k–Nearest Neighbors

1 k-NN classifier

1. Load the training set and pick a positive integer k.
2. Receive a new example to classify.
3. From the training set, select k examples nearest to the new example according to a chosen metric.
4. Classify the new example as the most frequent class among the nearest neighbors.

Questions

Question 1.

Use the k-NN classifier (with $k = 3$) to classify new examples based on the training sets below.

(a) Training set:

A(1, 3), A(2, 1), A(2, 3), B(4, 3), B(6, 3)

Examples to classify:

- (1, 5)
- (2, 6)
- (3, 4)

(b) Training set:

A(5, 4, 1), A(4, 3, 0), B(1, 2, 3), B(2, 0, 4), C(6, 1, 1), C(5, 0, 1).

Examples to classify:

- (4, 4, 0)
- (1, 1, 5)
- (6, 0, 0)

Mini-project: k-NN

The goal is to implement the k-NN classifier. The program should take 3 arguments:
`k`: positive natural number being the k-NN hyperparameter.
`train-set`: name of the file containing the training set in csv format.
`test-set`: name of the file containing the test set.

Requirements:

- The program should apply k-NN classifier based on the train set to each vector from the test set and produce the accuracy (proportion of correctly classified examples from the test set).
- The program should additionally provide a simple interface (not necessarily graphical) to enable the user to input single vectors to be classified.
- Test the program using training data in `iris.data` and test data in `iris.test.data`.
- **Important:** the program should be able to load any dataset (in a format similar to `iris.data`), with an arbitrary number of dimensions/classes.
- **Optionally:** prepare a graph (excel, python, etc.) showing the accuracy vs the value of k.
- **Optionally:** also classify examples in the *WDBC* dataset provided in the files `wdbc.data` and `wdbc.test.data` [Source].