MiniProject 1: Getting Started with Machine Learning

COMP 551, Fall 2023, McGill University
Contact TAs: Nima Fathi and Vraj Patel

Please read this entire document before starting the assignment.

Preamble

- This mini-project is **due on October 5th at 11:59 pm (EST, Montreal Time).** There is a penalty of 2^k percent penalty for k days of delay, which means your grade will be scaled to be out of $100 - 2^k$. No submission will be accepted after 6 days of delay.
- This mini-project is to be completed in groups of three. All members of a group will receive the same grade except when a group member is not responding or contributing to the project. If this is the case and there are major conflicts, please reach out to the group TA for help and flag this in the submitted report. Please note that it is not expected that all team members will contribute equally. However, every team member should make integral contributions to the project, be aware of the content of the submission and learn the full solution submitted.
- You will submit your assignment on MyCourses as a group. You must register your group on MyCourses and any group member can submit. See MyCourses for details.
- We recommend to use **Overleaf** for writing your report and **Google colab** for coding and running the experiments. The latter also gives access to the required computational resources. Both platforms enable remote collaborations.
- You should use Python for this and the following mini-projects. You are free to use libraries with general utilities, such as matplotlib, numpy and scipy for Python, unless stated otherwise in the description of the task. In particular, in most cases you should implement the models and evaluation functions yourself, which means you should not use pre-existing implementations of the algorithms or functions as found in SciKit learn, and other packages. The description will specify this in a per case basis.

Background

In this miniproject you will implement two ML models—Linear regression, Logistic regression (with gradient descent)—and provide analysis on these two models on two distinct datasets. The goal is to get started with programming for Machine Learning and learn how these two commonly used models work.

Task 1: Acquire, preprocess, and analyze the data

Your first task is to acquire the data, analyze it, and clean it (if necessary). We will use two fixed datasets in this project, as outlined below.

- **Dataset 1: Boston Housing dataset**

https://www.kaggle.com/datasets/fedesoriano/the-boston-houseprice-data?select=boston.csv
or
http://lib.stat.cmu.edu/datasets/boston
This dataset consists of 506 data samples and 14 real attributes. **Use 13 attributes instead, one of the features called "B" should be removed from the dataset since it has some ethical problems.** The target value is the Median value of owner-occupied homes in $1000's ('MEDV' feature). You can directly download it from the Kaggle link given before or the other link from CMU university:

- **Dataset 2: Wine dataset:**
  https://archive.ics.uci.edu/dataset/109/wine
  This dataset consist of 178 data samples and 13 attributes. The dataset is distributed in three different classes.

The essential subtasks for this part of the project are:

1. Load the datasets into NumPy or Pandas objects in Python.
2. Clean the data. Are there any missing or malformed features? Are there any other data oddities that need to be dealt with? **You should remove any examples with missing or malformed features and note this in your report.**

   If you choose to play with Pandas dataframes, a handy line of code that might be helpful is df[*df.eq('?').any(1)*], where df is the dataframe, and '?' represents a missing value in the datasets. This is a straightforward way to handle this issue by simply eliminating rows with missing values. You are welcome to explore other possible ways.
3. Compute basic statistics on the data to understand it better. E.g., what are the distributions of the different classes, what are the distributions of some of the numerical features?

Task 2: Implement the models

You are free to implement these models as you see fit, but you should follow the equations that are presented in the lecture slides, and you must implement the models from scratch (i.e., you **CANNOT** use SciKit Learn or any other pre-existing implementations of these methods). However, you are free to use relevant code given in the course website.

In particular, your three main tasks in the part are to:

1. Implement analytical linear regression solution for Dataset 1.
2. Implement logistic regression with gradient descent for Dataset 2.
3. Implement mini-batch stochastic gradient descent for both linear and logistic regression.

You are free to implement these models in any way you want, but you must use Python and you must implement the models from scratch (i.e., you cannot use SciKit Learn or similar libraries). Using the NumPy or Pandas package, however, is allowed and encouraged. Regarding the implementation, we recommend the following approach (but again, you are free to do what you want):

- Implement both models as Python classes. You should use the constructor for the class to initialize the model parameters as attributes, as well as to define other important properties of the model.
- Each of your models classes should have (at least) two functions:
  - Define a `fit` function, which takes the training data (i.e., $\mathbf{X}$ and $\mathbf{Y}$)—as well as other hyperparameters (e.g., learning rate and batch size)—as input. This function should train your model by modifying the model parameters.

- Define a `predict` function, which takes a set of input points (i.e., $\mathbf{X}$) as input and outputs predictions (i.e., $\hat{\mathbf{y}}$) for these points.

Task 3: Run experiments

The goal of this project is to have you be familiar with how to train models.

Split each dataset into training, and test sets. Use test set to estimate performance in all of the experiments after training the model with training set. Evaluate the performance using the corresponding cost function for the classification and regression tasks. You are welcome to perform any experiments and analyses you see fit, **but at a minimum you must complete the following experiments in the order stated below***:*

1. ***For both datasets****, perform an 80/20 train/test split and report the performance metrics on both the training set and test set for each model. Please include metrics such as Mean Squared Error (MSE) for Linear Regression and accuracy, precision, recall, and F1-score for Logistic Regression.*

2. ***For both datasets****, use a 5-fold cross-validation technique and report the performance metrics on both the training set and test set for each model. Again, include appropriate performance metrics for each model. Check this link for more information.*

   Note: 5-fold cross-validation is a technique where the dataset is divided into five equal parts (folds), and a model is trained and evaluated five times, each time using a different fold as the validation set and the remaining four folds for training.

