Jeremy Marcus Tan
August 21, 2024

Case Study 1: Extended KNN

An extended version of the K-Nearest-Neighbors (KNN) algorithm, called weighted KNN, has several advantages over the classic KNN algorithm which gives equal weight to all of the nearest neighbors. The classic KNN algorithm simply follows the majority-rule approach which assigns the class of the test instance based on the class of the majority of the nearest neighbors. The weighted KNN gives different weights to each of the nearest neighbors inversely related to the distance of the neighbor to the data point it is trying to classify. It gives more weight to the nearer neighbors while giving less weight on the data points which are further away. The first major advantage of the weighted KNN algorithm over the classic KNN is that it eliminates possible ties when classifying. When using an even value for k, the classic KNN algorithm can result in a tie where an equal number of neighbors belong to both of the classes. The weighted KNN resolves this problem by classifying the test point based on the class of the nearest neighbors. Second, the classic KNN is prone to incorrectly classifying data points because it gives equal weight to all neighbors regardless of how close each neighbor is to the test point. For example, consider a relatively large value for k, say 10, and where the four nearest neighbors to the point to be classified belong to class A, while the other six neighbor points belong to class B. According to the classic KNN, the test point will be classified as B, despite the nearest neighbors all belonging to class A. On the other hand, the weighted KNN algorithm will correctly classify the point as belonging to class A since it will give proportionally more weight to the nearer neighbors.

Weighted KNN

Given a test point x_q to be classified and its k closest neighbors, $x_1, x_2, \ldots, x_k$ with class labels, $y_1, y_2, \ldots, y_k$. The class labels are given by v.

$$y_v = \sum_{i=1}^{k} w_i I(v = y_i)$$

$I(v = y_i)$ returns 1 if the label of the ith neighbor is v, $I(v = y_i)$ returns 0 if the label is not v. The weight of the ith neighbor is given by $w_i = \frac{1}{d(x_i, x_q)}$.

The weighted KNN algorithm classifies the test point x_q the class v with the greatest weighted sum, $max(y_v)$.