Homework 2

AMATH 582/482, Winter 2022

Assigned Jan 28, 2022. Due on Feb 11, 2020 at midnight.

Directions, Reminders and Policies

Read these instructions carefully:

- **You are required to upload a PDF report to Canvas along with a zip of your code. Note the PDF and the zip should be uploaded separately.**
- 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
- I suggest you use LaTeX(Overleaf is a great option) to prepare your reports. A template is provided on Canvas under the Syllabus tab. You are also welcome to use Microsoft Word or any other software that properly typesets mathematical equations.
- I encourage collaborations, 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.
- Your homework will be graded based on how completely you solved it as well as neatness and little things like: did you label your graphs and include figure captions. **The homework is worth 20 points. 10 points will be given for the overall layout, correctness and neatness of the report, and 10 additional points will be for specific technical things that the TAs will look for in the report itself.**
- **Late submissions will not be accepted on Canvas, send them to bamdadh@uw.edu directly. Late reports are subject to a 2 points/day penalty up to five days. They are 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** $16 - 6 = 10$**.** Basically, you will lose 2% of your overall course grade for each day the report is late. So be careful.

Problem Description: Classifying Digits

Your goal in this home work is to train a classifier 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. You may download these data sets from Google drive:

- `MNIST_training_set.py` and `MNIST_test_set.py` are python binaries for those of you who are using Python.
- `MNIST_training_set.mat` and `MNIST_test_set.mat` are MATLAB .mat files for MATLAB users.
- The `.csv` files are text files containing the entire data set, splitting training and test features and labels, for those using other languages and other formats.

The training set contains 2000 instances of handwritten digits, the "features" are 16×16 black and white images while the "labels" are the corresponding digit (note the images are shaped as vectors of size 256 and need to be reshaped for visualization). The test set has the same attributes except that there are only 500 instances. The provided Helper notebook imports the data and visualizes some of the features and labels.

Figure 1 The first 64 features from the training set. The helper notebook prints the corresponding labels.

Some comments and hints

Here are some pointers.

1. `sklearn` has functionalities for PCA and Ridge regression with crossvalidation. Using these can make your life much easier.
2. Don't forget to center X_{train} before computing the PCA modes if you plan to use SVD. If you are using `sklearn`'s PCA function then you don't need to worry about this as it centers the data by default.

3. Make note that in task 3 you will be projecting your sub-training and sub-test sets corresponding to specific digits on the PCA modes that were computed on the entire training set.
4. In step 3 we assign labels -1 and +1 to images of the digits 1 and 8. This is done to make the output of your classifier normalized and is common practice in binary classification where we wish to distinguish only two classes in a data set.

Tasks

Below is a list of tasks to complete in this homework and discuss in your report. We will use $X_{\text{train}} \in \mathbb{R}^{2000\times 256}$ to denote the matrix of training features and $Y_{\text{train}} \in \mathbb{R}^{2000}$ to denote the vector of training labels. Similarly $X_{\text{test}} \in \mathbb{R}^{500\times 256}$ and $Y_{\text{test}} \in \mathbb{R}^{500}$ are the test features and labels.

1. Use PCA to investigate the dimensionality of X_{train} and plot the first 16 PCA modes as 16×16 images.
2. How many PCA modes do you need to keep in order to approximate X_{train} up to $60\%, 80\%$ and 90% in the Frobenius norm? Recall the identity

$$\|B\|_F^2 = \sum_{j=1}^{\min\{m,n\}} \sigma_j(B)^2, \qquad B \in \mathbb{R}^{m\times n}.$$

   Do you need the entire 16×16 image for each data point?
3. Train a classifier to distinguish the digits 1 and 8 via the following steps:

   - First, you need to write a function that extracts the features and labels of the digits 1 and 8 from the training data set. Let us call these $X_{(1,8)}$ and $Y_{(1,8)}$.
   - Then project $X_{(1,8)}$ on the first 16 PCA modes of X_{train} computed in step 1, this should give you a matrix A_{train} which has 16 columns corresponding to the PCA coefficients of each feature and 455 rows corresponding to the total number of 1's and 8's in the training set. .
   - Assign label -1 to the images of the digit 1 and label $+1$ to the images of the digit 8. This should result in a vector $b_{\text{train}} \in \{-1, +1\}^{455}$.
   - Use Ridge regression or least squares to train a predictor for the vector b_{train} by linearly combining the columns of A_{train}.
   - Report the training mean squared error (MSE) of your classifier

$$\text{MSE}_{\text{train}}(1,8) = \frac{1}{\text{length of } b_{\text{train}}} \times \|A_{\text{train}}\hat{\beta} - b_{\text{train}}\|_2^2.$$

   - Report the testing MSE of your classifier

$$\text{MSE}_{\text{test}}(1,8) = \frac{1}{\text{length of } b_{\text{test}}} \times \|A_{\text{test}}\hat{\beta} - b_{\text{test}}\|_2^2,$$

     where you need to construct analogously the matrix A_{test} and b_{test} corresponding to the digits 1 and 8 from the test data set.
4. Use your code from step 3 to train classifiers for the pairs of digits $(3, 8)$ and $(2, 7)$ and report the training and test MSE's. Can you explain the performance variations?