Homework 3

AMATH 482, Winter 2025

Assigned Feb 17th, 2025. Checkpoint due Feb 25th, 2025. Report and Code due on March 2nd, 2025 at midnight.

Directions, Reminders and Policies

Read these instructions carefully: There are two stages for submitting the assignment.

- **You will be submitting a *Checkpoint* approximately one week after the assignment was published (see due date above). The checkpoint includes submission of your code in progress (at least 1/3) (2 points) and taking the checkpoint quiz (3 possible points).**
- The report should be a maximum of 6 pages long with references included. Minimum font size 10pts and margins of at least 1inch on A4 or standard letter size paper.
- Do not include your code in the report. Simply create a zip file of your main scripts and functions, without figures or data sets included, and upload the zip file to Canvas.
- Your report should be formatted as follows:
  - Title/author/abstract: Title, author/address lines, and short (100 words or less) abstract. This is not meant to be a separate title page.
  - Sec. 1. Introduction and Overview
  - Sec. 2. Theoretical Background
  - Sec. 3. Algorithm Implementation and Development
  - Sec. 4. Computational Results
  - Sec. 5. Summary and Conclusions
  - Acknowledgments (no more than four or five lines, also see the point below on collaborations)
  - References
- LaTeX(Overleaf is a great option) is recommended to prepare your reports. A template is provided on Canvas in Homework/Files. You are also welcome to use Microsoft Word or any other software that correctly typesets mathematical equations and properly allows you to include figures.
- Collaborations are encouraged; however, everything that is handed in (both your report and your code) should be your work. You are welcome to discuss your assignments with your peers and seek their advice but these should be clearly stated in the acknowledgments section of your reports. This also includes any significant help or suggestions from the TAs or any other faculty in the university. You don't need to give all the details of the help you received, just a sentence or two. A similar guideline applies to the use of Large Language Models (LLM). These are permitted for the study of topics and code presented in class and a better grasp of the problem and its solution. However, everything that is handed in (both your report and your code) should be your work and cannot be based on LLM content (modified or direct). Any use of external help should be specified in the acknowledgments section of the report.
- **Late reports are subject to a 2 points/day penalty up to five days. They will be no longer accepted afterwards. For example, if your report is three days late and you managed to get $16/20$, your final grade will be $16 - 6 = 10$, so be careful with late submission.**

Problem Description: MNIST Digit Classification

Your goal in this assignment is to train classifiers to distinguish images of handwritten digits from the famous MNIST data set. This is a classic problem in machine learning and often times one of the first benchmarks one tries new algorithms on. The data set is split into training and test sets. You will train your classifiers using the training set while the test set is only used for validation/evaluation of your classifiers. The data (both train and test sets) is from Yann Lecun http://yann.lecun.com/ and placed in google drive for you to download. Alternatively you can also use sklearn.datasets to load the dataset. For Python, parse it into matrices using the HW3Helper notebook provided. For Matlab there are codes available online that will help you to do this (e.g. https://github.com/sunsided/mnist-matlab).

Figure 1 First 64 Digits in MNIST Dataset

Some comments and hints

Here are some useful comments and facts to guide you along the way.

1. In this assignment, it is advised to use `sklearn` for many of the analyses that you would like to perform since MNIST dataset is larger than the toy datasets that we previously considered. In particular, you will find that direct application of SVD on the training set would be time-consuming while PCA implementation in *sklearn.decomposition* is more optimized. Read the instructions for each function you will be using in https://scikit-learn.org/stable/user_guide.html and check that your dimensions and formatting are compatible with *sklearn* convention.
2. You are welcome to use the code samples from class for various classifiers that we applied to the IRIS dataset. (These will be presented in class in the coming week)

Tasks

Below is a list of tasks to complete in this assignment and discuss in your report.

1. You will need to reshape each image into a vector and stack the vectors into matrices X_{train} and X_{test} respectively. Perform PCA analysis of the digit images in the train set. Plot the first 16 PC modes as 28×28 images (see an example on the previous page of how multiple images can be displayed in a grid).
2. Inspect the cumulative energy of the singular values and determine k: the number of PC modes needed to approximate 85% of the energy. You may also want to inspect several approximated digit images reconstructed from k truncated PC modes and plot them to make sure that the image reconstruction using truncated modes is reasonable.
3. Write a function that selects a subset of particular digits (all samples of them) from X_{train}, y_{train}, X_{text} and y_{test} and returns the subset as new matrices X_{subtrain}, y_{subtrain}, X_{subtext} and y_{subtest}.
4. Select the digits 1,8 using step 3, project the data onto k-PC modes computed in steps 1-2, and apply the Ridge classifier (linear) to distinguish between these two digits. Perform cross-validation and testing and discuss your results.
5. Repeat the same classification procedure for pairs of digits 3,8 and 2,7. Report your results and compare them with the results in step 4. If there is any difference can you explain it?
6. Use all the digits and perform multi-class classification with Ridge and KNN classifiers. Report your results and discuss how they compare between the methods. Which method performs the best?
7. **Bonus (+2 points)**: Implement an alternative classifier, that we did not cover in class, (e.g. SVM), and compare its results with the classifiers in the previous step.