School of Computer Science
The University of Adelaide

Artificial Intelligence
Assignment 1

Semester 1 2023
Due 11:59pm Wednesday 3 May 2023

1 Wine Quality Prediction with 1NN (K-d Tree)

Wine experts evaluate the quality of wine based on sensory data. We could also collect the features of wine from objective tests, thus the objective features could be used to predict the expert's judgement, which is the quality rating of the wine. This could be formed as a supervised learning problem with the objective features as the data features and wine quality rating as the data labels.

In this assignment, we provide objective features obtained from physicochemical statistics for each white wine sample and its corresponding rating provided by wine experts. You are expect to implement **k-d tree (KDT)**, and use the training set to train your k-d tree, then provide wine quality prediction on the test set by searching the tree.

Wine quality rating is measured in the range of 0-9. In our dataset, we only keep the samples for quality ratings 5, 6 and 7. The 11 objective features are listed as follows [1]:

- f_acid: fixed acidity
- v_acid: volatile acidity
- c_acid: citric acid
- res_sugar: residual sugar
- chlorides: chlorides
- fs_dioxide: free sulfur dioxide
- ts_dioxide: total sulfur dioxide
- density: density
- pH: pH

- sulphates: sulphates
- alcohol: alcohol

Explanation of the Data.
`train`: The first 11 columns represent the 11 features and the 12th column is the wine quality. A sample is depicted as follows:

f_acid	v_acid	c_acid	res_sugar	chlorides	fs_dioxide	ts_dioxide	density	pH	sulphates	alcohol	quality
8.10	0.270	0.41	1.45	0.033	11.0	63.0	0.99080	2.99	0.56	12.0	5
8.60	0.230	0.40	4.20	0.035	17.0	109.0	0.99470	3.14	0.53	9.7	5
7.90	0.180	0.37	1.20	0.040	16.0	75.0	0.99200	3.18	0.63	10.8	5
8.30	0.420	0.62	19.25	0.040	41.0	172.0	1.00020	2.98	0.67	9.7	5
6.50	0.310	0.14	7.50	0.044	34.0	133.0	0.99550	3.22	0.50	9.5	5

`test`: Similar to `train`, but without the 12th colum as they are the values your model will predict. A sample is depicted as follows:

f_acid	v_acid	c_acid	res_sugar	chlorides	fs_dioxide	ts_dioxide	density	pH	sulphates	alcohol
7.0	0.360	0.14	11.60	0.043	35.0	228.0	0.99770	3.13	0.51	8.900000
6.3	0.270	0.18	7.70	0.048	45.0	186.0	0.99620	3.23	0.47	9.000000
7.2	0.290	0.20	7.70	0.046	51.0	174.0	0.99582	3.16	0.52	9.500000
7.1	0.140	0.35	1.40	0.039	24.0	128.0	0.99212	2.97	0.68	10.400000
7.6	0.480	0.28	10.40	0.049	57.0	205.0	0.99748	3.24	0.45	9.300000

1.1 1NN (K-d Tree)

From the given training data, our goal is to learn a function that can predict the wine quality rating of a wine sample, based on the objective features. In this assignment, the predictor function will be constructed as a k-d tree. Since the attributes (objective features) are continuously valued, you shall apply the k-d tree algorithm for continuous data, as outlined in Algorithms 1. It is the same as taught in the lecture. Once the tree is constructed, you will search the tree to find the 1-nearest neighbour of a query point and label the query point. Please refer to the search logic taught in the lecture to write your code of 1NN search.

Algorithm 1 BuildKdTree(P, D)

```
Require: A set of points P of M dimensions and current depth D.
if P is empty then
    return null
else if P only has one data point then
    Create new node node
    $node.d \leftarrow d$
    $node.val \leftarrow val$
    $node.point \leftarrow$ current point
    return node
else
    $d \leftarrow$ D mod M
    $val \leftarrow$ Median value along dimension among points in P.
    Create new node node.
    $node.d \leftarrow d$
    $node.val \leftarrow val$
    $node.point \leftarrow$ point at the median along dimension d
    $node.left \leftarrow$ BuildKdTree( points in P for which value at dimension $d$ is less than or equal to $val$, D+1)
    $node.right \leftarrow$ BuildKdTree( points in P for which value at dimension $d$ is greater than $val$, D+1)
    return node
end if
```

Note: Sorting is not necessary in some cases depending on your implementation. Please figure out whether your code needs to sort the number first. Also, if you compute the median by yourself, when there's an even number of points, say [1,2,3,4], the median is 2.5.

1.2 Deliverable

Write your k-d tree program in Python 3.6.9 in a file called `nn_kdtree.py`. Your program must be able to run as follows:

```
$ python nn_kdtree.py [train] [test] [dimension]
```

The inputs/options to the program are as follows:

- `[train]` specifies the path to a set of the training data file.
- `[test]` specifies the path to a set of testing data file.

- `[dimension]` is used to decide which dimension to start the comparison. (Algorithm 1).

Given the inputs, your program must construct a k-d tree (following the prescribed algorithms) using the training data, then predict the quality rating of each of the wine sample in the testing data. Your program must then **print to standard output** (i.e., the command prompt) the list of predicted wine quality ratings, vertically based on the order in which the testing cases appear in `[test]`.

1.3 Python libraries

You are allowed to use the Python standard library to write your k-d tree learning program (see `https://docs.python.org/3/library/` for the components that make up the Python v3.6.9 standard library). In addition to the standard library, you are allowed to use NumPy and Pandas. Note that the marking program will not be able to run your program to completion if other third-party libraries are used. You are NOT allowed to use implemented tree structures from any Python package, otherwise the mark will be set to 0.

