parametric Vs Non-Parametric Methods in Classification

2. Non-Parametric Methods

- **Definition**: Do not assume any specific form for the data distribution.
- **Examples**: k-Nearest Neighbors (k-NN), Decision Trees, Support Vector Machines (SVM), Random Forests.
- **Advantages**:
  - Flexible and can model complex patterns in data.
  - Often perform better when the relationship between input features and target is unknown or nonlinear.
- **Disadvantages**:
  - Higher computational cost, especially with large datasets.
  - May overfit if not properly regularized or pruned.

Key Differences: Parametric vs. Non-Parametric

Feature	**Parametric Methods**	**Non-Parametric Methods**
Assumptions	Strong (e.g., linearity)	Minimal or none
Flexibility	Limited	High
Data Requirement	Low	High
Complexity	Simple	Complex

Example Applications

Parametric Methods:

- Logistic regression for predicting a binary outcome like spam detection.

Non-Parametric Methods:

- k-NN for classifying handwritten digits.

k-Nearest Neighbors (k-NN)

Overview:

- k-Nearest Neighbors (k-NN) is a non-parametric, lazy learning algorithm.
- Used for both classification and regression tasks.

Key Features:

- **Non-Parametric:** Makes no assumptions about data distribution.
- **Lazy Learning:** No explicit training; stores data and predicts only when required.
- **Instance-Based:** Predictions rely on similarity between new and training data.

How k-NN Works

1. **Calculate Distance:** Compute the distance (e.g., Euclidean) between the new data point and all training points.
2. **Find Nearest Neighbors:** Identify the k closest training points.
3. **Vote (Classification):** Assign the class that is most common among the k neighbors.
    - For regression: Use the mean or weighted average of the neighbors' values.

Choosing k:

- Small k: Sensitive to noise and overfits.
- Large k: Robust but may underfit.
- Optimal k: Found via cross-validation.

Advantages and Disadvantages of k-NN

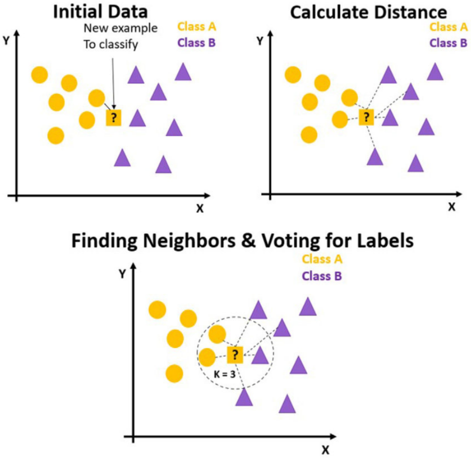
Advantages:

- Simple and easy to implement.
- No explicit training phase; quickly adapts to new data.
- Effective for small datasets with low dimensions.

Disadvantages:

- Computationally expensive for large datasets.
- Sensitive to noisy or irrelevant features.
- Requires careful choice of distance metric.

KNN Algorithm Visualization



Distance Calculation in KNN

Distance Calculation:

- Euclidean Distance:

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2}$$

- Manhattan Distance:

$$d(x, x_i) = \sum_{j=1}^{n} |x_j - x_{ij}|$$

- Minkowski Distance (generalized):

$$d(x, x_i) = \left(\sum_{j=1}^{n} |x_j - x_{ij}|^p \right)^{1/p}$$

Predicion in KNN

Prediction:

- For Classification:

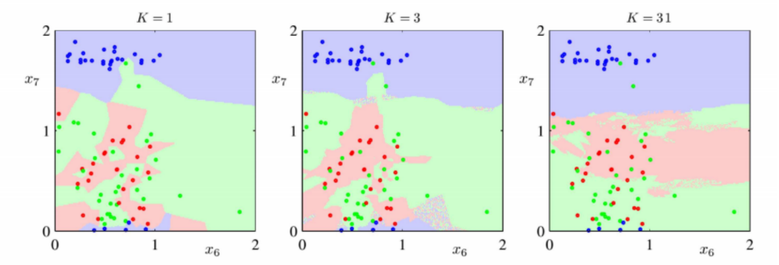
$$\hat{y} = \text{argmax}_c \sum_{i \in N_k} \mathbb{I}(y_i = c)$$

  where N_k is the set of k-nearest neighbors and $\mathbb{I}$ is the indicator function.

