

## Assignment 5

### Unsupervised learning as the supervision key!

#### Homeworks Guidelines and Policies

- **What you must hand in.** It is expected that the students submit an assignment report (HW5\_[student\_id].pdf) as well as required source codes (.m or .py) into an archive file (HW5\_[student\_id].zip). Please combine all your reports just into a single .pdf file.
  - **Pay attention to problem types.** Some problems are required to be solved *by hand* (shown by the ✍ icon), and some need to be implemented (shown by the 🐍 icon). Please do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
  - **Don't bother typing!** You are free to solve by-hand problems on a paper and include their pictures in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
  - **Reports are critical.** Your work will be evaluated mostly by the quality of your report. Do not forget to explain your answers clearly, and provide enough discussions when needed.
  - **Appearance matters!** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
  - **MATLAB is also allowable.** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python functions.
  - **Be neat and tidy!** Your codes must be separated for each question, and for each part. For example, you have to create a separate .py file for part b. of question 3, which must be named 'p3b.py'. (or .ipynb)
  - **Use bonus points to improve your score.** Problems with bonus points are marked by the ★ icon. These problems usually include uncovered related topics, or those that are only mentioned briefly in the class.
  - **Moodle access is essential.** Make sure you have access to Moodle, because that is where all assignments as well as course announcements are posted. Homework submissions are only made through Moodle.
- 
- **Assignment Deadline.** Please submit your work **before the end of February 2<sup>nd</sup>**.
  - **Delay policy.** During the semester, students are given only **12 free late days** which they can use them in their own ways. Afterwards, there will be a 20% penalty for every late day, and no more than four late days will be accepted.
  - **Collaboration policy.** We encourage students to work together, share their findings, and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
  - **Any questions?** If there is any question, please do not hesitate to contact us through the [Telegram group chat](#) or following email addresses: [m.ebadpour@aut.ac.ir](mailto:m.ebadpour@aut.ac.ir) and [atiyeh.moghadam@aut.ac.ir](mailto:atiyeh.moghadam@aut.ac.ir).

### 1. Delve Deep into Clustering Methods!

(20 Pts.)



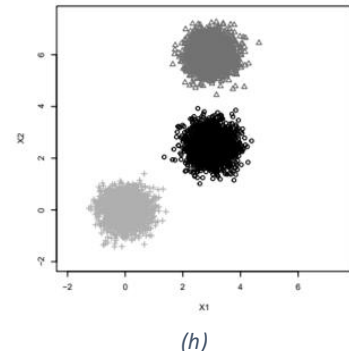
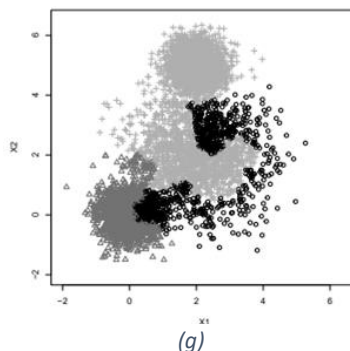
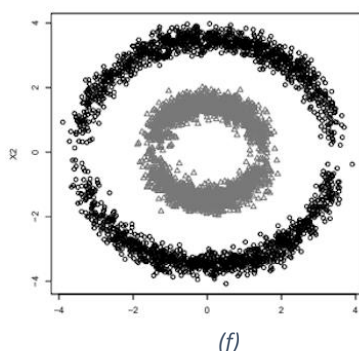
**Keywords:** *unsupervised learning, clustering, k-means clustering, hierarchical clustering, agglomerative clustering, single linkage, complete linkage.*

**Unsupervised learning** is like solving a puzzle without a guide – the system explores the data, identifying patterns and relationships on its own, without the luxury of labeled examples to follow. One powerful tool within unsupervised learning is **clustering**, akin to grouping puzzle pieces that belong together. Clustering algorithms strive to discover inherent structures within data, creating clusters that capture similarities and differences among the elements. In this problem, we will assess your fundamental comprehension of clustering algorithms, with a particular focus on K-means and hierarchical clustering.

As you may know, in some clustering algorithms, we need to employ a similarity/distance measure. There are five popular distance metrics in clustering: Euclidean distance, cosine measure, Manhattan distance, average distance (average of the squares of subtractions), and Mahalanobis distance. Suppose you want to select a distance metric for your clustering algorithm. choose the most appropriate distance metrics for the given datasets and explain why they are the optimal choices.

