Technical Writing Assignment
Logan Gilmour

k-Nearest Neighbors

The k-Nearest Neighbors algorithm is a lazy-learning method that classifies entities based on the k-nearest training entities. 'Nearness' is calculated using a distance measure which, given two entities, returns a numerical distance between them. The algorithm is 'lazy' in that training entities are simply labeled and stored until a new entity requires classification. Then, the k-nearest training entities are found by comparing the new entity to training entities with the distance measure, and the new entity is classified using the majority label among those neighbors.

The distance measure is often a euclidean distance in an n-dimensional feature space, where n is the number of attributes used in classification. However, if some attributes are discrete rather than continuous (For example, words rather than numbers), an overlap or other method of determining distance might be used.

To meaningfully combine attributes, the distance measure must be well designed. Take an example of classification of cars as 'Compact Car', 'Luxury Car', 'Super Car', and so on. If the distance measure naively compares cars on the attributes 'Cost' and 'Top Speed' with no scaling, cost will be emphasized over top speed because cost has a far greater magnitude. The distance between 200kph and 300kph has far more meaning than the distance between $60 000 and $61 000, but the difference of 1000 is much larger than the difference of 100. An unscaled measure will essentially throw away the 'top speed' attribute.

k-NN requires all training entities to be stored, likely in memory. In a naive implementation, each classification requires the new entity to be compared to every training entity, meaning large training sets become impractical if many classifications are needed. With better data-structures for training entities, it is posible reducing the average number of comparisons required.