CHAPTER 1

The k-Nearest Neighbor Classifier

The k-nearest neighbors (k-NN) algorithm is a type of instance-based learning algorithm used for classification and regression. Given a new, unknown observation, k-NN algorithm searches through the entire dataset to find the 'k' training examples that are closest to the new instance, and predicts the label based on these 'k' nearest neighbors.

1.1 The k-NN Algorithm

The algorithm can be summarized as follows:

1. Given a new observation $\mathbf{x}$, compute the distance between $\mathbf{x}$ and all points in the training set.
2. Identify the 'k' points in the training data that are closest to $\mathbf{x}$.
3. If k-NN is used for classification, output the most common class among these 'k' points as the prediction. If k-NN is used for regression, output the average of the values of these 'k' points as the prediction.

The distance between points can be calculated using various metrics, the most common one being the Euclidean distance:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where n is the number of features, and x_i and y_i are the corresponding features of $\mathbf{x}$ and $\mathbf{y}$.

1.2 Choosing the Right 'k'

The choice of 'k' has a significant impact on the k-NN algorithm. A small 'k' (like 1) can capture a lot of noise and lead to overfitting, while a large 'k' can smooth over many

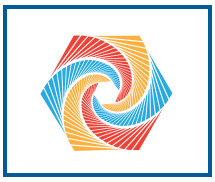
details and potentially lead to underfitting. Cross-validation is typically used to select an optimal 'k'.

1.3 Considerations

Although the k-NN algorithm is simple to understand and implement, it can be computationally intensive for large datasets, as it requires computing the distance between every pair of points. Additionally, it's sensitive to the choice of the distance metric and the scale of the features.

Appendix A

Answers to Exercises



Index