k-Nearest Neighbors

Pedro Larrañaga, Concha Bielza

Computational Intelligence Group
Artificial Intelligence Department
Universidad Politécnica de Madrid

Machine Learning
Master in Data Science + Master in HMDA

Outline

1. The basic k-NN algorithm
2. Variants of the basic k-NN
3. Distance selection

Outline

1 The basic k-NN algorithm

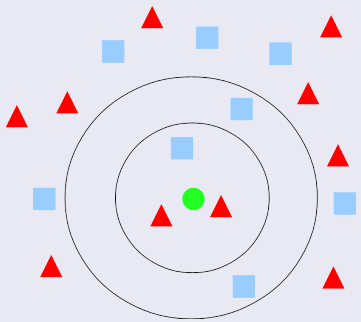
2 Variants of the basic k-NN

3 Distance selection

The basic k-NN algorithm

Main idea

- The k-nearest neighbor classifier (Fix and Hodges, 1951) is one of the best-known and most widely used nonparametric classifiers
- Simple, intuitive, no explicit model (transduction), lazy learning
- k-NN algorithm predicts the unknown class based on the classes associated with the k predictor instances of the training set that are closer to $\mathbf{x}$, using a simple majority decision rule



- If $k = 3$ (inner circle), the instance is assigned to the second class because there are two triangles and only one square inside the inner circle
- If $k = 5$ (outer circle), it is assigned to the first class (three squares vs. two triangles inside the outer circle)

The basic k-NN algorithm

Pseudocode for the basic k-NN algorithm

Algorithm 1: The basic k-nearest neighbor classifier

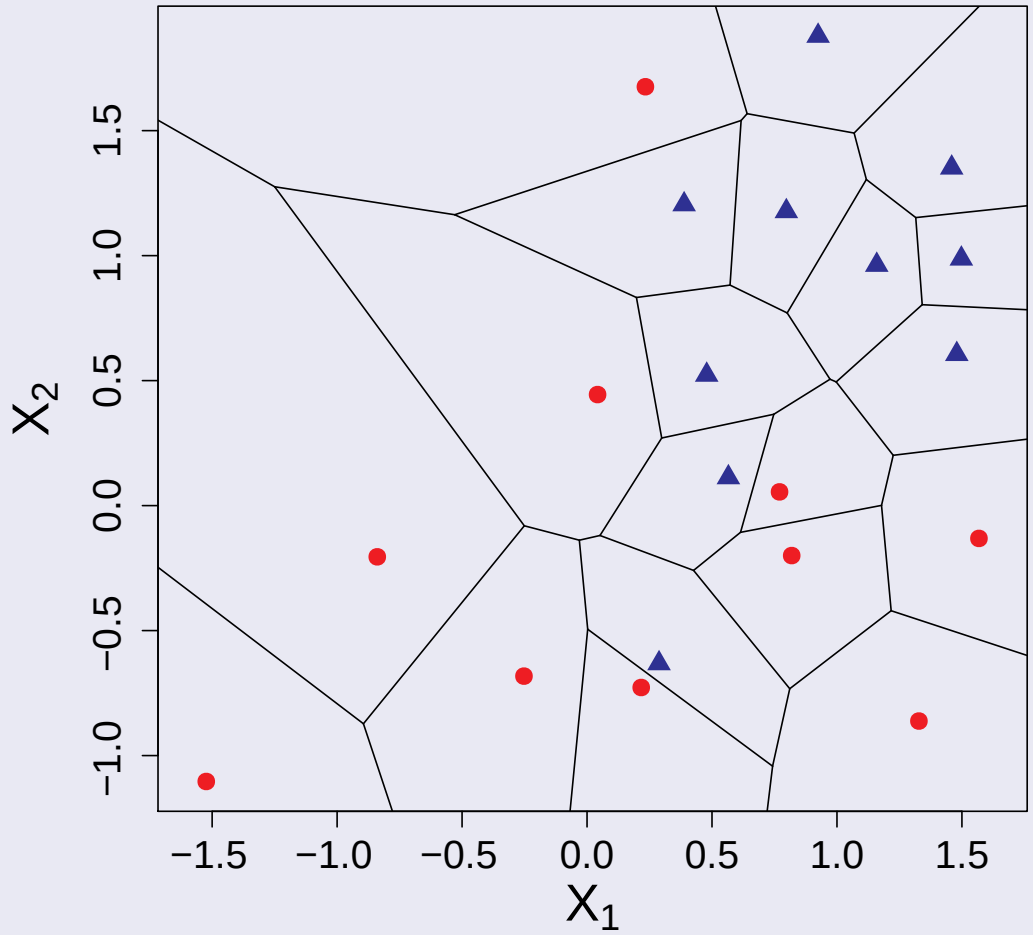
Input : A data set $\mathcal{D} = \{(\mathbf{x}^1, c^1), \ldots, (\mathbf{x}^N, c^N)\}$ of labelled instances, a new instance $\mathbf{x} = (x_1, \ldots, x_n)$ to be classified

