Lab 3: Supervised Learning: More Classification Algorithms

Due date: Monday, October 22, 11:59pm.

Lab Assignment

This lab continues your work on development of classification algorithms. We are asking you to complete the implementation of the `C4.5` algorithm so that it properly processes numeric attributes, and test the work on this algorithm on at least one dataset (Iris). Additionally, we are asking you to build an implementation of the Random Forest classifier based on your `C4.5` implementation, and compare the accuracy of the Random Forest on the available datasets to the accuracy of `C4.5` alone. Finally, we are asking you to implement the K Nearest Neighbors classifier.

Assignment Preparation

Continue working with your Lab 2 partner on this assignment, as you will be adding to the code base built during lab 2. (Note, you will be required to switch you partner for Lab 4).

Datasets

We will be using a number of datasets from the UCI (Univeristy of California at Irvine) Machine Learning Dataset Repository. Each of the datasets has a long history of being used for classification.

Below is a list of the datasets provided with brief description of each. Please note, for most datasets, a file with additional information is also made available. In the interest of saving space in this document, I will not repeat the information found in those files - you should consult them in order to determine the variables provided to you and their domains.

All datasets can be downloaded from the Lab 3 data page:

http://users.csc.calpoly.edu/∼dekhtyar/466-Spring2018/labs/lab02.html

We also provide links to the UCI Repository pages for each dataset. Please note, that in some cases, we have changed file names, and, modified the files themselves to make parsing easier (You can, however, use any files as input). You are also allowed to modify the format of input files to your liking (adding or removing header lines, etc...)

Iris Dataset

The Iris dataset is one of the most popular Machine Learning datasets. It is a simple dataset containing a few hundred records. Each record depicts physical dimensions of a specific iris flower from one of three different subspecies of iris: *Iris Setosa*, *Iris Versicolor*, or *Iris Virginica*. There are four physical parameters measured for each flower, listed below in the order in which they occur in the data file:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm
- Petal width in cm

The dataset is available from the UCI Machine Learning repository. Lab 2 data page,

http://users.csc.calpoly.edu/∼dekhtyar/466-Spring2018/labs/lab02.html

contains the links to the data page and the data file.

Letter Recognition Dataset

The Letter Recognition dataset[1] contains 20,000 records each describing 16 features extracted from an image of a letter. The class variable is the letter itself (26 letter of latin alphabet). The data file, `letter-recognition.data.csv` contains 17 columns. The first column is the letter, the remaining columns are the 16 extracted features (described in the `letter-recognition.names.txt` files). All features are numeric (integer).

Wine Quality Dataset

The Wine Quality dataset[2] contains information about the chemical properties of around 4900 red and white varieties of Portuguese Vinho Verde wine.

[1] https://archive.ics.uci.edu/ml/datasets/Letter+Recognition.
[2] https://archive.ics.uci.edu/ml/datasets/Wine+Quality

Each wine is described by 11 numeric features (such as acidity, citric acid, residual sugar, etc..), followed by the class variable describing the quality of the wine. The class variable takes values 0,1,. . . , 10. The goal is to predict from the chemistry of the wine its quality.

Note that there are two separate datasets, `winequality-red-fixed.csv` and `winequality-white-fixed.csv` corresponding to red and white wine collections. Because red and white wines have very different chemical profiles, treat these two datasets as independent and separate, i.e., build separate classifiers for red and white wine.

Credit Approval Dataset

The Credit Approval dataset [3] contains information about credit approval decisions of 690 individuals. Due to the sensitive nature of the data, not only are the identities of the individuals hidden, but the 15 features describing the credit applicants and their applications are obscured. The features are named A1 through A15, feature A16 is the class variable taking two values: `"+"` for approved application and `"-"` for rejected application. The independent variables (features) are a mix of numeric (continuous) variables and categorical variables. The domains of all variables are also abstracted, and are specified in the `crx.names.txt` file.

Your goal is to predict whether the credit application is approved or rejected.

Seeds Dataset

The Seeds dataset [4]shares a certain similarity with the Iris dataset. The dataset is comprised of 210 descriptions of individual seeds from three varieties of wheat: Kama, Rosa, and Canadian. The data file `seeds_dataset.csv` has seven numeric (continuous, real) features (area, perimeter, compactness, length of kernel, width of kernel, assymetry coefficient, length of kernel groove, in that order) describing each seed. The eights column in the dataset, with values 1, 2, or 3 is the class variable describing the variety of wheat.

Your goal is to predict the variety of wheat based on the physical characteristics of the seed.

Mushroom Dataset

The Mushroom dataset[5] contains over 8,000 descriptions of physical appearance of gilled mushrooms from 23 species of the *Agaricus and Lepiota* family. Some species are edible, some are poisonous or not edible. The

[3] https://archive.ics.uci.edu/ml/datasets/Credit+Approval
[4] https://archive.ics.uci.edu/ml/datasets/seeds
[5] https://archive.ics.uci.edu/ml/datasets/Mushroom

dataset, stored in the `agaricus-lepiota.data.csv` file has 22 categorical attributes describing the physical characteristics of each mushroom (e.g., cap size, cap color, odor, gill size, stalk size, veil type, and so on). The `agaricus-lepiota.names.txt` file contains the full description of each column, including all possible values. The first attribute in the data file is the class variable which takes two values: `"p"` for poisonous/not edible mushrooms, and `"e"` for edible mushrooms.