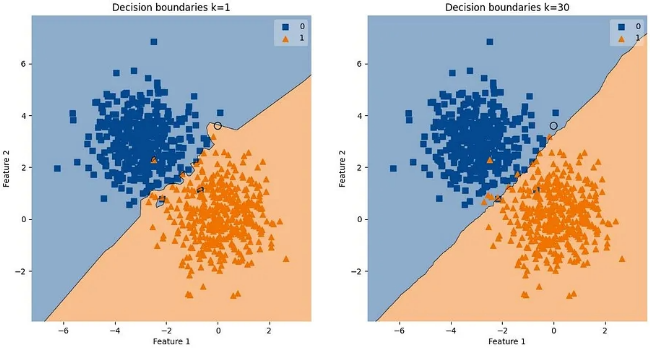
- For Regression:

$$\hat{y} = \frac{1}{k} \sum_{i \in N_k} y_i$$

Effect of K



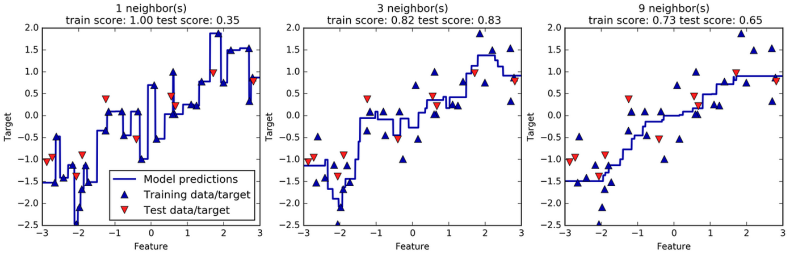
Effect of k


Decision boundaries k=1
Decision boundaries k=30
■ 0
▲ 1
Feature 1
Feature 2
−6
−4
−2
0
2
4
6

kNN for Regression

Although k-NN is primarily introduced for classification tasks, it can also be adapted for regression problems with some modifications. For instance, after identifying the nearest neighbors, instead of using majority voting, we can take the mean or median of the target variable values of these neighbors and use it as the predicted value.

KNN for Linear Regression



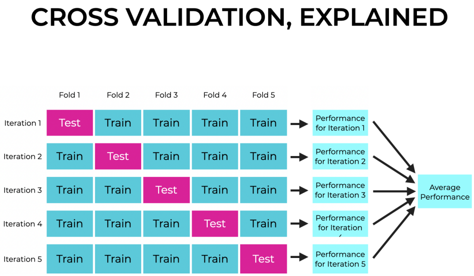
1 Overview

2 Cross Validation

Cross Validation

- **Purpose:** Technique for evaluating how well a model generalizes to unseen data.
- **How It Works:** Split data into k folds; train on $k-1$ folds and validate on the remaining fold.
- **Repeat Process:** Repeat k times, rotating the test fold each time. Average of all scores is the final score of the model.
- Cross-validation reduces overfitting and provides a more reliable estimation of model performance.
- **Note:** The model must be retrained at each iteration to avoid reusing a model that has already seen the test data, ensuring unbiased evaluation.

K-Fold Cross Validation



CROSS VALIDATION, EXPLAINED

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Iteration 1	Test	Train	Train	Train	Train	Performance for Iteration 1
Iteration 2	Train	Test	Train	Train	Train	Performance for Iteration 2
Iteration 3	Train	Train	Test	Train	Train	Performance for Iteration 3
Iteration 4	Train	Train	Train	Test	Train	Performance for Iteration
Iteration 5	Train	Train	Train	Train	Test	Performance for Iteration 5

Average Performance

Leave-One-Out Cross-Validation (LOOCV)

- **How It Works:** Uses a single data point as the validation set ($k = 1$) and the rest as the training set. Repeat for all data points.
- **Properties:**
  - No Data Wastage: Every data point is used for both training and validation.
  - High Variance, Low Bias.
  - Computationally Expensive: Requires training the model N times for N data points, making it slow for large datasets.
  - Best for small datasets.

For more information and code check the related notebook

End of Classification