CogSci I: Exercise NN

Barbara Plank, University of Copenhagen

September 1, 2014

Exercises

Today we have seen the nearest neighbor classifier. Here are a couple of exercises related to nearest neighbor (NN) and statistics. You can get the code and data from github: `https://github.com/bplank/cog-sci-1-2014` see folder `ex1-nn`.

Algorithm 1 Nearest neighbor classification algorithm

Require: train data set $\mathcal{D}$ of size $M \times N$,
data point $\mathbf{x}$ to classify, distance function $dist$

1: find data point $\mathbf{x}^c$ which is nearest (closest) to $\mathbf{x}$ according to $dist$:

$$(\mathbf{x}^c, y^{min}) = \underset{(\mathbf{x_i}, y_i) \in \mathcal{D}}{\arg\min}\, dist(\mathbf{x}, \mathbf{x^i})$$

2: **return** assign label of $\mathbf{x}^c$ to $\mathbf{x}$: $\hat{y} = y_{min}$

label	f1	f2
0	5.1	3.5
0	4.9	3
0	4.7	3.2
0	4.6	3.1
0	5	3.4
1	6.7	3
1	6.3	2.5
1	6.5	3
1	6.2	3.4
1	5.9	3

Table 1: Data

1. For the data set shown in Table 1 (same as `knntrain.dat` in folder): What is the mean of the feature f_1 over the whole data set, what is the mean of f_2?
2. The following formula denotes the standard deviation for a vector $\mathbf{v}$ with N elements (or dimensions), where μ is the mean of the vector: $\mathbf{v} = \langle v_1, v_2, .., v_N \rangle$:

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(v_i - \mu)^2}$$

What is the standard deviation of f_1 (over the whole data set)?

3. What is the Euclidean distance between the first data point in the data set $x_1 = \langle 5.1, 3.5 \rangle$ and a test data point x_t=⟨5.5,3.2⟩, i.e., $d_{Euc}(x_1, x_t)$? Reminder:

$$dist_{Euc}(p,q) = \sqrt{\sum_{i=1}^{m}(q_i - p_i)^2}$$

4. Implement the k-nearest neighbor algorithm and apply it using Euclidean distance to the new data point x_t=⟨5.5,3.2⟩ given the training data set in Table 1. To do so, extend the code `knn.py` by adding the following (cf., `your code here`):

   - read in the data set `knntrain.dat`
   - in the `predict` function add the code that examines k nearest neighbors (the majority vote function is already given, you need to supply it the list of labels for the k closest neighbors).

   What is the predicted class for x_t=⟨5.5,3.2⟩ if you use one neighbor ($k = 1$) versus more neighbors (e.g., $k = 3$)?

5. What are strengths and weaknesses of the k-nearest neighbor algorithm?

6. (optional) Plot the data set shown in Table 1 (you can do so on paper, or if you prefer, use your preferred plotting environment, e.g., python or R). Where would the new data point x_t=⟨5.5,3.2⟩ be in the plot? Use the plot to check what your implementation gave you for $k = 1$ and $k = 3$ neighbors.