Your goal is to predict whether a given mushroom is edible or poisonous/not edible.

Task 1: Complete C4.5 implementation

Your first task is to complete your `C4.5` implementation by adding the functionality to classify on numeric attributes. A number of datasets presented to you in the lab contain numeric attributes, making it impossible for you to complete the rest of the lab assignment unless your `C4.5` algorithm implementation handles numeric attributes properly.

Task 2: Random Forset implementation

Implement the Random Forest classifier. Your implementation shall behave as follows.

Input Parameters. Your implementation shall take as input the following parameters:

- m or NumAttributes: this parameter controls how many attributes each decision tree built by the Random Forest classifier shall contain.
- k or NumDataPoints: the number of data points selected randomly with replacement to form a dataset for each decision tree.
- N or NumTrees: the number of the decision trees to build.

Dataset Selection. Write a method/function that given a full dataset D and the parameters NumAttributes and NumDataPoints selects the appropriate number of data points, and randomly selects NumAttributes and returns the constructed set back. This functionality will be used by your Random Forest classifier.

Behavior. Your Random Forest implementation shall take as input a training set D, and the three input parameters described above. It shall produce the requisite number of small decision trees, as guided by the input parameters. Each decision tree shall be produced by an appropriate call to your `C4.5` implementation.

Evaluation. You will use 10-fold cross-validation to compute the accuracy of the Random Forest classifier on the datasets you select.

The program. Your Random Forest implementation shall be named randomForest.py or randomForest.java (or a similar file name for a programming language of your choice). It shall take as input the dataset filename, and the three input parameters described above. It shall produce, as output, a `results.txt` or `results.csv` file that produces predictions for each individual row in the dataset based on the 10-fold cross-validation evaluation. Separately, it shall also output the confusion matrix and the accuracy of prediction.

Task 3: K Nearest Neighbors

Your final task is to implement the KNN classifier. This classifier shall take one parameter, K - the number of neighbors to consider. Please note, that because KNN is a lazy classifier, there is no training step. Because of this, your implementation shall consist of a single program (`knn.py` or `knn.java`) which takes as input a dataset file, the parameter K, and any additional arguments necessary for your program to properly parse and evaluate the dataset.

The program shall produce the prediction for each input data point, and format the output in the same way as the outputs of your C4.5 and Random Forest classification programs.

For numeric attributes, implement standard distance measures (Manhattan, Eucledian distance). If you do not like their performance, you can look into cosine similarity measure.

For categorical attributes, you can choose either of the solutions we discussed. You can implement any of the similarity measures for categorical attributes from the handout, or you can convert categorical variables into numeric using one hot encoding or other dummification procedures. If implementing multiple distance/similarity functions, make sure that they can be triggered by an input parameter to your program.

Evaluation and Report

With the three implemented methods you shall conduct an evaluation study. This study has two goals:

1. Find the best hyperparameter values for each of the three classifiers for each of the chosen datasets.
2. Compare the accuracy of the three classifiers to each other for each of the chosen datasets.

Dataset use. You shall conduct your study using **no less** than three different datasets (Wine Evaluation counts as one dataset, although you have to build to classifier models for it). You must use at least one dataset that contains numeric attributes, and at least one dataset that uses categorical attributes.

Report. Write and submit a report documenting your findings. In your report, describe the details of your implementation of C4.5, Random Forest and KNN classifiers, provide the description of your evaluation procedures for each of the methods, and describe the comparative study of your implementations and the results. Discuss your findings.

More specifically, for your classifier evaluation procedures include the following information:

- For each dataset, specify which hyper-parameters you were tuning, and what values of the hyper-parameters you were considering.
- For each dataset, provide the observed accuracy results for your study in tabular, and (if you can) graphical form.
- For each dataset, include the discussion of what how the tuning went, and how the best model performed.

For your comparative study,

- Specify exactly which hyper-parameter settings for each method you used in your comparison.
- Describe the cross-validation procedure you used.
- Provide results in tabular and (if you can) graphical form.
- Provide the analysis of the accuracy of your methods on each of the datasets.
- Provide overall analysis of your methods - which tended to perform better on all datasets, and which tended to perform worse?

Submission Instructions

The following is an **updated** list of deliverables. New deliverables are *in italics*.

- `README`. Shall contain the names and email addresses of all students in the team. Also, put any specific instructions and notes in this file. (e.g., if you choose a different implementation language, include translation/running instructions). *Include any instructions on how to run your classifers.*

- All your programs implementing C4.5, Random Forest and KNN classifiers.
- Outputs of your classifiers on the datasets you chose for the best values of hyper-parameters (include the list of output files in your README file with appropriate hyperparameter values)
- Your written report submitted as a PDF document named `Lab3-report.pdf`.

Submit the `README` file and your report as separate files. Submit the rest of your files in a single archive named either `lab03.zip` or `lab03.tar.gz`.

To submit use the following command:

```
$ handin dekhtyar lab03 <files>
```