1. **Problem 2**

The MNIST dataset is a dataset of 60,000 training and 10,000 test examples of handwritten digits, originally constructed by Yann Lecun, Corinna Cortes, and Christopher J.C. Burges. It is very widely used to check simple methods. There are 10 classes in total ("0" to "9"). This dataset has been extensively studied, and there is a history of methods and feature construc- tions at **[https://en.wikipedia.org/wiki/MNIST\_database](https://en.wikipedia.org/wiki/MNIST_database )** and at the original site, [**http://yann.lecun.com/exdb/mnist/**](http://yann.lecun.com/exdb/mnist/) . You should notice that the best methods perform extremely well.

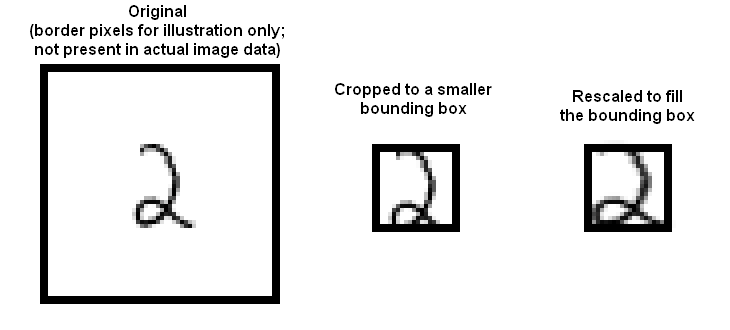
There is also a version of the data that was used for a Kaggle competition. I used it for convenience so I wouldn't have to decompress Lecun's original format. I found it at [**http://www.kaggle.com/c/digit-recognizer**](http://www.kaggle.com/c/digit-recognizer) .

If you use the original MNIST data files from [**http://yann.lecun.com/exdb/mnist/**](http://yann.lecun.com/exdb/mnist/) , the dataset is stored in an unusual format, described in detail on the page. You should begin by reading over the technical details. Writing your own reader is pretty simple, but web search yields readers for standard packages. There is reader code for R available (at least) at [**https://stackoverflow.com/questions/21521571/how-to-read-mnist-database-in-r**](https://stackoverflow.com/questions/21521571/how-to-read-mnist-database-in-r) . Please note that if you follow the recommendations in the accepted answer there at [**https://stackoverflow.com/a/21524980**](https://stackoverflow.com/a/21524980) , you must also provide the readBin call with the flag signed=FALSE since the data values are stored as unsigned integers. You need to use R for this course, but for additional reference, there is reader code in MATLAB available at [**http://ufldl.stanford.edu/wiki/index.php/Using\_the\_MNIST\_Dataset**](http://ufldl.stanford.edu/wiki/index.php/Using_the_MNIST_Dataset) .

Regardless of which format you find the dataset stored in, the dataset consists of 28 x 28 images. These were originally binary images, but appear to be grey level images as a result of some anti-aliasing. I will ignore mid grey pixels (there aren't many of them) and call dark pixels "ink pixels", and light pixels "paper pixels"; you can modify the data values with a threshold to specify the distinction, as described here [**https://en.wikipedia.org/wiki/Thresholding\_(image\_processing)**](https://en.wikipedia.org/wiki/Thresholding_(image_processing)) . The digit has been centered in the image by centering the center of gravity of the image pixels, but as mentioned on the original site, this is probably not ideal. Here are some options for re-centering the digits that I will refer to in the exercises.

* + Untouched: Do not re-center the digits, but use the images as is.
  + Bounding box: Construct a 20 x 20 bounding box so that the horizontal (resp. vertical) range of ink pixels is centered in the box.
  + Stretched bounding box: Construct a 20 x 20 bounding box so that the horizontal (resp. vertical) range of ink pixels runs the full horizontal (resp. vertical) range of the box. Obtaining this representation will involve rescaling image pixels: you find the horizontal and vertical ink range, cut that out of the original image, then resize the result to 20 x 20. Once the image has been re-centered, you can compute features.

Here are some pictures, which may help



* + **Part 2A**

Investigate classifying MNIST using naive Bayes. Fill in the accuracy values for the four combinations of Gaussian v. Bernoulli distributions and untouched images v. stretched bounding boxes in a table like this. Please use 20 x 20 for your bounding box dimensions.

|  |  |  |
| --- | --- | --- |
| **Accuracy** | **Gaussian** | **Bernoulli** |
| **Untouched images** |  |  |
| **Stretched bounding box** |  |  |

Which distribution (Gaussian or Bernoulli) is better for untouched pixels? Which is better for stretched bounding box images?

* + **Part 2B** Investigate classifying MNIST using a decision forest. For this you should use a library. For your forest construction, try out and compare the combinations of parameters shown in the table (i.e. depth of tree, number of trees, etc.) by listing the accuracy for each of the following cases: untouched raw pixels; stretched bounding box. Please use 20 x 20 for your bounding box dimensions. In each case, fill in a table like those shown below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Accuracy** | **depth = 4** | **depth = 8** | **depth = 16** |
| **#trees = 10** |  |  |  |
| **#trees = 20** |  |  |  |
| **#trees = 30** |  |  |  |