K-Nearest Neighbors

Sauman Das

November 2021

1 Introduction

K-Nearest Neighbors is one of the simplest machine learning algorithms that exist, but can be very effective in certain applications. In this lecture, we will be going through the algorithm but also the common applications of this model.

2 The Algorithm

In order to apply the K-Nearest Neighbors algorithms, we need 2 main components: a dataset and the k value. The following steps can be used to classify a single datapoint x:

1. From the training dataset, identify the k closest points to x using a distance metric (e.g. Euclidean or Manhattan)
2. Identify the class of each of the k closest points which is given because these points are part of the training dataset
3. The predicted class for x is the most common class from the k closest points

These three steps summarize the entire algorithm. Note how simple the algorithm is which means implementing it without libraries like Sci-kit Learn is possible.

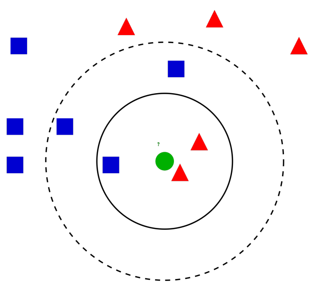

Figure 1: KNN Classification

3 Lazy Learner

Although we refer to KNNs as a machine learning “model”, in reality there is no model that is constructed. The only parameter that we learn is the value k. Unlike other machine learning models, we must store the entire training set in order to classify a test case. With decision trees, once we created the model, we did not need the training set anymore and only used the decision tree. However, with a KNN, we go through the entire training set when classifying each testing point.

4 Pros and Cons

Let’s start with the positive aspects of KNNs. There is only one parameter to tune which is k. This leads to an added benefit which is the simplicity of the model. Understanding how the model works is quite simple and it can be implemented efficiently. Because KNNs are lazy learners, there is no training time. This is likely the primary benefit of KNNs because in cases where we adaptively increase the training data, we do not have spend additional time on training.

As one can imagine, the amount of memory the KNN requires is a lot as the entire training dataset must be stored. This can potentially cause problems with RAM overflow or Resource Exhaustion errors. Furthermore, because we have to traverse the entire dataset and sort by the distance value for each test case, the runtime can be significantly larger than other models.

5 Application - Recommendation Systems

At some point in your life, you have probably reaped the benefit of a recommendation system. This may have been while searching for a movie to watch on Netflix or find a news article that matches the topic you were searching for. When desiging a recommender system, the underlying question we are answering is the following: “Can you give me the 3 most similar movies to the one I have selected”.

KNNs are the easiest way to solve this problem. We follow the same algorithm described in Section 2 modified slightly. We are given a feature vector for our selected movie and we have feature vectors for all the other movies. We go through each of these and then provide the 3 closest matches based on the distance metric.