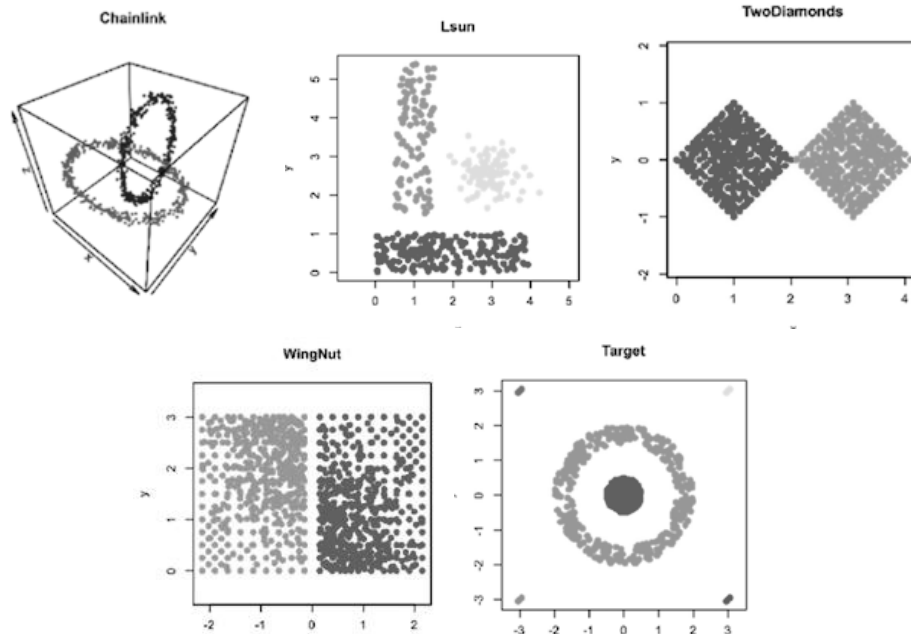
- datasets with a bunch of outliers.
- datasets with compact or isolated clusters.
- datasets where the orientation or direction of vectors is more important than magnitude.
- datasets where clusters have different variances along different dimensions.
- datasets with redundant values or high correlation between the features.

One of three clustering analyses, namely K-means clustering, hierarchical clustering with single linkage, and hierarchical clustering with complete linkage, is separately performed on each of 3 different datasets. For hierarchical clustering, the Euclidean distance is used for the interpair dissimilarity measure. The data points and the clustering results are plotted below, with color (and shape) representing cluster membership. For each case, list the clustering methods that could have generated the result.



Consider the following datasets, and assume we want to apply k-means clustering, hierarchical clustering with single linkage, and hierarchical clustering with complete linkage on them. What

would be the final clusters in each of these datasets? Color them on the images and briefly explain. (choose k arbitrarily)



- i. chainlink dataset.
- j. Lsun dataset.
- k. TwoDiamonds dataset.
- l. WingNut dataset.
- m. Target dataset.

Finally consider the figure below:

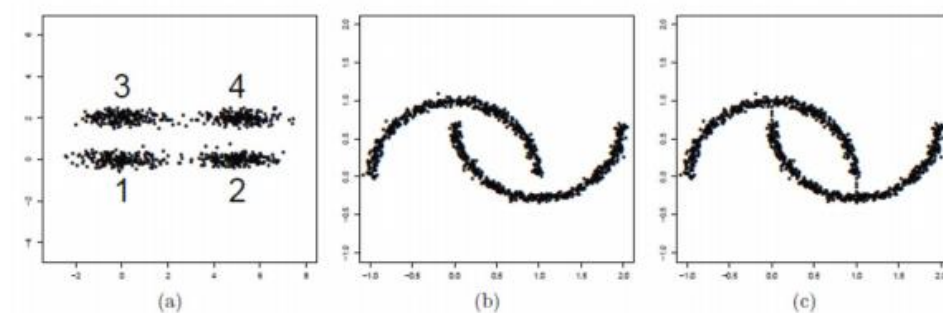


Figure 1 three assumed data for clustering

- n. Consider the data in (a). What would be the result if we extract two clusters from the tree given by hierarchical clustering on this dataset using single linkage? Describe your answer in terms of the labels 1-4 given to the four 'clumps' in the data. Do the same for complete linkage and average linkage.
- o. Which of the three types of linkage (if any) would successfully separate the two 'half-moons' in (b)? Which about (c)? Briefly explain your answer.

## 2. Solving clustering problems by hand!

(15 Pts)



**Keywords:** *unsupervised learning, clustering, k-means clustering, hierarchical clustering, single linkage, average linkage, complete linkage.*

Clustering methods play a pivotal role in uncovering underlying structures within datasets, enabling the categorization of similar data points and revealing patterns that might otherwise remain hidden. Various clustering algorithms employ distinct approaches to group data based on similarity or dissimilarity measures.

