Project 2: Once more with k-means

MTH448

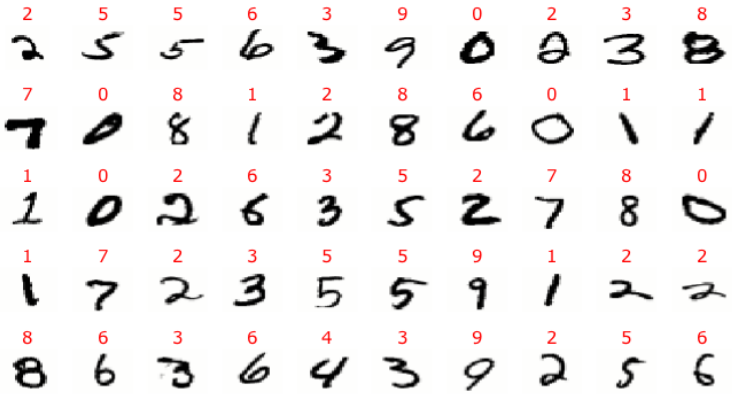
Due: 3/4/2025

1 Introduction

In this project you will apply the K-Means Clustering algorithm to the MNIST database.

1.1 MNIST Database

The MNIST database i contains 60,000 images of hand-written digits (converted into 28×28 pixel images) and labels of what the digit was supposed to be (see the notes from Lecture 4 for a discussion of how the database is formatted). Some sample entries in the database are presented below:



2 Project Objectives

2.1 Part 1:

- Use the K-Means algorithm to split the MNIST images into 10 clusters.
- Display the centroids. Do they resemble digits?
- Check how useful the clusters are for classifying images of digits. That is, check how accurately we can predict which image corresponds to which digit based on which cluster the image belongs to.

2.2 Part 2:

Note: For full credit, in your report you can choose to investigate one of the following approaches (2a or 2b). You can also experiment with both of them, but this is not required.

An issue with the k-NN algorithm is that it can be slow if we have a big set of training data, and if this data is highly dimensional. The goal of this part of the project is to investigate two different ways of dealing with this issues using k-means clustering. One way is to reduce the amount of training data, and the second is to reduce dimensionality of data.

2.2.1 Part 2a:

The goal of this part is to experiment with a possible way of making k-NN predictions faster, by reducing the amount of the training data.

1. Split the MNIST images into training and test data.
2. For each digit 0-9 select all training images corresponding to this digit, and then use k-means to split these images into several (e.g. 100) clusters. Use centroids of these clusters as new training data for k-NN. This will make the amount of training data smaller, since every cluster of the original training data will be replaced by a single centroid.
3. Investigate how the prediction accuracy and speed of k-NN using this new training data compares to the case when you use training data without clustering. Also, if you train k-NN using the centroid training data and then train it using the same amount of randomly selected training data is there a difference in the prediction accuracy?

2.2.2 Part 2b:

The goal of this part is to experiment with a possible way of making k-NN predictions faster, by reducing dimensionality of the MNIST data.

1. Split MNIST images into training and test data.

2. Use K-Menas clustering to reduce the dimensionality of the training data.

    - Say your training data D is $N \times 784$. If you cluster D^T into k clusters, your data is now essentially $N \times k$.

3. Use your new clustering to reduce the dimensionality of your training and testing data.

4. Investigate how the prediction accuracy and speed of KNN using the reduced data compares to the predictions done without dimensionality reduction.

5. Experiment with dimensionality reduction done with different numbers of clusters and compare the results.

Note: Use the KNN and KMeans implemented in sklears, they will run faster then the version we implemented from scratch.

3 Grading

The report grade will have the following distribution:

Element	**Weight**	**What will be graded**
Introduction Section	10%	Quality of narrative.
Conclusion Section	10%	Quality of narrative.
Report Content	30%	Work done on developing the project. Your analysis, insights, observations, and interpretations. Quality of the narrative of the report.
Python Code	30%	Quality of the Python code included in the report. Relevance of the code to the project. Code organization and readability. Documentation of code by code comments.
Presentation	20%	Organization of the report. Text formatting. Use of LaTeX to typeset mathematics. Formatting of code output. Quality of graphs and plots.

3.1 Introduction

The introduction is the first section of your report. It describes the project (the underlying math) and the goals of the report. It should be written in a way that is engaging and understandable to a student who has some background in math and coding but does not take this course. The introduction SHOULD NOT be in list form. While some of the concepts in the introduction will be repeated from the project description, you should state the concepts in your own words and not copy from the project description.

3.2 Conclusion

This section should summarize your results and major findings. It can also include potential future extensions of the project. This should be the last section of the report.

3.3 Content and presentation

Outside of the two section mentioned your report should be broken up into sections and subsections in such a way as to maximize readability and comprehensibility. Where appropriate use Latex to format math.

Please note, projects ARE NOT just sets of coding exercises. They are reports in which you explore a particular math problem or phenomenon. Projects should be readable and interesting for a person not taking this class (and thus not having access to the project description) but familiar with math and python.

3.4 Code

Your code should be readable and understandable. Code should be broken up into small snippets each of which gets its own cell. Words should be used to describe what the code is doing and what the logic of your approach is. DO NOT put all of your code into a single section with no words or discussion.

Code should be readable with understandable variable names and comments included to explain what is not clear.

All python code included in the report should be written in such a way that it can be executed sequentially. For example, a function should never be used prior to its definition.

All code included must work. Do not submit code with errors.

Code output must serve the narrative of your report and should be formatted for easy reading and understanding.

3.5 Use of external resources

You are allowed but not at all required to consult resources outside of what is presented in the course. If you use an external resource in a significant way (not just googling an error message or a command that you forgot) please include a citation in your report. You are welcome to use features of python that were not shown in this class but you must understand fully what the feature does. The instructor reserves the right to ask you to explain any fragment of your code. An inability to do so may result in significant grade reduction or even more extreme consequences.

3.6 Collaboration

While students are allowed (and even encouraged) to collaborate and discuss projects the final submission must be the work of solely the submitting individual. Any assignment that is suspected of not being the student's own work

will receive a zero and further disciplinary actions may also follow if they are deemed necessary. Any use of generative AI (e.g., ChatGPT) is prohibited in this class and will be considered a violation of UB's academic integrity policy.