ENGG4030/ESTR4300 Spring 2018 Homework 3

Release date: Mar 19, 2018

Due date: Apr 9, 2018 23:59

The solution will be posted right after the deadline, so no late homework will be accepted!

Every Student MUST include the following statement, together with his/her signature in the submitted homework.

I declare that the assignment submitted on Elearning system is original except for source material explicitly acknowledged, and that the same or related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/.

Signed (Student_____) Date:_____

Name_____ SID_____

Submission notice:

- Submit your report in a single PDF document on Elearning

General homework policies:

A student may discuss the problems with others. However, the work a student turns in must be created COMPLETELY by oneself ALONE. A student may not share ANY written work or pictures, nor may one copy answers from any source other than one's own brain.

Each student **MUST LIST** on the homework paper the **name of every person he/she has discussed or worked with**. If the answer includes content from any other source, the student **MUST STATE THE SOURCE**. Failure to do so is cheating and will result in sanctions. Copying answers from someone else is cheating even if one lists their name(s) on the homework.

If there is information you need to solve a problem but the information is not stated in the problem, try to find the data somewhere. If you cannot find it, state what data you need, make a reasonable estimate of its value, and justify any assumptions you make. You will be graded not only on whether your answer is correct, but also on whether you have done an intelligent analysis.

Q1 [60 marks]: k-means Clustering

The MNIST database is a dataset of handwritten digits, comprising 60000 training examples and 10000 test examples. In this question, we will implement the k-means algorithm using the test set of MNIST dataset. The data can be downloaded here: http://yann.lecun.com/exdb/mnist/. The MNIST test set contains 10 various digits, totally 10000 instances, with representative images shown in Fig. 1. Each of the digits is a 28x28 pixel image, resulting in a 784 dimensional space.

The training set contains two files:

(1) *train-images-idx3-ubyte*: training set images (9912422 bytes)

(2) *train-labels-idx1-ubyte*: training set labels (28881 bytes)

And the testing set contains the following 2 files:

(3) *t10k-images-idx3-ubyte*: testing set images (1648877 bytes)

(4) *t10k-labels-idx1-ubyte*: testing set labels (4542 bytes)

(1) contains image instances. Rows are images and columns are pixels with values from 0 to 255. (2) contains the true labels of images in (1). You can get more detailed information of the data from http://yann.lecun.com/exdb/mnist/

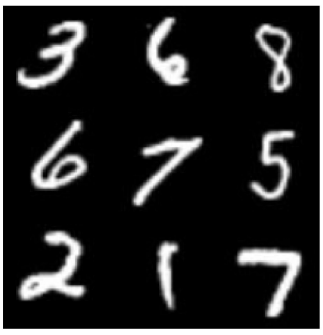

Fig. 1. An image of representative digits from the MNIST dataset.

(a) **[25 marks]** Refer to the lecture notes, you will need to implement K-means using MapReduce to perform clustering for the training-set. For this question, the number of clusters is 10. Output the vector representation of each centroid and the number of digit image assigned to each cluster like the following example:

[Centroid ID]: [Centroid Vector Representation], [Number of Digit Images]
Centroid 0: [784-dimension variable], 1000
Centroid 1: [784-dimension variable], 2000
Centroid 2: [784-dimension variable], 3000
……

Submit both your code and results.

Hint:

1. You should use multiple rounds MapReduce to implement the K-means algorithm. For each round, each mapper processes part of the training data points and store all the current centroids. The reducers will update the centroids based on "partial sum" information transferred from all the mappers.
2. Before the last round of the training process, there is actually no need to remember (or store) the cluster-membership assignment of each training data point. You just need to keep track of enough information to enable the computation of the centroid at the reducer (e.g. the partial sum of the training data points in a mapper for each cluster, and the number of training data points in a mapper that is assigned to each cluster, etc). You need to store/ output the cluster membership assignment for each data point after k-means has converged. Or you can implement another program to assign each training data point to the final clusters produced by k-means.

(b) **[20 marks]** Use the clustering results in (a) to classify the test-set of handwritten digit and calculate the accuracy of the clustering results. The true labels of images are stored in the file *t10k-labels-idx3-ubyte,* so, you can compare the results with true labels.

