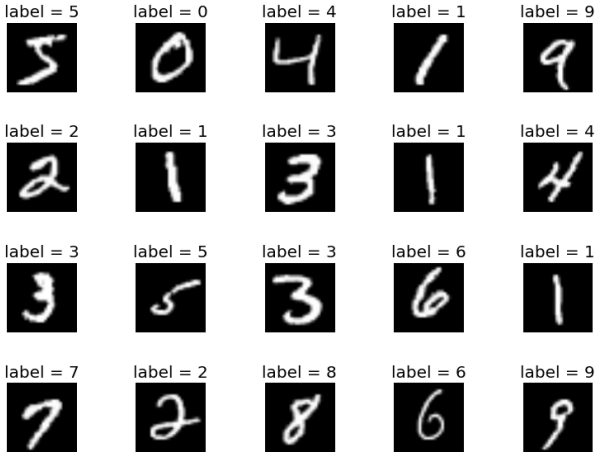
Project 1: Clustering

Release date: 29th of Jan, 2018
Due date: 11th of Feb, 2018

Description

In this project, you will implement the clustering algorithm k-means on hand written digits dataset called MNIST digit dataset. Sample images is shown below.
The MNIST data consists of 20,000 examples of 28 × 28 images of digits (i.e., numbers from 0-9).



There are two files for this project:

- **digits-raw.csv**: contains the pixel information for each image
  (first column: image id, second column: class label, remaining 784 columns: pixel features).
- **digits-embedding.csv**: contains a precomputed 2 features for each image (first column: image id, second column: class label, third and fourth columns: image features).

Design Tasks

You are required to implement k-means clustering algorithm on the MNIST dataset. You are free to design your project the way you see fit in terms of functions, parameters…etc. However, your project should meet certain design constraints. You may ***NOT*** use ready implementation for kmeans.

You must implement a Class called **MyKmeans**. ***All*** your needed functions and attributes should be included in this class, such that a user using your class should be able to use it as follows:

```
km = MyKmeans() #creating an object

km.readData('digits-raw.csv') #reading, parsing the data.
clusters = km.cluster(iterCount=50, k=5, centroids=[]) #perform clustering with initially
random centroids
SC = km.calculateSC(clusters) #calculating the SC coeff. for the clustering performed

km.readData('digits- embedding.csv')
clusters = km.cluster(iterCount=10, k=3, centroids=[10,20,30]) #perform clustering using the
provided initial centroids
SC = km.calculateSC(clusters) #calculating the SC coeff. for the clustering performed
```

As shown in the previous example, your class should implement those functions:

1. ***readData (filename)***

   Which takes a file name (digits-raw.csv or digits-embedding.csv) as an input then parse it.

2. ***cluster (iterCount, k, centroids)***

   Inputs:
   - **iterCount** (int): the number of iterations for the algorithm.
   - **k** (int): the number of clusters.
   - **centroids** (list<int>) (optional): if passed, it's an array of image ids that should be used as the initial cluster heads (centroids).
     If this parameter is not passed, then centroids are selected randomly.

   Returns:
   - The list of clusters (list<list<int>>). Each cluster is a list containing all the image ids belonging to it.

3. ***calculateSC (clusters)***

   This function takes the clusters formed and calculate the Silhouette Coefficient (SC) for it.

Clustering Tasks

Consider three versions of the data for each of the tasks below:

(i) Use the full dataset digits-embedding.csv,
(ii) Use only the subset of the data (digits-embedding.csv) consisting of the digits 2, 4, 6 and 7.
(iii) Use only the subset of the data (digits-embedding.csv) consisting of the digits 6 and 7.

For each dataset configuration (i, ii, iii), it's required

1. Visualize the images using their 2D features from (**digits-embedding.csv**), coloring the points to show their corresponding class labels. I.e. all images of the digit 1 colored in black, images of digit 2 colored in blue … and so on.
2. Cluster the data with different values of K $\in$ [2,4,8,16,32]. For each K repeat the experiment for 10 different times each with random centroids and calculate the average Silhouette Coefficient (SC) of each K after the 10 trials.
3. Construct a plot showing the average Silhouette Coefficient (SC) on y-axis as a function of K on the x-axis.
4. Choose the best value for K based on your figure, and comment on why the chosen K makes sense.

Turning in Your Work

- Submit your completed file (project1.py) on Blackboard.
- You are required to prepare a small document to report the previous plots and visualization.

Rubric

Project 1	
Task 1: Reading Data	**5%**
Task 2: Implementation of Kmeans	**30%**
Task 3: Calculation of Silhouette Coefficient (SC)	**25%**
Task 4: Successfully running experiments	**20%**
Task 5: Choosing the right k	**5%**
Task 6: Report	**15%**
TOTAL	**100%**