CS7641 Assignment 1
Supervised Learning

1 Assignment Weight

The assignment is worth 15% of the total points.

Read everything below carefully as this assignment has changed term-over-term.

2 Objective

The purpose of this project is to explore techniques in supervised learning. It is important to realize that understanding an algorithm or technique requires understanding how it behaves empirically under a variety of circumstances. As such, rather than implement each of the algorithms, you will be asked to experiment with them and compare their performance. This is quite involved and also possibly quite different from what you are used to; however, it is central and in many ways the essence of supervised learning.

3 Procedure

First, you should design two interesting classification problems. For the purposes of this assignment, a classification problem is just a set of training examples and a set of test examples. You can download data, take from your own research, or make up your own. Be careful about the datasets you choose, though. You'll need to explain why they are interesting, use them in later assignments, and have a deep understanding of them.

After selecting two interesting classification problems, you will go through the process of exploring the data, tuning the algorithms you've learned about, and writing a thorough analysis of your findings. You need not implement any learning algorithm yourself; however, you must participate in the journey of exploring, tuning, and analyzing. Concretely, this means:

- You may program in any language you wish and are allowed to use any library, **as long as it was not written specifically to solve this assignment**.
- TAs must be able to recreate your experiments on a standard linux machine if necessary.
- The analysis you provide in the report is paramount.

You should experiment with five learning algorithms on each dataset. They are:

- **Decision Trees**. Be sure to use some form of pruning. You are not required to use information gain (for example, there is something called the GINI index that is sometimes used) to split attributes, but you should describe whatever it is that you do use.
- **Neural Networks**. You may use networks of nodes with as many layers as you like and any activation function you see fit.
- **Boosted Decision Trees**. As with decision trees, you will want to use some form of pruning. Since you are using boosting you can afford to be much more aggressive about your pruning.
- **Support Vector Machines**. Make sure to try at least two different kernel functions.
- **k-Nearest Neighbors**. Make sure to try different values of k.

Each algorithm is described in detail in your textbook, the assigned readings on Canvas, and on the internet. Instead of implementing the algorithms yourself, you should use libraries that do this for you and make sure to provide proper attribution. Also, note that you'll need to do some fiddling to obtain good results and graphs, and this might require you to modify these libraries in various ways.

3.1 Experiments and Analysis

Your report should contain:

- A description of your classification problems, and why you feel they are interesting. Think hard about this. To be interesting the problems should be non-trivial on the one hand, but capable of admitting comparisons and analysis of the various algorithms on the other. Avoid the mistake of working on the largest most complicated and messy dataset you can find. The key is to be interesting and clear, no points for hairy and complex.
- The training and testing error rates you obtained running the various learning algorithms on your problems. At the very least you should include graphs that show performance on both training and test data as a function of training size (note that this implies that you need to design a classification problem that has more than a trivial amount of data) and – for the algorithms that are iterative – training times/iterations. Both of these kinds of graphs are referred to as learning curves.
- Graphs for each algorithm showing training and testing error rates as a function of selected hyperparameter ranges. This type of graph is referred to as a model complexity graph (also sometimes validation curve). Please experiment with more than one hyperparameter and make sure the results and subsequent analysis you provide are meaningful.
- Analyses of your results. Why did you get the results you did? Compare and contrast the different algorithms. What sort of changes might you make to each of those algorithms to improve performance? How fast were they in terms of wall clock time? Iterations? Would cross validation help? How much performance was due to the problems you chose? Which algorithm performed best? How do you define best? Be creative and think of as many questions you can, and as many answers as you can.

Analysis writeup is limited to 10 pages.

Please keep your analysis as short as possible while still covering the requirements of the assignment.

3.2 Acceptable Libraries

Here are a few **examples** of acceptable libraries. You can use other libraries as long as they fulfill the conditions mentioned above.

