Final Project

Song Liu (song.liu@bristol.ac.uk)
GA 18, Fry Building,
Microsoft Teams (search "song liu").

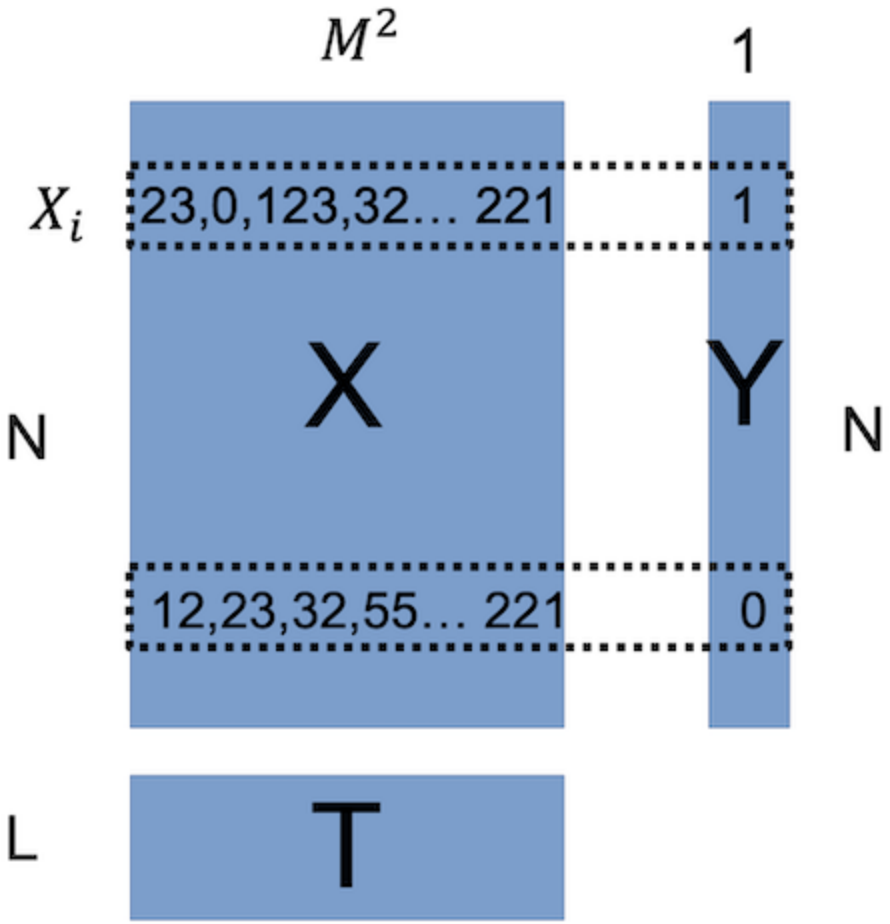
Final Project: "Recognising" Images

- In this final project, you will write a program that "recognises" handwritten digits.
- Given **test images**, your program guesses whether these images are digit "1" or not.
    - The "guessing" is done using the k-nearest neighbours algorithm, a widely known machine learning algorithm.
- This project worth 25% of your total score in this unit.
    - You will get a score from 0-100.

Part I, Loading Images from File (25%)

- A lab folder will be provided, containing the skeleton code, data files.
- The folder contains 3 `.matrix` files.
  - `X.matrix` contains a N by M^2 matrix X where each row is a flattened grayscale M by M image.
  - `Y.matrix` contains a N by 1 matrix Y where each row is a scalar, indicating whether the corresponding row in X is digit 1 or not.
  - `T.matrix` contains a L by M^2 matrix T where each row is a flattened M by M **test image**.
  - X and Y together are called "training set" in machine learning, while T is the "testing set". Y is called the "labels" of X.

Part I, Data Structure



- If $Y_i = 1$, then the image X_i is a handwritten digit 1. If $Y_i \neq 1$, the image X_i is NOT a handwritten digit 1.