1.4 Submission

You must submit your program files on Gradescope. Please use the course code **6ZWNXV** to enrol into the course. Instructions on accessing Gradescope and submitting assignments are provided at `https://help.gradescope.com/article/5d3ifaeqi4-student-canvas`. **For undergraduates**, please submit your k-d tree program (`nn_kdtree.py`) to **Assignment 2 - UG**. If there are any questions or issues with Gradescope, please contact Ankit Yadav via email at ankit.yadav@adelaide.edu.au.

1.5 Expected run time

Your program must be able to terminate within 600 seconds on the sample data given.

1.6 Debugging Suggestions

Step-by-step debugging by checking intermediate values/results will help you to identify the problems of your code. This function is enabled by most of the Python IDE. If not in your case, you could also print the intermediate values out. You could use sample data or create data in the same format for debugging.

1.7 Assessment

I will compile and run your code on several test problems. If it passes all tests, you will get **15%** (undergrads) or **12%** (postgrads) of the overall course mark. **For undergraduates**, bonus marks of **3%** will be awarded if Section 2 is completed correctly.

There will be no further manual inspection/grading of your program to award marks on the basis of coding style, commenting or "amount" of code written.

1.8 Using other source code

You may not use other source code for this assignment. All submitted code must be your own work written from scratch. Only by writing the solution yourself will you fully understand the concept.

1.9 Due date and late submission policy

This assignment is due by 11:59pm Wednesday 3 May 2023. If your submission is late, the maximum mark you can obtain will be reduced by 25% per day (or part thereof) past the due date or any extension you are granted.

Continues next page for postgraduate section.

2 Wine Quality Prediction with Random Forest

For postgraduate students, completing this section will give you the remaining **3%** of the assignment marks.

In this task, you will extend your knowledge learned from k-d tree to k-d forest. The process for a simplified k-d forest given N input-output pairs is:

(1) Randomly select a set of N' distinct samples (i.e., no duplicates) where $N' = N *$ 80% (round to integer). This dataset is used for constructing a k-d tree (i.e., the root node of the k-d tree).

(2) Build a k-d tree on the dataset from (1) and apply Algorithm 1.

(3) Repeat (1) and (2) until reaching the maximum number of trees.

This process is also shown in Algorithm 2. In k-d forest learning, a sample set is used to construct a k-d tree. That is to say, different trees in the forest could have different root data. For prediction, the k-d forest will choose the most voted label as its prediction.

For the wine quality prediction task, you shall apply Algorithm 2 for k-d forest learning and apply Algorithm 3 to predict the wine quality for a new wine sample.

To generate samples, please use the following (incomplete) code to generate the same samples as our testing scripts:

```
import random
...
N= ...
N’=...
index_list = [i for i in range(0, N)]  # create a list of indexes for all data
sample_indexes = []
for j in range(0,n_tree):
    random.seed(rand_seed+j)  #  random_seed is one of the input parameters
    subsample_idx = random.sample(index_list, k=N’) # create unique N’ indices
    sample_indexes = sample_indexes + subsample_idx
```

Algorithm 2 KdForest(*data, d_list, rand_seed)*

```
Require: data in the form of N input-output pairs $\{\mathbf{x}_i, y_i\}_{i=1}^{N}$, d_list a list of depths.
$forest \leftarrow []$
$n\_trees \leftarrow len(d\_list)$
$sample\_indexes \leftarrow N' * n\_trees$ integers with value in $[0, N)$ generated by using above method
$count \leftarrow 0$
for $count < n\_trees$ do
    $sampled\_data \leftarrow N'$ data pairs selected by $N'$ indexes from sample_indexes sequentially
    $n$ = BuildKdTree(sampled_data, d_list[count]) $\Longrightarrow$ Algorithm 1
    $forest.append(n)$
end for
return $forest$
```

Algorithm 3 Predict_KdForest($forest, data$)

```
Require: $forest$ is a list of tree roots, data in the form of attribute values $\mathbf{x}$.
$labels \leftarrow []$
for Each tree $n$ in the $forest$ do
    $label \leftarrow$ 1NN search on tree $n$
    $labels.append(n)$
end for
return the most voted label in labels
```

2.1 Deliverables

Write your random forest program in Python 3.6.9 in a file called `nn_kdforest.py`. Your program must be able to run as follows:

```
$ python nn_kdforest.py [train] [test] [random_seed] [d_list]
```

The inputs/options to the program are as follows:

- `[train]` specifies the path to a set of training data file.
- `[test]` specifies the path to a set of testing data file.
- `[random_seed]` is the seed value generate random values.
- `[d_list]` is a list of depth values (In Algorithm 2 n_trees==len(d_list)).

Given the inputs, your program must learn a random forest (following the prescribed algorithms) using the training data, then predict the quality rating of each wine sample in the testing data. Your program must then **print to standard output** (i.e., the command prompt) the list of predicted wine quality ratings, vertically based on the order in which the testing cases appear in `[test]`.

Submit your program in the same way as the submission for Sec. 1. **For postgraduates**, please submit your learning programs (`nn_kdtree.py` and `nn_kdforest.py`) to **Assignment 2 - PG**. The due date, late submission policy and code reuse policy are also the same as in Sec. 1.

2.2 Expected run time

Your program must be able to terminate within 600 seconds on the sample data given.

2.3 Debugging Suggestions

In addition to Sec. 1.6, another value worth checking when debugging is (but not limited to):

- the *sample_indexes* – by setting random seed, the indexes should be the same each time you run the code

2.4 Assessment

I will compile and run your code on a single test case. If it passes, you will get **3%** of the overall course mark.

∼∼∼ The End ∼∼∼

References

[1] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems 47*, 4 (2009), 547–553.