5. K-Nearest Neighbor

Pedro Larrañaga

Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country

Madrid, 25th of July, 2006

Outline

Outline

Basic Ideas

K-NN $\equiv$ IBL, CBR, lazy learning

- A new instance is classified as the most frequent class of its K nearest neighbors
- Very simple and intuitive idea
- Easy to implement
- There is not an explicit model (transduction)
- K-NN $\equiv$ instance based learning (IBL), case based reasoning (CBR), lazy learning

Outline

Algorithm for the basic K-NN

BEGIN

Input: $D = \{(\mathbf{x}_1, c_1), \ldots, (\mathbf{x}_N, c_N)\}$

$\mathbf{x} = (x_1, \ldots, x_n)$ new instance to be classified

FOR each labelled instance (x_i, c_i) calculate $d(\mathbf{x}_i, \mathbf{x})$

Order $d(\mathbf{x}_i, \mathbf{x})$ from lowest to highest, $(i = 1, \ldots, N)$

Select the K nearest instances to $\mathbf{x}$: $D^K_{\mathbf{x}}$

Assign to $\mathbf{x}$ the most frequent class in $D^K_{\mathbf{x}}$

END

Figure: Pseudo-code for the basic K-NN classifier

Algorithm for the basic K-NN

Example

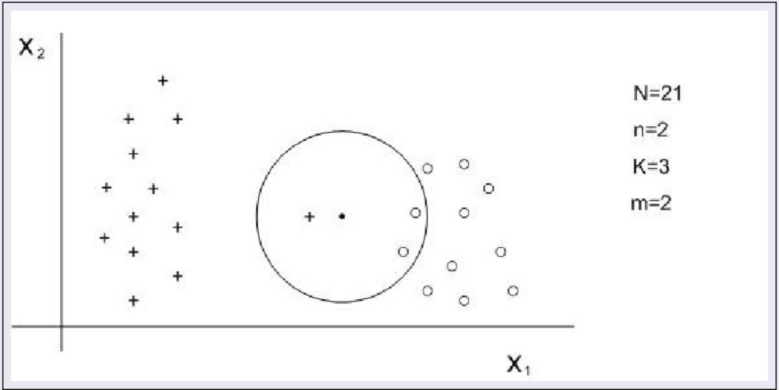

Figure: Example for the basic 3-NN. *m* denotes the number of classes, *n* the number of predictor variables, and *N* the number of labelled cases

Algorithm for the basic K-NN

The accuracy is not monotonic with respect to K

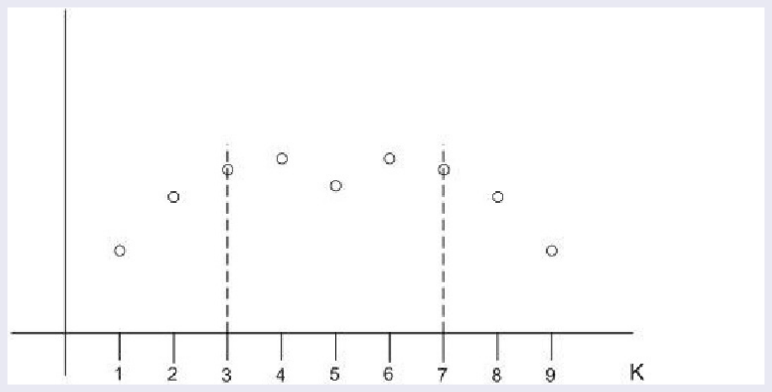

Figure: Accuracy versus number of neighbors

Outline

K-NN with rejection

Requiring for some guarantees

- Demanding for some guarantees before an instance is classified
- In case that the guarantees are not verified the instance remains unclassified
- Usual guaranty: threshold for the most frequent class in the neighbor

