Lab 3 - Nearest Neighbors for Missing Data

CS 207

Due: Monday, February 29th at 11:59pm

1 Summary

In short, your goals for this lab are to:

1. make a kd-tree and use it for nearest neighbor prediction.

2. add the ability to fill in missing data based on the nearest neighbor to your data cleaning script.

3. do this for your project data and submit the result.

4. provide an analysis website of what you found.

This will all be in python and have no visualization component.

This should all be completed in the lab3_nearestneighbors directory. You will need to copy your lab1 work as well as your project data into the directory.

2 Prediction using nearest neighbors

Your goal will be to use nearest neighbors to predict / fill in missing values of a given point. You’ll need to create a kd-tree implementation to answer nearest neighbor queries.

Once you have implemented nearest neighbor queries, be sure to do some example queries and consider the quality of those results. Write a paragraph of analysis about what you find and put it in the index.html file. You should include an analysis of how the results of using nearest neighbors to fill in missing values differs from the other methods you used in lab 1.

You will likely have the easiest time of this if you do your nearest neighbor implementation in python, integrating it with your data cleaning script from lab 1. Here are some sub-task suggestions to get you started.
1. Create a class `Point` that can calculate distance to another `Point`.

2. Create a class `Node` that stores a `Point` at its root and has pointers to left and right children.

3. Create a median finding algorithm (it’s ok to sort and then take the median) that takes a list of `Points` and a dimension and returns the median `Point` from that dimension as well as a list of the `Points` before the median and a list of `Points` after the median (neither of which include the median). If you use this on a large dataset and it’s taking awhile, you might want to do what we discussed in class and sample from the dataset to find a point close to the median and not the exact median.

4. Create a `KDTree` class that can create a kd-tree when given a list of `Points`. For debugging purposes, you may want to add the functionality to print out the tree as well.

3 Extra Credit

You should absolutely not start this until the main portion of the lab has been completed and tested.

There are a couple of options for extra credit. If you do either option, you should indicate this clearly in your `.html` file and write a paragraph about how (if at all) it changed your results.

1. Implement approximate nearest neighbor searching and use it for prediction.

2. Implement k-nearest neighbor searching and use it for prediction.