Project: 2 CS 205. Introduction to Artificial Intelligence

Due Date: Thursday of finals week at 11:59pm

Feature Selection with Nearest Neighbor

This is a team project; you need to work in teams of size two. In brief, the project asks you to:

1. Implement Nearest Neighbor
2. Implement Feature Search (using Nearest Neighbor)
3. Process three synthetic datasets (of increasing sizes)
4. Process a real-world classification dataset
5. Write a report.

As we have seen in class this quarter, the nearest neighbor algorithm is a very simple, yet very competitive classification algorithm. It does have one major drawback however; it is very sensitive to irrelevant features. With this in mind you will code up the nearest neighbor classifier, and then use it inside a "wrapper" which does various kinds of searches (listed below)

1) Forward Selection
2) Backward Elimination

Don't be scared by the phrase "search algorithm", in this case it is really simply nested loops, nothing else.

To make life simple, you can assume the following. I will only give you datasets that have two classes. I will only give you datasets that have continuous features.
Think carefully before you start coding this. Students in the past seem to have made this more complicated than it need be. In particular, in Matlab I was able to write the nearest neighbor algorithm in 8 lines of code, and the two search algorithms in another 17 lines of code.

C++ and Java programs tend to be longer, but even so, I would be surprised if this took more than 100 lines of code (although I will not penalize you for this).

Very important: Make sure your nearest neighbor algorithm is working correctly before you attempt the search algorithms. We will provide some test datasets for this purpose.

You may use some predefined utility routines, for example sorting routines. However I expect all the major code to be original. You must document any book, webpage, person or other resources you consult in doing this project (see the first day's handout).

You may consult colleagues at a high level, discussing ways to implement the tree data structure for example. But you may **not** share code. At most, you might illustrate something to a colleague with pseudocode.

You will hand in a report, see Project_2_sample_report.pdf

You must keep the evolving versions of your code, so that, if necessary, you can demonstrate to the course staff how you went about solving this problem (in other words, we may ask you to prove that you did the work, rather than copy it from somewhere).

You can use a simple text line interface or a more sophisticated GUI (but don't waste time making it pretty unless you are sure it works and you have lots free time). However, your program should have a trace like the one below, so that it can be tested.

The data files will be in the following format. ASCII Text, IEEE standard for 8 place floating numbers. This is a common format; you should be able to find some code to load the data into your program, rather than writing it from scratch (as always, document borrowed code). The first column is the class, these values will always be either "1"s or "2"s. The other columns contain the features, which are **not** normalized. There may be an arbitrary number of features (for simplicity I will cap the maximum number at 64). There may an arbitrary number of instances (rows), for simplicity I will cap the maximum number at 2,048. Below is a trivial sample dataset. The first record is class "2", the second is class "1" etc. This example has just two features.

```
2.0000000e+000  1.2340000e+010  2.3440000e+000
1.0000000e+000  6.0668000e+000  5.0770000e+000
2.0000000e+000  2.3400000e+010  3.6460000e+000
1.0000000e+000  4.5645400e+010  3.0045000e+000
```

Datasets are here https://www.dropbox.com/sh/3jc2hzyotau7inf/AADCTOgQSA_afOkV_NRJ1HRoa?dl=0

You need to work on three files of different sizes

You are working in teams of two.

The **small** dataset you should use is the day of the month for the birthday of the younger member.
So, for John (04/04/1969) and Mary (15/12/2001) that would be CS170_small_Data__15.txt

The **large** dataset you should use is the day of the month for the birthday of the older member.
So, for John (04/04/1969) and Mary (15/12/2001) that would be CS170_large_Data__4.txt

The **Xlarge** dataset you should use is sum of the months for both team members .
So, for John (04/04/1969) and Mary (15/12/2001) that would be (4+12): CS170_XXXlarge_Data__**16**.txt

Here are the solutions for some files, so you can check your work

On small dataset 32 the error rate can be about 0.954 when using only features 3 1 5
On small dataset 33 the error rate can be about 0.949 when using only features 8 7 3
On large dataset 32 the error rate can be about 0.963 when using only features 3 7 6
On large dataset 33 the error rate can be about 0.9655 when using only features 4 5 10

Important note: You will find **Xlarge** too computationally demanding to fully process. In class, I will give you some hints on how to process this, including

- Accelerating the code by early abandoning
- Casting as an Anytime Algorithm
- Sampling
- etc

The output should look like this

Welcome to Bertie Woosters Feature Selection Algorithm.
Type in the name of the file to test : **eamonns_test_2.txt**

Type the number of the algorithm you want to run.