The popular k-means algorithm partitions data into k clusters by minimizing the within-cluster variance, whereas hierarchical clustering builds a tree-like structure, **dendrogram**, representing the hierarchical relationships among data points. Each clustering method comes with its strengths and weaknesses, making the choice dependent on the specific characteristics of the data and the desired outcome of the analysis. In this problem we are going to solve some basic clustering problems.

First, assume the dataset X as below:

<b>X1</b>	5.9	4.6	6.2	4.7	5.5	5	4.9	6.7	5.1	6
<b>X2</b>	3.2	2.9	2.8	3.2	4.2	3	3.1	3.1	3.8	3

perform a K-Means clustering on the given dataset, where k is set to 3 and the centers of 3 clusters are initialized as  $\mu_1 = (6.6, 3.7)$ ,  $\mu_2 = (6.2, 3.2)$ ,  $\mu_3 = (6.5, 3.0)$ . Use the Manhattan distance as the distance function.

- How many iterations are required for the clusters to converge? what are the centers of clusters after the clustering converges?
- As you know, k-means clustering is sensitive to initialization. Is the recommended initialization a good choice? If not, recommend another initialization and execute the algorithm with your preferred initialization. How many iterations are needed for the clusters to converge? Is there a suitable method for finding the initial centers?

Next, consider the Euclidian distance matrix given below of a sample dataset consisting of five two-dimensional points.

	<i>P1</i>	<i>P2</i>	<i>P3</i>	<i>P4</i>	<i>P5</i>
<i>P1</i>	0.0	-	-	-	-
<i>P2</i>	0.18	0.0	-	-	-
<i>P3</i>	0.20	0.38	0.0	-	-
<i>P4</i>	0.24	0.10	0.36	0.0	-
<i>P5</i>	0.35	0.28	0.15	0.22	0.0

- c. apply single linkage hierarchical clustering to the distance matrix. show the steps involved and draw the dendrogram.
- d. apply complete linkage hierarchical clustering to the distance matrix. show the steps involved and draw the dendrogram.
- e. apply average linkage hierarchical clustering to the distance matrix. show the steps involved and draw the dendrogram.

### 3. Support Vector Machines: An All-Rounder Tool!

(15+5 Pts)



**Keywords:** linear discriminant functions, support vector machine, SVM, credit scoring, kernel SVM, gaussian kernel.

The development of credit scoring models is crucial for financial institutions to differentiate between defaulters and non-defaulters when making credit-granting decisions. Pattern recognition techniques have demonstrated successful performance in credit scoring. SVMs are particularly beneficial for this type of problem because they excel at handling non-linear relationships in data and addressing class imbalances. In this problem, our objective is to delve deeper into the understanding of Support Vector Machines (SVMs) and gain a more comprehensive knowledge of their capabilities.

Figure 3 plots decision boundaries of SVM classifiers using different kernels and/or different slack penalty  $C$ . The data points labelled +1 and -1 are represented by circles and triangles, respectively. The support vectors for each decision boundary is illustrated as solid circles and triangles. For each of the following scenarios, find the matching plot from Figure 3 (there is a one-to-one match). Justify each choice briefly in 1-2 sentences.

- A hard-margin linear SVM.
- A soft-margin linear SVM with  $C = 0.1$ .
- A soft-margin linear SVM with  $C = 10$ .
- A hard-margin kernel SVM with  $k(x, z) = (x^T z)^2$ .
- A hard-margin kernel SVM with  $k(x, z) = \exp(-||x - z||^2)$ .

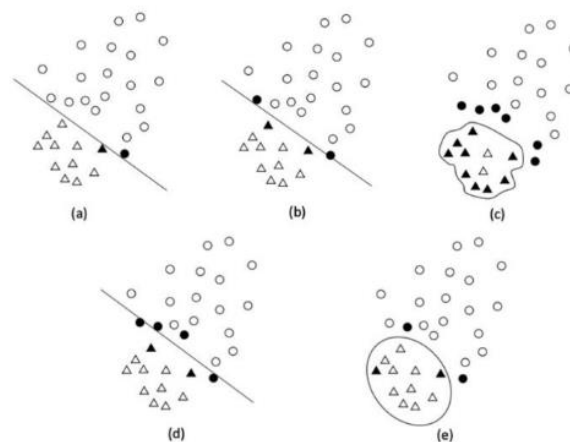


Figure 3: SVM Decision Boundaries

Now assume having the following two-dimensional training data:



Figure 2: According to 'AI Events,' the Head of New Technologies at the Central Bank of Iran has announced a plan to create models for reading financial reports using machines.