Machine learning algorithms:

- scikit-learn (python)
- Weka (java)
- e1071/nnet/random forest(R)
- ML toolbox (matlab)
- tensorflow/pytorch (python)

Plotting:

- matplotlib (python)
- seaborn (python)
- yellowbrick (python)
- ggplot2 (R)

4 Submission Details

You must submit:

- A file named README.txt containing instructions for running your code (see note below)
- A file named yourgtaccount-analysis.pdf containing your writeup (GT account is what you log in with, not your all-digits ID)

Note: we need to be able to get to your code and your data. Providing entire libraries isn't necessary when a URL would suffice; however, you should at least provide any files you found necessary to change and enough support and explanation so we can reproduce your results on a standard linux machine.

5 Rescoring Criteria

When your assignment is scored, you will receive feedback explaining your errors and successes in some level of detail. This feedback is for your benefit, both on this assignment and for future assignments. It is considered a part of your learning goal to internalize this feedback.

If you are convinced that your score is in error in light of the feedback, you may request a rescore within a week of the score and feedback being returned to you. A rescore request is only valid if it includes an explanation of where the grader made an error.

It is important to note that because we consider your ability to internalize feedback a learning goal, we also assess it. This ability is considered 10 percent of each assignment. We default to assigning you full credit. If you request a rescore and do not receive at least 5 points as a result of the request, you will lose those 10 points.

6 Plagiarism and Proper Citation

The easiest way to fail this class is to plagiarize. **Using the analysis, code or graphs of others in this class is considered plagiarism**. The assignments are designed to force you to immerse yourself in the empirical and engineering side of ML that one must master to be a viable practitioner and researcher. It is important that you understand why your algorithms work and how they are affected by your choices in data and hyperparameters. The phrase "as long as you participate in this journey of exploring, tuning, and analyzing" is key. We take this very seriously and you should too.

What is plagiarism?

If you copy any amount of text from other students, websites, or any other source without proper attribution, that is plagiarism. The most common form of plagiarism is copying definitions or explanations from wikipedia or similar websites. We use an anti-cheat tool to find out which parts of the assignments are your own and there is a near 100 percent chance we will find out if you copy or paraphrase text or plots from online articles, assignments of other students (even across sections and previous courses), or website repositories.

What does it mean to be original?

In this course, we care very much about your analysis. It must be original. Original here means two things: 1) the text of the written report must be your own and 2) the exploration that leads to your analysis must be your own. Plagiarism typically refers to the former explicitly, but in this case it also refers to the latter explicitly.

It is well known that for this course we do not care about code. We are not interested in your working out the edge cases in k-nn, or proving your skills with python. While there is some value in implementing algorithms yourselves in general, here we are interested in your grokking the practice of ML itself. That practice is about the interaction of algorithms with data. As such, the vast majority of what you're going to learn in order to master the empirical practice of ML flows from doing your own analysis of the data, hyper parameters, and so on; hence, you are allowed to steal ML code from libraries but are not allowed to steal code written explicitly for this course, particularly those parts of code that automate exploration. You will be tempted to just run said code that has already been overfit to the specific datasets used by that code and will therefore learn very little.

How to cite:

If you are referring to information you got from a third-party source or paraphrasing another author, you need to cite them right where you do so and provide a reference at the end of the document [Col]. Furthermore, “if you use an author’s specific word or words, you must place those words within quotation marks and you must credit the source.” [Wis]. It is good style to use quotations sparingly. Obviously, you cannot quote other people’s assignment and assume that is acceptable. Speaking of acceptable, citing is not a get-out-of-jail-free card. You cannot copy text willy nilly, but cite it all and then claim it’s not plagiarism just because you cited it. Too many quotes of more than, say, two sentences will be considered plagiarism and a terminal lack of academic originality.

Your README file will include pointers to any code and libraries you used.

If we catch you...

We report all suspected cases of plagiarism to the Office of Student Integrity. Students who are under investigation are not allowed to drop from the course in question, and the consequences can be severe, ranging from a lowered grade to expulsion from the program.

References

[Col] Williams College. *Citing Your Sources: Citing Basics.* URL: `https://libguides.williams.edu/citing`.

[Wis] University of Wisconsin - Madison. *Quoting and Paraphrasing.* URL: `https://writing.wisc.edu/handbook/assignments/quotingsources`.

Original assignment description written by Charles Isbell. Updated for Spring 2024 by John Mansfiled and Theodore LaGrow. Modified for LaTeX by John Mansfield.