CS7641 Assignment 2
Randomized Optimization

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 random search. As always, it is important to realize that understanding an algorithm or technique requires more than reading about that algorithm or even implementing it. One should actually have experience seeing how it behaves under a variety of circumstances.

As such, you will be asked to implement several randomized search algorithms. In addition, you will be asked to exercise your creativity in coming up with problems that exercise the strengths of each method.

As always, you may program in any language that you wish insofar as you feel the need to program. As always, *it is your responsibility* to make sure that we can actually recreate your narrative, if necessary.

3 Procedure

3.1 The Problems You Give Us

You must implement four local random search algorithms. They are:

1. Randomized hill climbing
2. Simulated annealing
3. A genetic algorithm
4. MIMIC

You will then create three optimization problem domains. For the purpose of this assignment an ”optimization problem” is a fitness function one is trying to maximize (as opposed to a cost function one is trying to minimize). This choice doesn’t make the problem easier or harder, but picking one of the dualities maintains consistent grading. Please note that the problems you create should be over discrete-valued parameter spaces. Bit strings are preferable.

You will apply all four search techniques to these three optimization problems. The first problem should highlight advantages of your genetic algorithm, the second of simulated annealing, and the third of MIMIC. Be creative and thoughtful. It is not required that the problems be overly-complicated or painful long in runtime. They can be simple. For example, the 4-peaks and k-color problems are rather straightforward, but illustrate relative strengths rather neatly.

3.2 The Problems Given to You

In addition to analyzing discrete optimization problems, you will also use the first three algorithms to find good weights for a neural network. In particular, you will use them instead of backprop for the neural network you used in Assignment 1 on at least one of the problems you created for Assignment 1. Notice that this assignment is about an optimization problem and about supervised learning problems. That means that looking at only the loss or only the accuracy will not tell you the whole story. You will need to integrate your knowledge on optimization problem analysis and supervised learning nuances to craft a detailed report.

Below are common pitfalls:

- The weights in a neural network are continuous and real-valued instead of discrete so you might want to think a little bit about what it means to apply these sorts of algorithms in such a domain.
- There are different loss and activation functions for NNs. If you use different libraries across your assignments, you either need to make sure those are the same or retune your model using the new library.

3.3 Experiments and Analysis

Including consideration from your Assignment 1 report for experiments and analysis, your Assignment 2 report should contain:

- The results you obtained running the algorithms on the networks. Why did you get the results you did? What sort of changes might you make to each of those algorithms to improve performance? Supporting graphs and/or tables should be included to help with arguments and strengthen hypotheses.
- A description of your optimization problems, and why you feel that they are interesting and exercise the strengths and weaknesses of each approach. 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.
- Understanding of each algorithm's tuning for selected hyperparameter ranges. Please experiment with more than one hyperparameter and make sure the results and subsequent analysis you provide are meaningful. You are required to state your optimal parameters with rationale but not explicitly required to include graphs.
- 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.

3.4 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 libraries:

- mlrose-hiive (python) `https://pypi.org/project/mlrose-hiive/`
- ABAGAIL (java) `https://github.com/pushkar/ABAGAIL`
- pytorch (python) `https://pytorch.org/`

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.

7 Version Control

- 01/16/2024 - John Mansfiled updated resources and pitfalls for improved details.
- 02/15/2024 - Theodore LaGrow updated Section 3.3 Bullet 3 to clarify language on what is needed for hyperparameter exploration and demonstration.

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.