$x_1$	$x_2$	$y$
-3	9	-1
-2	4	-1
-1	1	-1
0	-3	1
1	1	1
2	4	-1
3	9	-1

- f. Plot the data (no need to provide any code if you do this in Python). Is the data linearly separable?
- g. Recall that the dual formulation of the SVM classifier for a dataset of size  $n$  is:

$$\arg \max_{a_1, \dots, a_n} \sum_{i=1}^n a_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n a_i a_j y_i y_j x_i \cdot x_j, \quad \text{subject to } a_i \geq 0 \text{ for all } i, \sum_{i=1}^n a_i y_i = 0$$

Solve for  $(a_1, \dots, a_7)$  for the provided dataset. (Hint: Using the plot in the previous section and your knowledge of SVMs, try to simplify the (dual) objective before doing any calculus.)

- h. For your SVM, compute the corresponding weight vector ( $w \in \mathbb{R}^2$ ) and bias  $t$ . Superimpose a plot of the SVM onto the scatter in (f). What is the margin for this model?

Consider the task of training a support vector machine using the Gaussian kernel  $K(x, z) = \exp(-\|x - z\|^2 / s^2)$ . We will show that as long as there are no two identical points in the training set, we can always find a value for the bandwidth parameter  $s$  such that the SVM achieves zero training error.

- ★ i. The decision function learned by the support vector machine can be written as:

$$f(x) = \sum_{i=1}^n a_i y_i k(x_i, x) + b.$$

Assume that the training data  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  consists of points which are separated by at least a distance of  $\epsilon$ ; that is,  $\|x_j - x_i\| \geq \epsilon$  for any  $i \neq j$ . Find values for the set of parameters  $\{a_1, \dots, a_n, b\}$  and Gaussian kernel width  $s$  such that  $x_i$  is correctly classified, for all  $i = 1, \dots, n$ . [Hint: Let  $a_i = 1$  for all  $i$  and  $b = 0$ . Now notice that for  $y \in \{-1, +1\}$  the prediction on  $x_i$  will be correct if  $|f(x_i) - y_i| < 1$ , so find a value of  $s$  that satisfies this inequality for all  $i$ .]

- ★ j. Suppose we run a SVM with slack variables using the parameter  $s$  you found in part (i). Will the resulting classifier necessarily obtain zero training error? Why or why not? A short explanation (without proof) will suffice.



#### 4. Doing Some Image Processing Using K-Means

(20 Pts.)



**Keywords:** *Clustering Problem, Supervised Learning, K-Means Clustering, Image Segmentation, Image Compression*

Although **K-Means** might sound like a basic **Clustering** method, it has various applications in different areas, from machine learning and computer vision to astronomy and even agriculture. As an example, one can count several image processing problems that can be easily dealt with using K-Means method.

In this problem, you are to examine K-Means on two image processing task, namely **Image Segmentation** and **Image Compression**. The first deals with algorithms and methods which try to partition a digital image into several parts in order to change it into a more meaningful and easier to analyse representation, while the second handles the process of reducing the cost for storage or transmission of digital images.



Figure 4 An example of image segmentation, with initial image (Top) and the result of k-means segmentation with k set to 5 (Bottom)

Let's start with a grayscale image segmentation. In this part, you are working with the image "trump\_new\_hairstyle.png" located in "inputs/P4" folder.

- a. Implement a function with a header as follows:

```
[centres,mask]=kmeans_seg_gray(img_gray,k)
```

This function takes a  $n \times m$  grayscale image `img_gray` and a constant value `k`, and applies K-Means algorithm upon the input image in such a way that it returns a  $nm \times 1$  vector `centres` which stores final centre points, and a  $n \times m$  matrix `mask` which determines which of the `k` clusters each pixel belongs to. Now apply it on the given image, and using the outputs `centres` and `mask`, create a new  $n \times m$  image in which the value of every pixel is set to its cluster centre. Display the results for  $k = 3, 5, 7, 9$ . Attach the resultant images to your report as well.



Figure 5 The grayscale input image given for part a.

**Hint:** Here, you must treat each pixel value as a point in 1-D space.

Now, let's deal with a color image in a similar fashion. Read the image "bald\_donald.png", located in the same folder.

- b. Implement a RGB version of the function in part a. with the following header:

```
[centres,mask]=kmeans_seg_rgb(img_rgb,k)
```



Figure 6 An image given for color image segmentation



It takes a  $n \times m \times 3$  color image `img_rgb` and a constant value  $k$ , and works exactly the same as the previous function, only the output variable `centres` is now a matrix of the size  $nm \times 3$ . Once again, apply your function upon the given input image, and display the segmentation results for  $k = 3, 5, 7, 9$ , and attach them to your report too.