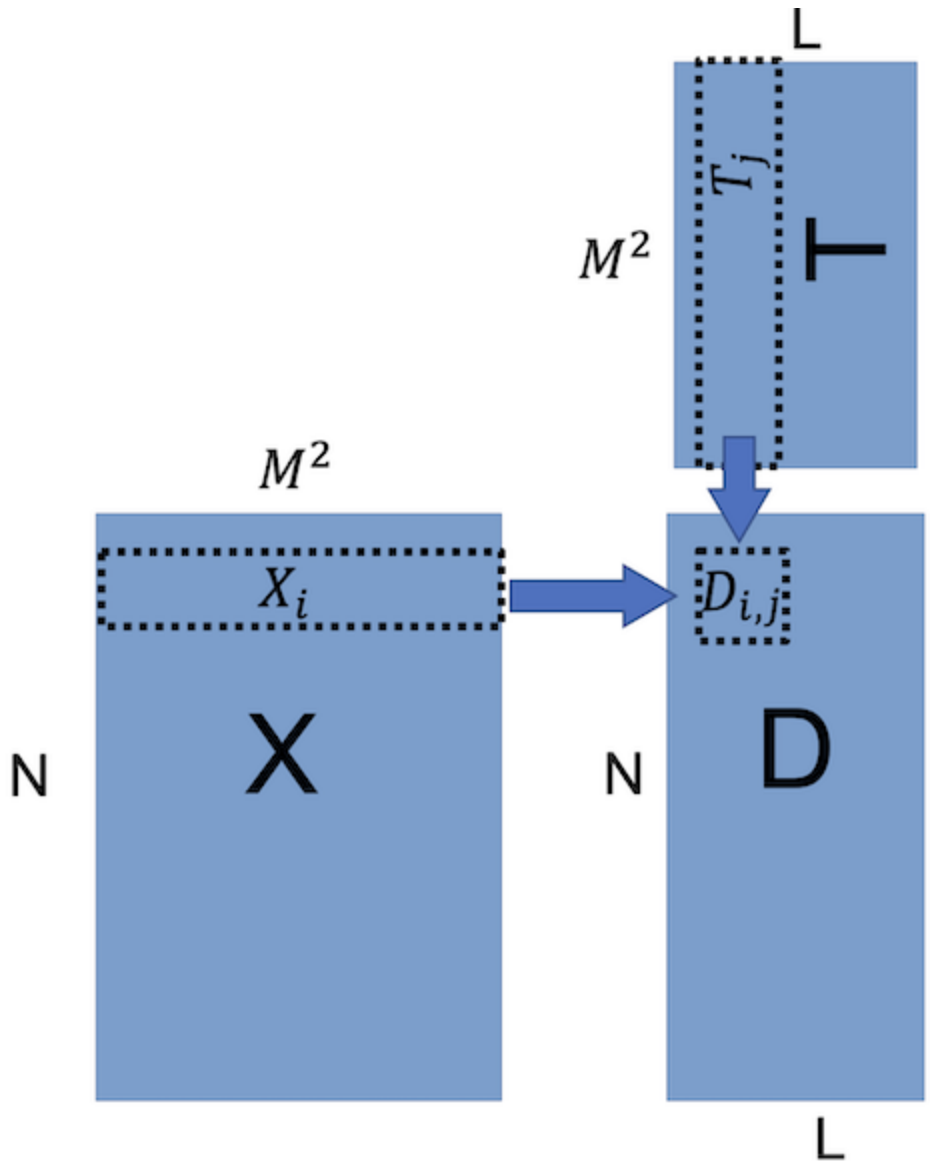
Part I, Loading Images from File

- The format of these matrix files are exactly the same as the ones we encountered at Lab 7.
- After loading the file, print out some basic statistics of X, Y and T.
    - What are M, N and L?
    - How many images in the training set X are digit 1?
- Relevant lectures:
    - Tutorial in week 5.
    - Lecture and Lab in week 7.

Part II Computing Distance Matrix D (25%)

- Construct an N by L matrix D, where the i, j-th element $D_{ij} = \mathrm{dist2}(X_i, T_j)$
  - X_i is the i-th row of X
  - T_j is the j-th row of T.
- $\mathrm{dist2}(a, b)$ computes the squared euclidean distance between two vectors a and b with K elements.
  - $\mathrm{dist2}(a, b) = \sum_{k=1}^{K} (a_k - b_k)^2$.

Part II (Computing D)



Part II Computing D

- **Hint**: Compare the computation of D and the matrix multiplication we have done before. What are the similarities and what are the dissimilarities?
    - Can you modify the matrix multiplication code to compute matrix D?
- **Hint**, you can write a function

```
void pairwise_dist(Matrix X, Matrix T, Matrix D)
```

    - where D is the output, storing the outcome.
- Partial points will be given for correctly written code for computing $\mathrm{dist}(a, b)$.

Part III Guessing Labels (20%)

- For each column of matrix D, find **the indices** of the five smallest elements.
    - Suppose the j-th column in D is a column vector $[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]^\top$,
    - The indices of the five smallest elements are $[0, 1, 2, 3, 4]$.
- Hint: Write a function

```
void minimum5(int len, int a[len], int indices[5]),
```

  It takes an array `a` with length `len` as input, then fills `indices[5]` with the indices of the five smallest elements.