3. ***For both datasets****, Sample growing subsets of the training data (20%,30%,...80%). Observe and explain how does size of training data affects the performance for both models. Plot two curves as a function of training size, one for performance in train and one for test.*

4. ***For both datasets****, try out growing minibatch sizes, e.g., 8, 16, 32, 64, and 128. Compare the convergence speed and final performance of different batch sizes to the fully batched baseline. Which configuration works the best among the ones you tried?*

   Note: This is for SGD only (Task2, third main task).

5. ***For both datasets****, Present the performance of both linear and logistic regression with at least three different learning rates (your own choice).*

6. ***For both datasets****, Given a variety of parameter configurations, select a performance metric and present the optimal parameter choice for each dataset. Please provide a rationale for your metric selection, along with an explanation of why you opted for that particular metric.*

7. ***Only for dataset1****, Gaussian Basis Functions:*

   - *Utilize Gaussian basis functions to enrich the feature set for Dataset 1.*
   - *Define each Gaussian basis function as follows:*

   $$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$$

   - *Employ a total of 5 Gaussian basis functions.*
   - *Set the spatial scale parameter, s, to a value of 1.*
   - *Select μ_j values randomly from the training set to determine the centers of these basis functions.*
   - *Use analytical linear regression to predict the target value.*

Compare the target and predicted values obtained with the new dataset with the results obtained with the original feature set, i.e. compare with the results obtained without Gaussian basis functions.

8. ***Only for dataset1**, Compare analytical linear regression solution with mini-batch stochastic gradient descent-based linear regression solution. What do you find? Why do you think mini-batch stochastic gradient descent is used when an analytical solution is available?*

Note: The above experiments are the minimum requirements that you must complete; however, this project is open-ended. *For example, what happens when you add momentum to the gradient descent implementation? What happens if you add regularization? How about different evaluation metrics for classification and regression problem? Which kind of feature preprocessing improves performance? You do not need to do all of these things, but you should demonstrate creativity, rigour, and an understanding of the course material in how you run your chosen experiments and how you report on them in your write-up.*

Deliverables

*You must submit two separate files to MyCourses (**using the exact filenames and file types outlined below**):*

1. ***code.zip**: Your data processing, classification and evaluation code (as some combination of .py and .ipynb files).*
2. ***writeup.pdf**: Your (max 5-page) project write-up as a pdf (details below).*

Project write-up

*Your team must submit a project write-up that is a maximum of five pages (single-spaced, 11pt font or larger; minimum 0.5 inch margins, an extra page for references/bibliographical content can be used). We highly recommend that students use LaTeX to complete their write-ups. **This first mini-project report has relatively strict requirements, but as the course progresses your project write-ups will become more and more open-ended**. You have some flexibility in how you report your results, but you must adhere to the following structure and minimum requirements:*

Abstract (100-250 words) *Summarize the project task and your most important findings. For example, include sentences like "In this project we investigated the performance of two machine learning models on two benchmark datasets".*

Introduction (5+ sentences) *Summarize the project task, the two datasets, and your most important findings. This should be similar to the abstract but more detailed. You should include background information and citations to relevant work (e.g., other papers analyzing these datasets).*

Datasets (5+ sentences) *Very briefly describe the datasets and how you processed them. Present the exploratory analysis you have done to understand the data, e.g. class distribution. Highlight any possible ethical concerns that might arise when working these kinds of datasets.*

Results (7+ sentences, possibly with figures or tables) *Describe the results of all the experiments mentioned in **Task 3** (at a minimum) as well as any other interesting results you find (Note: demonstrating figures or tables would be an ideal way to report these results).*

Discussion and Conclusion (5+ sentences) *Summarize the key takeaways from the project and possibly directions for future investigation.*

Statement of Contributions (1-3 sentences) *State the breakdown of the workload across the team members.*

References and appendix *Please consider enhancing the quality of this section by incorporating additional details, such as graphs and citing important papers as the sources of this information. It's worth noting that this section is not subject to the 5-page limit. Therefore, you have the flexibility to conduct more experiments and include relevant plots within this section, which can be cross-referenced in the results and discussion for a more comprehensive presentation.*

Evaluation

The mini-project is out of 100 points, and the evaluation breakdown is as follows:

- *Completeness (20 points)*
  - *Did you submit all the materials?*
  - *Did you run all the required experiments?*
  - *Did you follow the guidelines for the project write-up?*
- *Correctness (40 points)*
  - *Are your models implemented correctly?*
  - *Are your reported performance close to our solution?*
  - *Do you observe the correct trends in the experiments (e.g., how performance changes as the minibatch or learning rates changes?)?*
  - *Do you find notable features of the decision boundaries?*
- *Writing quality (25 points)*
  - *Is your report clear and free of grammatical errors and typos?*
  - *Did you go beyond the bare minimum requirements for the write-up (e.g., by including a discussion of related work in the introduction)?*
  - *Do you effectively present numerical results (e.g., via tables or figures)?*
- *Originality / creativity (15 points)*
  - *Did you go beyond the bare minimum requirements for the experiments?*
  - ***Note:** Simply adding in a random new experiment will not guarantee a high grade on this section! You should be **thoughtful and organized** in your report. That is, the distinctive ideas that you came up with should blend in your whole story. For instance, explaining the triggers behind them would be a great starting point.*

Final remarks

You are expected to display initiative, creativity, scientific rigour, critical thinking, and good communication skills. You don't need to restrict yourself to the requirements listed above - feel free to go beyond, and explore further.

*You can discuss methods and technical issues with members of other teams, but **you cannot share any code or data with other teams.***

Congratulations on completing your first course project! You are now familiar with the two most commonly used ML models. As the class goes on, we will see more machine learning models and their interesting applications in real life.