1. Finding the true label of each image from file *t10k-labels-idx3-ubyte.*
2. To determine the label of each cluster, you need to clean the noise images inside the cluster. The idea is to focus the label of images which are "closer" to the centroid of its assigned cluster while ignoring the other data points that are at the "peripheral zone" of the cluster. In particular, you can use the "majority label" of the data points near the centroid as the label of the cluster. Towards this end, we can consider only the m data points within a cluster that are nearest to the centroid of the cluster (in terms of Euclidean distance). We will set m = x% of the total number of data pointers within the cluster, where x is a threshold parameter. For example, if a cluster has 1000 images and x=50, then the majority of the label of the 500 images closest to the centroid of the cluster will be used as the label of the cluster.
3. Calculate the ratio of correctly clustered images to the total images in the corresponding cluster: if the true label of an image is the same as its cluster label, then it is correctly clustered. Otherwise, the image is clustered incorrectly.
4. Report the classification accuracy performance in the tables below for 4 different values of x, namely, x = 5, 10, 50 and 100%. For each table, you should use the same value of x for each of the 10 clusters output by k-means.
5. Compare the results in these four tables to determine the best value for x and explain your findings.

[Note: For part (b), you can implement the program using a single machine or MapReduce.]

Submit both your codes and results.

Table. 1. The Accuracy of Clustering Performance with x = 5%

Cluster Number	# images in the entire cluster	# of images considered (m) when determining the cluster label	Major Label of central images	# correctly clustered images	Classification Accuracy (%)
0					
1					
……					
9					
Total Set			NA		

Table. 2. The Accuracy of Clustering Performance with x = 10%

Cluster Number	# images in the entire cluster	# of images considered (m) when determining the cluster label	Major Label of central images	# correctly clustered images	Classification Accuracy (%)
0					
1					
……					
9					
Total Set			NA		

Table. 3. The Accuracy of Clustering Performance with x = 50%

Cluster Number	# images in the entire cluster	# of images considered (m) when determining the cluster label	Major Label of central images	# correctly clustered images	Classification Accuracy (%)
0					
1					
……					
9					
Total Set			NA		

Table. 4. The Accuracy of Clustering Performance with x = 100%

Cluster Number	# images in the entire cluster	# of images considered (m) when determining the cluster label	Major Label of central images	# correctly clustered images	Classification Accuracy (%)
0					
1					
……					
9					
Total Set			NA		

(c) **[15 marks]** In this part of the question, you are to perform n-fold cross validation (for n= 5 and 10, both mandatory) to further evaluate the accuracy of the handwritten-digital classification results based on the outcome of k-means clustering. Perform each n-fold cross validation exercise (for n=5 and 10) as follows:

Steps:

1. Merge the original training-set and testing set of the images in part (a) into one single data-set.
2. Randomly split the merged data-set into n equal-sized partitions of data points.
3. Choose one of the n partitions from Step 2 as the testing set while merging the remaining (n-1) partitions to become a new training data-set and then re-do part (a) and (b) to compute the corresponding classification accuracy EXCEPT that you will only use the best value of x, say x*, you have found in part (b), in determining the label of each cluster produced by k-means.
4. Repeat Step 3 for (n-1) times. Each time, you use a different partition (out of the n partitions) as a new testing set while merging the remaining (n-1) partitions as a new training set.

Submit both your code and results.

[Again for the n-fold cross validation exercises, you only need to use the best value of x, say x*, you found in part (b), in determining the label of each cluster produced by k-means.]

Table. 5. The accuracy of k-means clustering performance under 5-fold cross validation:

Testing set	Classification Accuracy (%)
Part 1	
Part 2	
……	
Part 5	
Average	

Table. 6. The accuracy of k-means clustering performance under 10-fold cross validation:

Testing set	Classification Accuracy %
Part 1	
Part 2	
……	
Part 10	
Average	

Q2 [20 marks + 30 bonus for ENGG4030]: Bernoulli Mixture Models

A Bernoulli Mixture Model (BMM) is a probabilistic model that assumes data points are sampled from a mixture of multi-dimensional Bernoulli distributions. In what follows, we will derive the optimal parameters for BMM which would maximize its log-likelihood function:

Consider a set of binary variables x_i, where $i = 1, ..., D,$ each of which is governed by a Bernoulli distribution with parameter q_i, so that

$$p(\mathbf{x}|\mathbf{q}) = \prod_{i=1}^{D} q_i^{x_i}(1-q_i)^{(1-x_i)}$$

where $\mathbf{x} = (x_1, x_2, ...x_D)^T$ and $\mathbf{q} = (q_1, ..., q_D)^T$. In other words, **x** follows a D-dimensional Bernoulli distribution where each variable x_i is independent of each other and Prob(x_i=1) = q_i which is the i-th element of **q**. Consider a mixture of K of such D-dimensional Bernoulli distributions with its density function given by:

$$p(\mathbf{x}|\mathbf{q}, \pi) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}|\mathbf{q_k}) \text{.....(*)}$$

where $\mathbf{q}=\{\mathbf{q_1},...,\mathbf{q_K}\}$ and $\pi = (\pi_1, \pi_2, ..., \pi_K)$. Now, consider a data set $\mathbf{X}=\{\mathbf{x_1},...,\mathbf{x_N}\}$ which is generated by the Bernoulli Mixture model of (*).

Now prove that the log-likelihood of $p(\mathbf{X})$ is given by: $\sum_{n=1}^{N} log \sum_{k=1}^{K} \pi_k p(\mathbf{x_n}|\mathbf{q_k})$. First note that

$$P(X) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p(x_n \mid q_k)$$

By taking logarithm of the above expression, we have:

$$\log P(X) = \log \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p(x_n \mid q_k)$$
$$= \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(x_n \mid q_k)$$

Define a variable $\gamma(z_{nk}) = \pi_k\, p(\mathbf{x_n}|\mathbf{q_k}) / \left(\sum_{j=1}^{K} \pi_j\, p(\mathbf{x_n}|\mathbf{q_j}) \right)$, which represents the "responsibility" of the $k-th$ cluster (i.e. the $k-th$ component of the Bernoulli mixture) for the data point (vector) $\mathbf{x_n}$.

Now prove that the best π_k after the first round of the EM-algorithm is $\frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$ and $\mathbf{q_k} = \sum_{n=1}^{N} \gamma(z_{nk})\ \mathbf{x_n} / \sum_{n=1}^{N} \gamma(z_{nk})$. In order to maximize with respect to π_k , we need to introduce a Lagrange multiplier to ensure that $\sum_k \pi_k = 1$. As a result, we now maximize the following quantity:

$$\log P(X) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(x_n \mid q_k) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$$

Taking derivative the above expression with respect to π_k , we have

$$\frac{\partial}{\partial \pi_k}\left(\log P(X) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)\right) = \frac{\partial}{\partial \pi_k}\left(\sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(x_n \mid q_k) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)\right)$$
$$= \sum_{n=1}^{N} \frac{\partial}{\partial \pi_k} \log \sum_{k=1}^{K} \pi_k p(x_n \mid q_k) + \lambda$$
$$= \sum_{n=1}^{N} \frac{p(x_n \mid q_k)}{\sum_{k=1}^{K} \pi_k p(x_n \mid q_k)} + \lambda$$
$$= \sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} + \lambda$$

Set it to 0, multiply both side by π_k and then sum over *k,* we have

$$\sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} + \lambda = 0$$
$$\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda \pi_k = 0$$
$$\sum_{k=1}^{K}\left(\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda \pi_k \right) = 0$$
$$N + \lambda = 0$$
$$\lambda = -N$$

Substituting $\lambda = -N$,

$$\sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} + \lambda = 0$$

$$\sum_{n=1}^{N} \frac{\gamma(z_{nk})}{\pi_k} - N = 0$$

$$\pi_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk})}{N}$$

Finding q_k

$$\frac{\partial}{\partial q_k} p(x_n \mid q_k) = x_n q_k^{x_n - 1}(1-q_k)^{1-x_n} - (1-x_n) q_k^{x_n}(1-q_k)^{-x_n}$$

$$= q_k^{x_n - 1}(1-q_k)^{-x_n}\left(x_n(1-q_k) - (1-x_n)q_k\right)$$

$$= q_k^{x_n - 1}(1-q_k)^{-x_n}(x_n - q_k)$$

$$\frac{\partial}{\partial q_k} \log P(X) = \frac{\partial}{\partial q_k} \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k p(x_n \mid q_k)$$

