Planning and Design: Naive Bayes
This assignment is easier than you think.

While there is a set of seemingly complex calculations that you will be expected to make, the overall process is quite simple and if you isolate the computation then the rest of your design is quite simple.
Planning and Design

Good designs begin with questions about the needs of the application. By answering some questions before we begin coding, we should be able to create a flexible, well-performing design and limit our need to re-factor repeatedly during the construction of our solution.

Let’s start with a few questions that will guide us.
Lecture-time Activity

I will go through a set of questions for you to consider.
Do NOT answer questions in the chat
Make a note of your own answers.
Compare your thoughts to the discussion that follows.
What things can change? How can we accommodate?

1. Size of data set
   a. Number of classes of images
   b. Number of example images
   c. Dimensions of images

2. Data source
   a. Different files (names)
   b. Other sources

3. Data format
   a. How much does your design depend on this exact format of input?
   b. How much should you care?
What do we know?

1) How much storage would you need for:
   a) 5000 images? What about 500,000?
   b) Each image using 100 x 100?
   c) more than 10 separate characters?

2) Do you need to be able to dynamically (at run-time) adapt to different image sizes?

3) What would it take to be able to adapt to more character images?
Your Design: Run-time Storage

1. In light of the potential for the dataset to change, do you need to hold all of the training data in memory at once?
   a. Is it necessary?
   b. Is it possible?
   c. Is it practical?
   d. What does it cost you?

2. Different image dimensions?
   a. Is it necessary?
   b. Is it possible?
   c. Is it practical?
   d. What does it cost you?

3. How would your application handle training data that included more than ‘0’ - ‘9’? Would it:
   a. Crash?
   b. Ignore the extraneous data?
   c. adapt?
Dataset: what makes sense?

How hard is it?

- Number of images
- Number of “labels” (classes of image)
- Dimensions of images

Costs:

Constants, variables, functions, runtime storage, compute time
Dataset: Processing the input

“However, in Naive Bayes, we make the (incorrect) assumption that these $n^2$ pixels are independent—this is the “naive” part of Naive Bayes. In other words, given that the image is a “0”, we assume that the probability of pixel $(i, j)$ being shaded, $P(F_{i,j} = 1 \mid \text{class} = 0)$, is independent of the probability of another distinct pixel $(p, q)$ being shaded, $P(F_{p,q} = 1 \mid \text{class} = 0)$.”

This means the probability of a given pixel being shaded is independent of the neighboring pixels.
Dataset: Processing the input

If we also consider that the ORDER that the training images occur has no impact on the probability of any given pixel being shaded for a given letter image then:

We can collect the NUMBER of times a given pixel is shaded for a given digit and compare that number to the total number of examples of the digit.
Dataset: Processing the input

\[ P(\text{Fi},j = 1 \mid \text{class} = 0) = \]
\[ \frac{\text{# of images belonging to class 0 where pixel (i, j) is shaded}}{\text{Total # of images belonging to class 0 in the training data}}. \]

\[ P(\text{Fi},j = 0 \mid \text{class} = 0) = \]
\[ \frac{\text{# of images belonging to class 0 where pixel (i, j) is unshaded}}{\text{Total # of images belonging to class 0 in the training data}}. \]

\[ P(\text{class} = 0) = \]
\[ \frac{\text{Total # of images belonging to class 0 in the training data}}{\text{Total # of training images}}. \]
Dataset: Processing the input

# of images belonging to class 0 where pixel (i, j) is shaded
# of images belonging to class 0 where pixel (i, j) is unshaded
Total # of images belonging to class 0 in the training data.
Total # of images belonging to class 0 in the training data.
Total # of images belonging to class 0 in the training data.
Total # of training images
Dataset: Processing the input

# of images belonging to class 0 where pixel (i, j) is shaded

# of images belonging to class 0 where pixel (i, j) is **un**shaded

Total # of images belonging to class 0 in the training data.

Total # of training images
Dataset: Processing the input

- # of images belonging to class 0 where pixel (i, j) is shaded
- # of images belonging to class 0 where pixel (i, j) is unshaded

Total # of images belonging to class 0 in the training data.

Total # of training images

Do these related to one another? Can we derive one from the other?
At the minimum, we need this data:

1. For each character that we’d like to identify (‘0’ - ‘9’), we need:
   Total number of training images that we have read of this type
2. For each PIXEL in the class, we need:
   Total number of occurrences of it being “shaded” in all images read
3. For the entire data set, we need:
   Total number of images read AND total number of unique characters

This data should absolutely NOT assume only values from this particular set and your solution should adapt in case you get a new training dataset.
It would be CONVENIENT if we could:

- accommodate an unknown number of candidate characters
  - Add them as they occur
  - Access their respective data models without knowing the identifiers in advance
- process each line of input as we read it.
- process an undetermined amount of input without recompiling
Organizing the Data: Data Structures

Consider:

1. Array
2. Vector
3. Map

Which of these would be useful for this application?

Consider how often a storage or retrieval operation would be needed. (Which accesses will you do most?)
Array Characteristics

Great for data with limited or fixed dimensions.

    auto my_array  = new int[100][100];

Easy to access data items directly by index.

    my_int = my_array[i][j];
Vector Characteristics

Is a template (STL), so the code gets written at compile-time to accommodate usage indicated at declaration.

Versatile because it grows as needed at run-time.

Great when an unknown number of items need to be added.

Uses functions to access items at specific indices (i,j).
Map Characteristics

Is a template (STL), so the code gets written at compile-time to accommodate usage indicated at declaration.

Versatile because it grows as needed at run-time.

Convenient retrieval of values by “key”

Not great for sequential or indexed storage
Data Source

Why use a stream?

- Console
- File
- Network

Why overload “>>”?  

- Consider the source
- Consider the resulting syntax
stream

What really is a stream?

Why use it?
stream: “<<” and “>>”

“<<” allows us to insert into the stream.
  
  cout: console out

“>>” allows us to extract from a stream

  cin: console in

http://www.cplusplus.com/reference/istream/istream/getline/
Overloading “>>&”

```cpp
stream >> my_function();
```

“Stream” provides us with an abstraction so that all stream functionality and syntax apply and we can ignore the particulars of the actual source. This is how we can use `getline(s, line)` and our code with be the same whether we are reading from a file or from the console.

`my_function()` accepts input from the stream but we will need to overload “>>&” to do so.
Overloading “>>”

“Friend” allows us to overload “>>” more conveniently by giving us access to the private members of the class.

https://www.tutorialspoint.com/cplusplus/input_output_operators_overloading.htm