

# Assignment 6: Unsupervised Clustering

UVA CS 6316 :  
Machine Learning

Out: Nov. 15 2019  
Due: Dec 15th, Sun midnight 11:59pm, 2019 @ Collab

- a** *The assignment should be submitted in the PDF format through Collob. If you prefer hand-writing QA parts of answers, please convert them (e.g., by scanning or using an app like Genuis Scan) into PDF form.*
- b** *For questions and clarifications, please post on Piazza.*
- c** *Policy on collaboration:*  
*Homework should be done individually: each student must hand in their own answers. It is acceptable, however, for students to collaborate in figuring out answers and helping each other solve the problems. We will be assuming that, with the honor code, you will be taking the responsibility to make sure you personally understand the solution to any work arising from such collaboration.*
- d** *Policy on late homework:*  
*Homework is worth full credit at the midnight on the due date. Each student has three extension days to be used at his or her own discretion throughout the entire course. Your grades would be discounted by 10% per day when you use these 3 late days. You could use the 3 days in whatever combination you like. For example, all 3 days on 1 assignment (for a maximum grade of 70%) or 1 each day over 3 assignments (for a maximum grade of 90% on each). After you've used all 3 days, you cannot get credit for anything turned in late.*
- e** *Policy on grading:*  
*1: 84 points in total. 44 points for code submission (and able to run). 20 points for scatter plots in your report. 10 points for knee-finding plot in your report. 10 points for prityMetric values in your report.*  
*2: 16 points in total.*  
*The overall grade will be divided by 10 and inserted into the grade book.*

## 1 Unsupervised Learning with Clustering

In this programming assignment, you are required to implement clustering algorithm: K-means Clustering and Gaussian Mixture Model. A ZIP file has been provided (“data\_sets\_clustering.zip”) that includes two different datasets. Please follow all instructions for submitting the source code.

You are required to submit a source-code file “clustering.py” containing the necessary functions for training and evaluations. The maximum number of iterations to be performed for both algorithms is 1000.

DO NOT use scikit-learn package in this problem and please implement yours from scratch.

### 1.1 Data description

We have provided two different datasets for evaluating your clustering code.

- **Dataset 1** : The first dataset consists of height and weight data for average people and baseball players. First column contains human height (inches) and second column has human weight (lbs). The third column has true labels of samples and will be used only for evaluating the clusters you discover.

- **Dataset 2** : The second dataset is for a speech versus music classification task. This dataset has been preprocessed and first 13 columns contain 13 features extracted from audio files. Last column has true labels of samples that will be used only for evaluations.

## 1.2 load data

- (Q1) You are required to code the following function for loading datasets:  
`X = loadData(fileDj)`

## 1.3 K-means Clustering

- (Q2) Next, code the following function to implement k-means clustering:  
`labels = kmeans(X, k, maxIter)`  
 Here  $X$  is the input data matrix,  $k$  is the number of clusters and `maxIter` is the maximum number of the iterations selected by you (max value =1000).
- (Q3) Implement k-means clustering for **Dataset 1** (use first two columns in the file as input) and use `scatter()` function in the `matplotlib` package to visualize the result. The two clusters must be in different colors.
- (Q4) Implement k knee-finding (also called elbow-finding) method for **Dataset 1** and  $k = \{1, 2, \dots, 6\}$  to select value of  $k$  (number of clusters) and plot graph for  $k$  versus objective function value (e.g. Slide 99, Lecture 20).
- (Q5) Now, code the following function to calculate the purity metric for the evaluation of results:  
`purityMetric = purity(labels, trueLabels)`  
 Use this function to evaluate the results of (Q3)

## 1.4 Gaussian Mixture Model

- In this section, assume  $k = 2$ .
- (Q6) You are required to code the following function in order to implement Gaussian Mixture Model:  
`labels = gmmCluster(X, k, covType, maxIter)`  
 Here  $X$  is the input data matrix,  $k$  is the number of clusters and `maxIter` is the maximum number of the iterations selected by you (max value =1000). `covType` is a parameter for the types of covariance matrices. It can be set to two values – “diag” and “full”. “diag” means that the covariance matrices are diagonal matrices and “full” means that the covariance matrices are full matrices. In this problem, we assume that the covariance matrices are same.
- Note: (1) We assume that the two covariance matrices of two clusters are same. In another word, when the type covariance matrices are full matrices, we can estimate the same covariance matrix by