K-NN with average distance

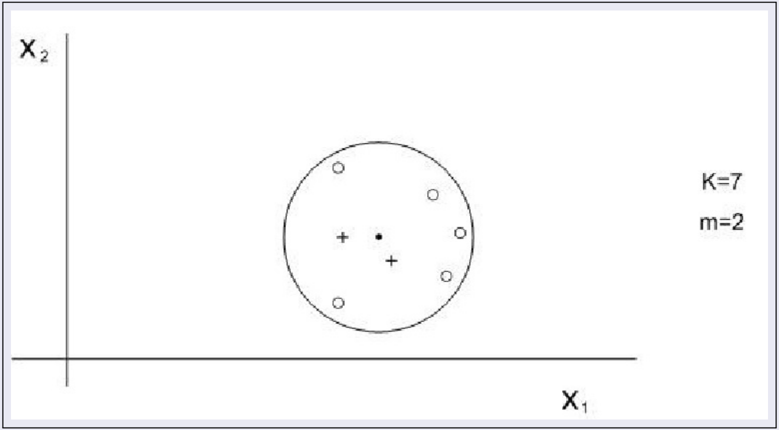

Figure: *K*-NN with average distance

K-NN with weighted neighbors

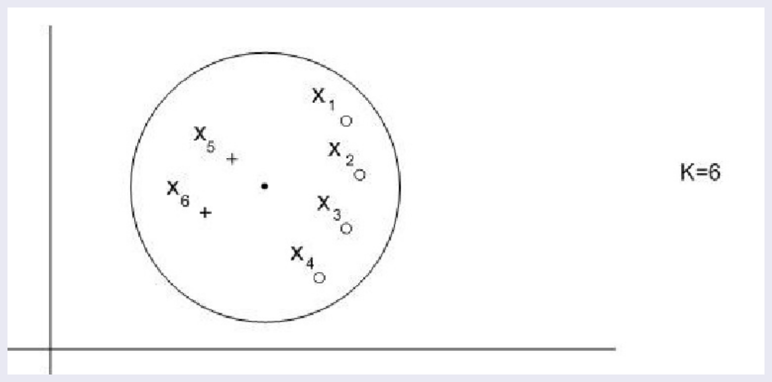

Figure: K-NN with weighted neighbors

K-NN with weighted neighbors

	$d(\mathbf{x}_i, \mathbf{x})$	w_i
$\mathbf{x}_1$	2	0.5
$\mathbf{x}_2$	2	0.5
$\mathbf{x}_3$	2	0.5
$\mathbf{x}_4$	2	0.5
$\mathbf{x}_5$	0.7	$1/0.7$
$\mathbf{x}_6$	0.8	$1/0.8$

Figure: Weight to be assigned to each of the 6 selected instances

K-NN with weighted variables

X_1	X_2	C
0	0	1
0	0	1
0	0	1
1	0	1
1	0	1
1	1	1
0	1	0
0	1	0
0	1	0
1	1	0
1	1	0
1	0	0

Figure: Variable X_1 is not relevant for C

K-NN with weighted variables

$$d(\mathbf{x}_l, \mathbf{x}) = \sum_{i=1}^{n} w_i d_i(x_{l,i}, x_i) \quad \text{with} \quad w_i = MI(X_i, C)$$

$$MI(X_1, C) = p_{(X_1,C)}(0,0) \log \frac{p_{(X_1,C)}(0,0)}{p_{X_1}(0) \cdot p_C(0)} + p_{(X_1,C)}(0,1) \log \frac{p_{(X_1,C)}(0,1)}{p_{X_1}(0) \cdot p_C(1)} +$$

$$p_{(X_1,C)}(1,0) \log \frac{p_{(X_1,C)}(1,0)}{p_{X_1}(1) \cdot p_C(0)} + p_{(X_1,C)}(1,1) \log \frac{p_{(X_1,C)}(1,1)}{p_{X_1}(1) \cdot p_C(1)} =$$

$$\frac{3}{12} log \frac{\frac{3}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{3}{12} log \frac{\frac{3}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{3}{12} log \frac{\frac{3}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{3}{12} log \frac{\frac{3}{12}}{\frac{6}{12} \cdot \frac{6}{12}} = 0$$

$$MI(X_2, C) = p_{(X_2,C)}(0,0) \log \frac{p_{(X_2,C)}(0,0)}{p_{X_2}(0) \cdot p_C(0)} + p_{(X_2,C)}(0,1) \log \frac{p_{(X_2,C)}(0,1)}{p_{X_2}(0) \cdot p_C(1)} +$$

$$p_{(X_2,C)}(1,0) \log \frac{p_{(X_2,C)}(1,0)}{p_{X_2}(1) \cdot p_C(0)} + p_{(X_2,C)}(1,1) \log \frac{p_{(X_2,C)}(1,1)}{p_{X_2}(1) \cdot p_C(1)} =$$

$$\frac{1}{12} log \frac{\frac{1}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{5}{12} log \frac{\frac{5}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{5}{12} log \frac{\frac{5}{12}}{\frac{6}{12} \cdot \frac{6}{12}} + \frac{1}{12} log \frac{\frac{1}{12}}{\frac{6}{12} \cdot \frac{6}{12}}$$

Outline

Wilson edition

Eliminating rare instances

- The class of each labelled instance, $(\mathbf{x}_l, c^{(l)})$, is compared with the label assigned by a K-NN obtained with all instances except itself
- If both labels coincide the instance in maintained in the file. Otherwise it is eliminated

Hart condensation

Maintaining rare instances

- For each labelled instance, and following the storage ordering, consider a K-NN with only the previous instances to the one to be considered
- If the true class and the predicted by the K-NN are the same the instance is not selected
- Otherwise (the true class and the predicted one are different) the instance is selected
- The method depends on the storage ordering

Outline

K-nearest neighbor

- Intuitive and easy to understand
- There is not an explicit model: transduction instead of induction
- Variants of the basic algorithm
- Storage problems: prototype selection

5. K-Nearest Neighbor

Pedro Larrañaga

Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country

Madrid, 25th of July, 2006