1) Forward Selection
2) Backward Elimination

1

This dataset has 4 features (not including the class attribute), with 345 instances.

Running nearest neighbor with all 4 features, using "leaving-one-out" evaluation, I get an accuracy of 75.4%

Beginning search.

Using feature(s) {1} accuracy is 45.4%
Using feature(s) {2} accuracy is 63.7%
Using feature(s) {3} accuracy is 71.4%
Using feature(s) {4} accuracy is 48.1%

Feature set {3} was best, accuracy is 71.4%

Using feature(s) {1,3} accuracy is 48.9%
Using feature(s) {2,3} accuracy is 70.4%
Using feature(s) {4,3} accuracy is 78.1%

Feature set {4,3} was best, accuracy is 78.1%

Using feature(s) {1,4,3} accuracy is 56.9%
Using feature(s) {2,4,3} accuracy is 73.4%

(Warning, Accuracy has decreased! Continuing search in case of local maxima)
Feature set {2,4,3} was best, accuracy is 73.4%

Using feature(s) {1,2,4,3} accuracy is 75.4%

Finished search!! The best feature subset is {4,3}, which has an accuracy of 78.1%

Processing a real-world classification dataset

After you have finished the synthetic datasets

Go to:

1. https://archive.ics.uci.edu/ml/datasets.php
   or
2. Kaggle.com (or similar)

Find a real-world classification dataset. Ideally find a dataset in a domain that you have some intuition for (or one you can find information about fairly easily)

Not all the datasets in those places exact fit our format, but a large fraction of them do.

Run the search algorithms discussed above, and write a report summarizing your results.

For example (I just made-up numbers and facts here)

I did forward search on Auto MPG Data Set, this dataset was taken from the StatLib library which is maintained at Carnegie Mellon University. The dataset was used in the 1983 American Statistical Association Exposition.
This dataset is a slightly modified version of the dataset provided in the StatLib library. In line with the use by Ross Quinlan (1993) in predicting the attribute "mpg", 8 of the original instances were removed because they had unknown values for the "mpg" attribute. The original dataset is available in the file "auto-mpg.data-original".

"The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes."

Feature Information:

1. mpg: continuous
2. cylinders: multi-valued discrete
3. displacement: continuous
4. horsepower: continuous
5. weight: continuous
6. acceleration: continuous
7. model year: multi-valued discrete
8. origin: multi-valued discrete
9. car name: string (unique for each instance)

The original MPG feature was continuous. In order to make it a class label, I discretize it in the following manner:

- If the MPG was 0 to 10, then class label 1
- If the MPG was 11 to 24, then class label 2
- If the MPG was 25 to inf, then class label 3

So now I have a three-class problem to solve, and not the original problem that was defined for this dataset.

When I ran forward search, the first feature chooses was horsepower. This makes sense, as horsepower is closed related to MPG (cars that have a high horsepower tend to use a lot of gas). The next feature chosen was…. Lorem ipsum dolor sit amet, mandamus constituto liberavisse ei quo, mel ea congue facilis dolores. Sit id quis nostrum. At dolorem noluisse eum. Doming theophrastus comprehensam per eu.

There were three features that were not choose by search. Next I will explain why I think they were not selected. The first was color, it is clear that a cars color is not related to its MPG, mandamus constituto liberavisse ei quo, mel ea congue facilis dolores. Sit id quis nostrum. At dolorem noluisse eum. Doming theophrastus comprehensam per eu.

Some important notes:
You may have to clean up the dataset in a few ways.

1. You will need to normalize continuous values (either 0/1 normalization or z-normalization)
2. If you have mostly continuous features, but a few categorical features (say, *married, single, divorced*). Then you should just delete the categorical features (you *can* do nearest neighbor with categorical features, but that is too messy for this project).
3. Trivial: many dataset have the class label in the last column, not the first column like we are used to.
4. The class labels might be categorical, but you expect integers, so you can recode them: Jewish = 1, Hindu = 2, etc

You need to explain all these steps in your report.