K-nearest neighbor algorithm

K-NEAREST NEIGHBOR is a simple algorithm that stores all available data points (examples) and classifies new data points based on a similarity measure. A variant of this algorithm addresses the task of function approximation.

In detail:

- Examples are described by **numerical attribute-values**. That is
  \[ \mathbf{x} = (x_1, x_2, x_3, \ldots, x_k) \] where \( k \) is the dimensionality of the instance space

- The complete example (or data) set \( D \) is simply stored in the "training phase". The data set \( D \) is defined as
  \[ D = \{(\mathbf{x}_i, y_i) \mid i = 1..n\}, \mathbf{x}_i \in \mathbb{R}^k, y_i \in \{0, 1\} \]

- Calculations are delayed until queries occur, **no trained model in the usual sense** is returned! Trained models are implicitly defined by the stored example set and the rules new instances are classified by.

- If there are \( k \) attributes all vectors can be interpreted as instances of \( \mathbb{R}^k \). The distance \( d(\mathbf{x}_1, \mathbf{x}_2) \) of two example vectors \( \mathbf{x}_1 \) and \( \mathbf{x}_2 \) is defined as their usual vector distance (Euclidean distance). That is:
  \[
d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + (x_{13} - x_{23})^2 + \ldots + (x_{1k} - x_{2k})^2}
\]

- The **distance** between two example vectors is regarded as a **measure for their similarity**. The smaller the distance the more similar the vectors.

- To classify a new instance \( \mathbf{x} \) from the set of stored examples, the \( K \) examples most similar to \( \mathbf{x} \) are determined. The new instance is assigned the class label most of these \( K \) examples belong to.

This approach is suited for **function approximation** as well. Instead of assigning the most frequent classification among the \( K \) examples most similar to an instance \( \mathbf{x} \), an average of the function values of the \( K \) examples is calculated as the prediction for the function value \( \mathbf{x} \).

A variant of this approach calculates a **weighted average** of the nearest neighbors. Given a specific instance \( \mathbf{x} \) that shall be classified, the weight of an example increases with increasing similarity to \( \mathbf{x} \).

A major problem of the simple approach of **K-NEAREST NEIGHBOR** is that the vector distance will not necessarily be suited for finding intuitively similar examples, especially if irrelevant attributes are present.

Source (modified): http://kiew.cs.uni-dortmund.de:8001/mlnet/instances/81d91e93-df4da3c279