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^T (x_i - \bar{x})$$

Here  $n$  is the number of samples and  $x_i$  is the  $i$ -th sample. If you need more information, please refer to : [https://en.wikipedia.org/wiki/Sample\\_mean\\_and\\_covariance](https://en.wikipedia.org/wiki/Sample_mean_and_covariance). For calculation, you can simply use the function: `numpy.cov()`. When the type of covariance matrices are set as diagonal matrices, (i.e., we only consider the variance of variables), you can simply use the diagonal of full covariance matrices and set other entries as 0. After you have estimated these covariance matrix, you just consider them as the known covariance matrices  $\Sigma$  (since we assume different clusters share are same known covariance). Then our formulation is the same as the pages 43-46 of L21.

- Note: (2) Furthermore, the homework problems are about multivariate Gaussian distribution, which is slightly different from the Gaussian equations used in E-step of L21 slide-45.

$$p(x = x_i | \mu = \mu_j) = \frac{1}{\sqrt{(2\pi)^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1} (x_i - \mu_j)\right)$$

. Here  $p$  is the number of features. In this assignment, therefore, the equation for the E-step should be:

$$[z_{ij}] = \frac{\frac{1}{\sqrt{(2\pi)^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1} (x_i - \mu_j)\right) p(\mu = \mu_j)}{\sum_{s=1}^k \frac{1}{\sqrt{(2\pi)^p \det(\Sigma)}} \exp\left(-\frac{1}{2}(x_i - \mu_s)^T \Sigma^{-1} (x_i - \mu_s)\right) p(\mu = \mu_s)} \quad (1)$$

The equation in the M-step is the same as slide 46 in L21.

- (Q7) Implement the Gaussian Mixture Model for **Dataset 1** (use first two columns in the file as input) and use `scatter()` function in the `matplotlib` package to visualize the result. The two clusters must be in different colors. You are required to implement two types of GMM, for `covType = "diag"` and `covType = "full"`.
- (Q8) Implement the Gaussian Mixture Model for **Dataset 2** (use first 13 columns in the file as input) and use `scatter()` function in the `matplotlib` package to visualize the result.  
**Note:** In this problem, only use the first two features as input to `scatter()` function. The two clusters must be in different colors. Set `covType = "diag"`.
- (Q9) Use the purity function coded in (Q5) to evaluate the results of (Q7) and (Q8).

## 1.5 How will your code be checked?

We will run the following command: `"python clustering.py DatasetDirectoryFullPath"` and your code should print the following results:

- ONE of the scatter plots (either (Q3), (Q7) or (Q8))
- k knee-finding plot in (Q4)
- ALL purityMetric values for results obtained in (Q3), (Q7) and (Q8)

## 1.6 Submission

Please submit your source code as `"clustering.py"` and PDF report containing your written answers via collab. In the report, you should include the following contents:

- ALL scatter plots generated in (Q3), (Q7) and (Q8)
- k knee-finding plot in (Q4)
- ALL purityMetric values for results obtained in (Q3), (Q7) and (Q8)

## 2 Sample QA Questions:

**Question: 2. Decision Trees** The following dataset will be used to learn a decision tree for predicting whether a person is happy (H) or sad (S) based on the color of their shoes, whether they wear a wig and the number of ears they have.

Color	Wig	Num. Ears	(Output) Emotion
G	Y	2	S
G	N	2	S
G	N	2	S
B	N	2	S
B	N	2	H
R	N	2	H
R	N	2	H
R	N	2	H
R	Y	3	H

- (a) [2 points] What is  $H(Emotion|Wig = Y)$  (where H is entropy)?
- (b) [2 points] What is  $H(Emotion|Ears = 3)$ ?
- (c) [3 points] Which attribute would the decision-tree building algorithm choose to use for the root of the tree (assume no pruning)?
- (d) [3 points] Draw the full decision tree that would be learned from this data (assume no pruning).

The next two parts do not use the previous example, but are still about decision tree classifiers.

(e) [3 points] Assuming that the output attribute can take two values (i.e. has arity 2) what is the maximum training set error (expressed as a percentage) that any dataset could possibly have?

(f) [3 points] Construct an example dataset that achieves this maximum percentage training set error (it must have two or fewer inputs and five or fewer records).