Finally, read the image “leo.png” in the mentioned folder.

- c. Apply K-Means image segmentation on the input image, with the variable  $k$  set to  $k = 32, 64, 128, 192$ . Display the segmented images. Can you notice any difference? Save the images (not figures) and report their sizes, and discuss whether K-Means can be useful in image compression or not.

**Note 1:** This part might sound a bit tedious, but it will worth the effort.

**Note2:** Save your results in the same format as the input image (.png) so that the comparison becomes meaningful.

**Note:** Please use your own implementation of algorithms.



Figure 7 A 600x900 input image, with the approximate size of 1Mb, given for part c.

### 5. K-Means Maneuver in Image Processing Territory

(25 Pts.)



**Keywords:** *K-Means Clustering, Vector Quantisation, Color Extraction, Image Segmentation, Image Compression*

Despite its simplicity, **K-Means** can be applied to various machine learning tasks. From document classification and data analysis to fraud detection and collaborative filtering, this algorithm still challenges even newly introduced methods in different applications, which many of them are known to be state-of-the-art.











The goal of this problem is to get you more familiar with some of the thing you can do with K-Means in the area of image processing. Given below are three different, but structurally similar image processing tasks and you are required to propose a method to solve them using K-Means.

- a. **Color Extraction** is the process of identifying and extracting key colors in images. It gives a better visual understanding of images while providing significant features for other computer vision tasks.

Load the image “tiny\_trump\_1.jpg”. Use K-Means to extract its 3, 5, 7 and 9 main colors. Display these colors properly in separate square shapes.



(a)

Proportional palette	Hex color	Area	Closest color name
	#211b1b	31.7 %	Bokara Grey (Grey)
	#c6b293	23.5 %	Sour Dough (Brown)
	#784028	14.1 %	Copper Canyon (Brown)
	#d1c6c1	9.9 %	Swiss Coffee (Grey)
	#6a5a4d	8.9 %	Domino (Brown)
	#464c2b	4.4 %	Walouru (Green)
	#4e301f	3.2 %	Indian Tan (Brown)
	#c3a36a	2.5 %	Putty (Yellow)
	#b57a67	1.2 %	Toast (Brown)
	#847e8b	0.7 %	Topaz (Grey)

(b)

Figure 8 An example image with its extracted main colors, called “palette”, sorted by area they occupy in the input image. The result is obtained using an online tool called TinEye ([here](#)) (a) Original image (b) Color extraction results.

- b. **Image Segmentation** is a common technique in image processing in which the goal is to divide an image into multiple parts or regions, often based on the characteristics of the pixels in the image.

Load the image “tiny\_trump\_2.jpg”. Use K-Means to divide the given image into 3, 5, 7 and 9 partitions.



(a)



(b)

Figure 9 Image segmentation applied to an example input image (a) Original image (b) The result of image segmentation

- c. **Image Compression** refers to techniques used for minimising the size of an image using the image data which are repeated in the image.

Load the image "tiny\_trump\_3.jpg". Use K-Means to reduce the size of the input image to %50, %75, %90 and %97 of the original image size (in KB).

**Hint:** You must find appropriate values for parameter  $k$ .

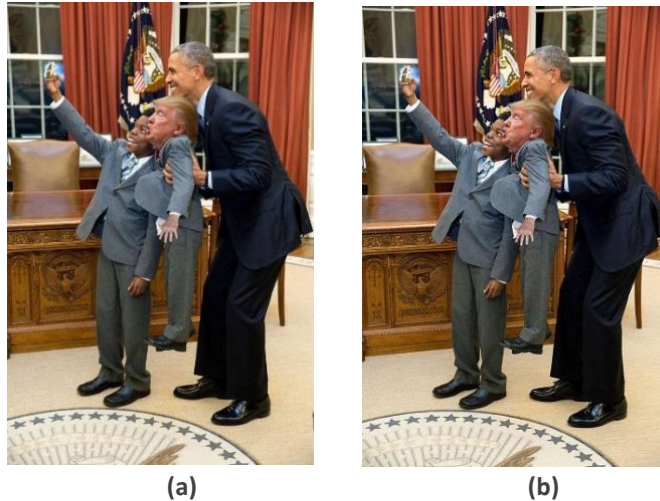


Figure 10 A compression technique has reduced the image size from 211KB to 83KB. Note that the difference is not properly noticeable here (a) Original image (b) Compressed image.

*Good Luck!*

*Mohsen Ebadpour, Atiyeh Moghadam, Romina Zakerian*