DSA4212 Year 2022-2023
Assignment No: 3

Deadline: 23:59, 23rd of April 2023

1 Projects:

Select **one** of the following projects. These projects are designed to encourage you to conduct independent research and investigations and expand your understanding of optimization and machine learning in general.

1. **Thompson problem:** consider $N = 300$ points $P_1, P_2, \ldots, P_N \in \mathbb{R}^3$ on the unit sphere: for $1 \leq i \leq N$, we have $\|P_i\| = 1$. How should these points be placed so that the quantity

$$\mathcal{E} = \sum_{i<j} \frac{1}{\|P_i - P_j\|}$$

   is minimized. Note that it is the norm, and not the norm squared, in the denominator.

2. **Travelling Salesman**: the file `cities.npy` contains the 2D coordinates $P_1, \ldots, P_{1000}$ of $N = 1000$ cities. Find a permutation $\sigma(1), \ldots, \sigma(N)$ of these N cities that minimises the total length L defined as

$$L = \|P_{\sigma(2)} - P_{\sigma(1)}\| + \|P_{\sigma(3)} - P_{\sigma(2)}\| + \ldots + \|P_{\sigma(N)} - P_{\sigma(N-1)}\| + \|P_{\sigma(1)} - P_{\sigma(N)}\|.$$

   Note that L is the length of a *tour* that visits each city once and comes back to the initial city.

3. **Natural Evolution Strategies:** The class of algorithms known as *Natural Evolution Strategies* (NES) has demonstrated significant potential in various fields of applied mathematics and machine learning. In this project, you will provide an overview of NES and apply these approaches to three optimization problems of your choice.

   Possible readings:

   (a) Wierstra, Daan, et al. ”Natural evolution strategies.” The Journal of Machine Learning Research 15.1 (2014): 949-980.

(b) Ollivier, Yann, et al. ”Information-geometric optimization algorithms: A unifying picture via invariance principles.” The Journal of Machine Learning Research 18.1 (2017): 564-628.

(c) Salimans, Tim, et al. ”Evolution strategies as a scalable alternative to reinforcement learning.” arXiv preprint arXiv:1703.03864 (2017).

4. **Regression with uncertainty**: The dataset `delays.csv` contains information related to $N = 717,003$ flights. The objective of this project is to predict the arrival delay (i.e., the first column named **ArrDelay**) with an associated uncertainty. You will use the first 80% of the dataset as the training set and the remaining 20% as the test set. For each flight, your model should predict two values: **(1)** an estimate μ_i of the arrival delay, and **(2)** a standard deviation estimate σ_i. The metric you aim to minimize is the average predictive negative log-likelihood, given by:

$$\frac{1}{N_{\text{Test}}} \sum_{i \in \text{Test}} \frac{(\text{ArrDelay}_i - \mu_i)^2}{\sigma_i^2} + \log(\sigma_i). \tag{1}$$

Indeed, you must not use the test dataset in any manner for building your model.

Remarks:

(a) It may be worth trying to create new features, and possibly clean some of the features

(b) If you need to compute *nearest neighbors*, the FAISS library may be useful.

(c) It may be interesting to read about Gaussian Process regression models.

(d) Start with simple models!

5. **Spin Glasses Ground State**: the file `spin_glasses.npy` contains a matrix $J \in \mathbb{R}^{N,N}$ with $N = 100$. The goal of this project is to find a configuration $(s_1, s_2, \ldots, s_N) \in \{-1, 1\}^N$, i.e. a vector of length $N = 100$ whose coordinates are either equal to -1 or $+1$, such that the quantity

$$H = \sum_{1 \le i,j \le N} J_{i,j}\, s_i\, s_j \tag{2}$$

is minimized. This discrete optimization is fundamental in many respects as many situations in computer science can be mapped to it.

6. **Word Embedding:** train from scratch and on a dataset of your choice a *Word Embedding* model. Evaluate its usefulness on a task of your choice.

7. **Topic of your choice:** propose to me a topic of your choice by the 31-st of March. I will let you know within a day if it is acceptable.

2 CANVAS Submission

There are (at least) 2 files to submit:

1. A pdf report. This report should not include any Python code. Instead, it should give an overview of your project. It should be at the very most 8 pages, but can also be significantly shorter (ie. do **not** write a long report, just for the sake of writing a long report).

2. A Jupyter notebook describing some (not necessarily all) of the experiments that you have performed. This can be split into several notebooks if necessary. This Jupyter notebook should be reproducible: anyone should be able to run it from scratch.

For submitting your work, you will:

1. Zip all your files into a **single zip-file**

2. Use the naming convention `GROUPXX.zip` where XX is your 2-digit group number (i.e. 01, 02, etc...).

3. Make sure that the pdf-report includes the name and student number of all the students in the group.

4. Upload the file on CANVAS.

Do not include anything else in the zip-file except the pdf report and the jupyter notebooks.

2.1 Grading

The evaluation will consider the following components:

1. **[40%]** Clarity and reproducibility of the Python code and PDF report.

2. **[40%]** Quality and appropriateness of the numerical experiments and literature review.

3. **[20%]** Proper citation and acknowledgment of resources used (e.g., books, GitHub code, articles, blog posts, Kaggle code). For instance, if your approach heavily relies on code or a blog post found online (which is perfectly acceptable), please do mention it. *Failure to properly acknowledge the sources and materials used will be considered plagiarism.*