Output: The class label for instance $\mathbf{x} = (x_1, \ldots, x_n)$

1 **for** *each labelled instance* $(\mathbf{x}^i, c^i)$ $i = 1$ **to** N **do**
2 | Calculate $d(\mathbf{x}^i, \mathbf{x})$
3 **endfor**
4 Order $d(\mathbf{x}^i, \mathbf{x})$ from lowest to highest, $i = 1$ **to** N
5 Select the k nearest instances to $\mathbf{x}$ obtaining the subset $\mathcal{D}_{\mathbf{x}}^k \subseteq \mathcal{D}$
6 Assign to $\mathbf{x}$ the most frequent class in $\mathcal{D}_{\mathbf{x}}^k$

The basic k-NN

Voronoi tessellation for a 1-NN classifier



The basic k-NN

Advantages and disadvantages

- Advantages
  1. Able to learn complex decision boundaries
  2. No loss of information because there is no modeling (abstraction) phase
  3. A local method
  4. Few assumptions about the data
  5. Easily adapted as an incremental algorithm and also works when the input is a data stream of instances
  6. Easily adapted to regression problems
- Disadvantages
  1. High storage requirements
  2. Slow in classification time
  3. Sensitive to the value of k, the distance metric choice, the existence of irrelevant variables, and noisy data set
  4. There is no explicit model

Outline

1 The basic k-NN algorithm

2 **Variants of the basic k-NN**

3 Distance selection

Variants of the basic k-NN

Variants

- Weighting neighbors
  - The contribution of each neighbor depends on its distance to the query instance, with more weight being attached to nearer neighbors
  - The weight w_j of the j-th neighbor can be defined as a decreasing function h of its distance to the instance to be classified, $\mathbf{x}$: $w_j = h(d(\mathbf{x}^j, \mathbf{x}))$
  - The query instance will be assigned to the label with the largest total weight
- Weighting predictor variables
  - A weight to each predictor variable, that can be proportional to its relevance with respect to the class variable
  - $d(\mathbf{x}, \mathbf{x}^i) = \sum_{j=1}^{n} w_j \delta(x_j, x_j^i)$, where w_j is the weight assigned to variable X_j, and $\delta(x_j, x_j^i)$ measures the distance between the j-th components of $\mathbf{x}$ and $\mathbf{x}^i$
  - The weight w_j may be proportional to the mutual information between X_j and C
- Average distance
  - The distances of the neighbors to the query instance are averaged for each class label, and the label associated with the minimum average distance is assigned to the query
- k-NN with rejection
  - Demands some guarantees before an instance is classified
  - If the guarantees are not met, the instance remains unclassified until processed by another supervised classification algorithm according to a cascading procedure
  - A usual guarantee refers to the threshold for the most frequent class for the neighbors of the instance to be classified

Outline

1 The basic k-NN algorithm

2 Variants of the basic k-NN

3 Distance selection

The metric learning problem

Distances for discrete and continuous predictors

- Euclidean distance (the standard) attaches the same importance to any variable and is not informative enough
- A general expression: $d(\mathbf{x}, \mathbf{x}^i) = \sum_{j=1}^{n} w_j \delta(x_j, x_j^i)$ where w_j is the weight assigned to variable X_j, and $\delta(x_j, x_j^i)$ measures the distance between the j-th components of $\mathbf{x}$ and $\mathbf{x}^i$

- Discrete predictors
  - Number of non-matching variables $\delta_{\text{no-matching}} = \begin{cases} 1 & \text{if } x_j \neq x_j^i \\ 0 & \text{if } x_j = x_j^i \end{cases}$
  - Value difference metric

$$d_{\text{VDM}}(\mathbf{x}, \mathbf{x}^i) = \sum_{j=1}^{n} w(x_j)\delta(x_j, x_j^i)$$

$\delta(x_j, x_j^i) = \sum_{c \in \Omega_C} (p(c|x_j) - p(c|x_j^i))^2$, where $p(c|x_j)$. The weight $w(x_j) = \sqrt{\sum_{c \in \Omega_C} p(c|x_j)^2}$ will be high for variable values that discriminate well between the class labels

- Continuous predictors
  - Minkowski distance

$$d_{\text{Minkowski}}(\mathbf{x}, \mathbf{x}^i) = \left(\sum_{j=1}^{n} |x_j - x_j^i)|^p \right)^{\frac{1}{p}}$$

  - Manhattan distance when $p = 1$
  - Euclidean distance when $p = 2$
  - Chebyshev distance when $p = \infty$: $d_{\text{Chebyshev}}(\mathbf{x}, \mathbf{x}^i) = \max_j |x_j - x_j^i|$

References

- D.W. Aha, D. Kibler, M.K. Albert (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37-66
- T.M. Covert, P.E. Hart (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13, 21-27
- P.A. Devijver, J. Kittler (1982). *Pattern Recognition. A Statistical Approach*. Prentice Hall
- E. Fix, J.L. Hodges (1951). *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation. Technical Report 4
- P.E. Hart (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3), 515-516