$$= \sum_{n=1}^{N} \frac{\pi_k}{\sum_{k=1}^{K} \pi_k p(x_n \mid q_k)} \cdot \frac{\partial}{\partial q_k} p(x_n \mid q_k)$$

$$= \sum_{n=1}^{N} \frac{\pi_k}{\sum_{k=1}^{K} \pi_k p(x_n \mid q_k)} \left(q_k^{x_n - 1}(1-q_k)^{-x_n}(x_n - q_k)\right)$$

$$= \sum_{n=1}^{N} \gamma(z_{nk}) \cdot \frac{\left(q_k^{x_n - 1}(1-q_k)^{-x_n}(x_n - q_k)\right)}{p(x_n \mid q_k)}$$

$$= \sum_{n=1}^{N} \frac{\gamma(z_{nk})(x_n - q_k)}{q_k(1-q_k)}$$

(Here we consider x_n and q_k as scalar, and the conclusion is also true is x_n and q_k are vectors)

Set $\frac{\partial}{\partial q_k} \log P(X) = 0$,

$$\sum_{n=1}^{N} \frac{\gamma(z_{nk})(x_n - q_k)}{q_k(1-q_k)} = 0$$

$$\sum_{n=1}^{N} \gamma(z_{nk})(x_n - q_k) = 0$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) x_n - \sum_{n=1}^{N} \gamma(z_{nk}) q_k = 0$$

$$q_k = \frac{\sum_{n=1}^{N} \gamma(z_{nk}) x_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$$

(a) **[20 marks]** Provide the pseudo code of an Expectation-Maximization algorithm (for a single machine OR a MapReduce cluster) which can be used to (1) estimate the parameters of a BMM model and (2) perform probabilistic clustering for a given set of observed data points. The pseudo code should include detail description on the list of input/ intermediate variables used and how each of them get updated during the E-step and M-step of each iteration.

The following part of Q2 (i.e. (b)) is OPTIONAL (carrying bonus points) for ENGG4030 students but MANDATORY for ESTR4300 students:

(b) **[20 or 30 marks]** Consider the images of the handwritten digits in the MNIST database described in Q1 of this homework. After binarization of the original grey-scale image to a bi-level black-and-white one, the color (black or white) of each pixel in an 28x28 image can then be considered as the outcome of a binary variable. As such, each image of a handwritten digit can be represented as a 784-dimensional data point generated from a mixture of 10 multi-dimensional Bernoulli components where each component is a 784-dimensional Bernoulli distribution.

Using the BMM and the EM-algorithm as discussed in part (a) of Q2, perform clustering of the training set and use the result to classify the 10,000 samples in the test-set of the MNIST database described in Q1(a) of this homework. Note that preprocessing is required to convert the grey-scale images to bi-level black and white ones. You may implement the BMM-EM algorithm EITHER in form of a standalone, sequential programme in the language of your choice, e.g. C, C++, Matlab, Java, etc OR under the MapReduce framework.

[The full-mark for a sequential implementation is 20 marks while that of a MapReduce implementation is 30 marks.]

Report the classification accuracy based on the BMM approach described above.
You should design your own method to determine the label of each cluster based on the BMM output.

You need to submit BOTH your code and the classification performance results.

Q3 [20 marks]: Dimensionality Reduction

The matrix below is a document × word matrix which shows the number of times a particular word occurs in some given documents. Here, each row represents a document which is characterized by the number of times each of the 7 words appeared in the document.

	energy	colorful	speed	elegant	spin	direction	artistic
Doc1	2	5	0	6	0	3	5
Doc2	1	4	9	0	0	3	0
Doc3	5	0	5	0	4	1	1
Doc4	0	8	0	4	8	1	0
Doc5	1	9	6	3	7	8	9
Doc6	10	2	3	0	5	2	5

(a) **[10 marks]** Originally, each document is located in a 7-D space. Using SVD, this set of documents can be approximately embedded into a 2-D space instead. Show the resultant U, Σ, V^T for achieving this goal. (Feel free to perform the SVD by hand or using any other package, e.g. Matlab.)

(b) **[10 marks]** Compute the vectors that represent the directions of the 2 new axes.
What are the coordinates of Doc 3 under the new 2-D system ?
Use these results to perform groupings i) among the 7 words and ii) among the 6 documents by clustering each of the items visually in the new 2-D space.