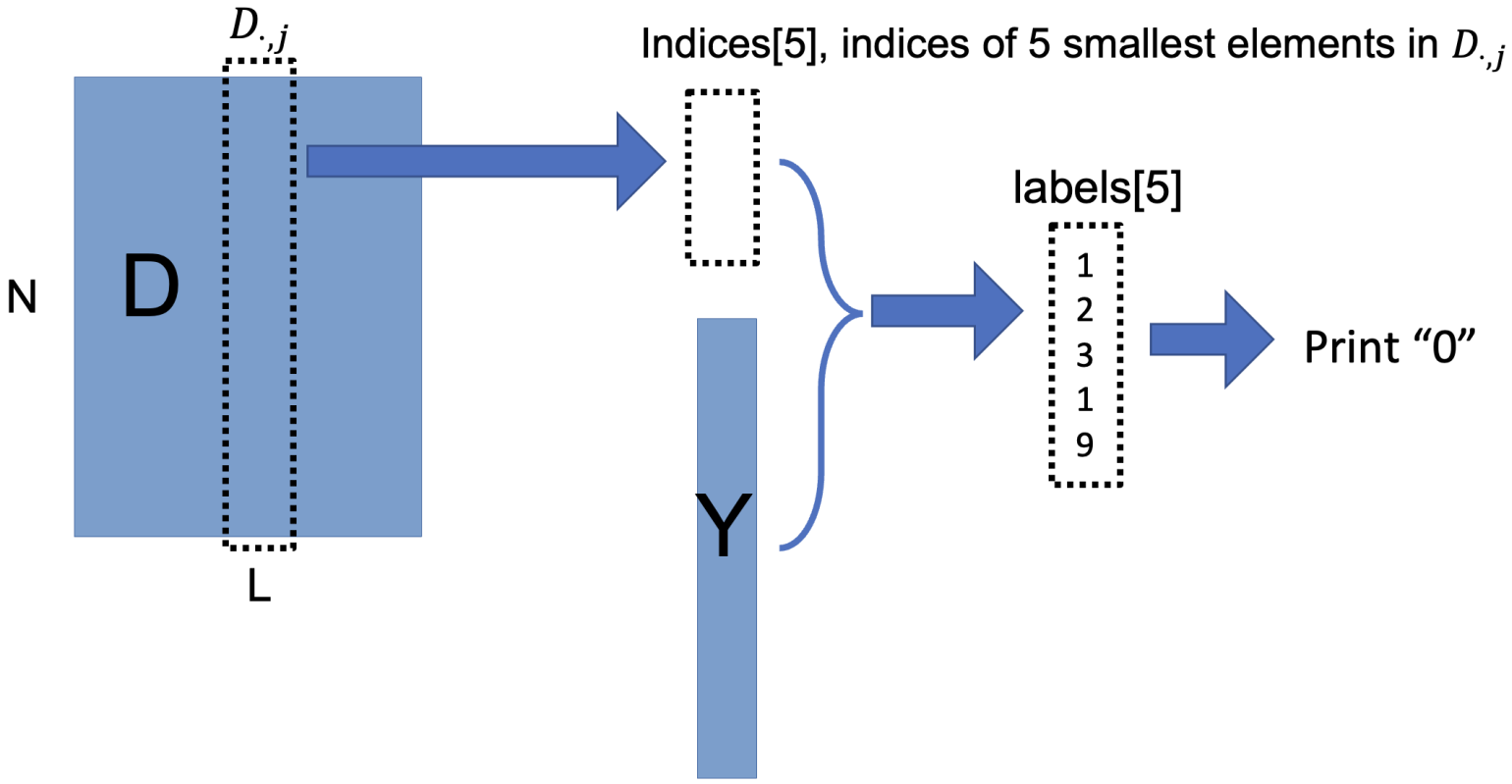
    - Review Lab in week 3.

Part III Guessing Labels

- Now, suppose `indices` contains the indices of the five smallest elements in column j of matrix D.
    - Create a new array `labels` with length 5.
    - Assign the value of $Y_{\text{indices}[i],0}$ to the i-th element in `labels`.
    - Count the number of `1` in `labels`.
- If the count >= 3, print out `1`. Otherwise, print `0`.
- Repeat above for all columns in D.

Part III Guessing Labels

For each column in D, do:



Part III Guessing Labels

- During Part III, at each column D_j, the print-out are your "guess" of the testing image T_j using 5-nearest neighbour algorithm.
    - If the print-out is 1, it means the algorithm thinks the image T_j is a digit 1.
    - If the print-out is 0, it means the algorithm thinks the image T_j is NOT a digt 1.

Part III Guessing Labels

- Hint: Write pseudo code for Part III before writing the real code.

Hint: Sample Output

- https://github.com/anewgithubname/MATH10017/blob/main/labs/sample_output.txt
- The visualization of digits is optional.

Final Project: Marking Criteria

- **5%:** You have submitted a C or C++ file.
- **10%:** Your code compiles and runs **without major error**.
  - It will be tested using `gcc` or `g++` in the lab pack.
  - Erratic behavior includes **crash**, **infinite loop**.
- **10-30%:** You have attempted the coursework "toward the right direction". However, your code does not produce any correct output given specific inputs within reasonable amount of runtime.
- Correctly write code for Part I, II and III will give you the remaining 70%.
  - Part I 25%, Part II 25% and Part III 20%.

Final Project: Dos and Don'ts

- You are encouraged to discuss with your classmates.
- Review relevant previous lab sessions before you start.
- You can use whatever material you can find to help you complete the task.
- You are only allowed to use standard features of C/C++.
  - You can use `stdio.h`, `stdlib.h`, `limits.h` and `math.h`.
  - If you want to use other libraries, consult with the lecturer or TA beforehand.

Final Project: Dos and Don'ts

- You are not allowed to copy code from each other or pass code to each other.
    - Similar submissions WILL trigger plagiarism investigation.

Final Project: Q&A

- We will only answer questions posted on the Blackboard forum or answering them during the lab sessions.
- We will inspect the forum regularly and try to respond in 24 hours.
- There is no guarantee after the holiday starts.
    - However, you can still communicate with each other on the forum.