Exercise 11: Feature Selection and Clustering

May 10, 2016

Load the provided dataset for this exercise using `load ex11`. There, you'll find two variables: `data` – a 300×2 matrix representing three hundred 2-dimensional data points, and `true_labels` – a 300×1 vector of integers associating each data point to one of the clusters.

1 Clustering using Lloyd's algorithm

Implement the k-means algorithm using Lloyd's method.

1.1 Initialize the centroids ($k = 3$)

Randomly pick 3 distinct data points (out of the $N = 300$) to be the initial centroids.

IMPORTANT NOTICE: Run the command `rng(2015);` before choosing the centroids. This command initializes your computer's random number generator and allows us to compare you answers to ours.

Hint: Use the `randperm(N, k)` function. Note that it would be wrong to just randomly pick 3 numbers between 1 and 300, because then you might get the same number twice.

1.2 Calculate distance matrix

Calculate the distance between each data point and each centroid, and store the results in a $N \times k$ matrix.

Hint: Use the function `pdist2(X, Y, 'euclidean');` where X is the data matrix and Y is the matrix containing the centroids.

1.3 Associate each data point to the closest centroid

Find the index of the closest centroid to each point. A cluster will be the set of all the points associated to a specific centroid.

Hint: `[~, closest_centroid] = min(distances, [], 2);`, where `distances` is the result from the previous question. Note that the `[]` tells MATLAB to ignore the second argument and then use 2 as the dimension over which to minimize. Try `min(distances, 2)` and see what happens.

1.4 Move each centroid to the mean value of the cluster

Using a loop (`i = 1:k`), calculate the mean of cluster i and change the value of centroid i to that.

Hint: `mean(data(closest_centroid==i, :), 1);`

1.5 Plot results

Use a scatter plot to visualize the different clusters.

Hint: Use `scatter(data(:,1), data(:,2), [], closest_centroid)`.

1.6 Repeat steps 1.2 to 1.4 several times until convergence

After each iteration, use the scatter plot to see if there was any change in the clusters*. If you don't see any change, that means the algorithm has converged on a solution. How many iterations did it take?

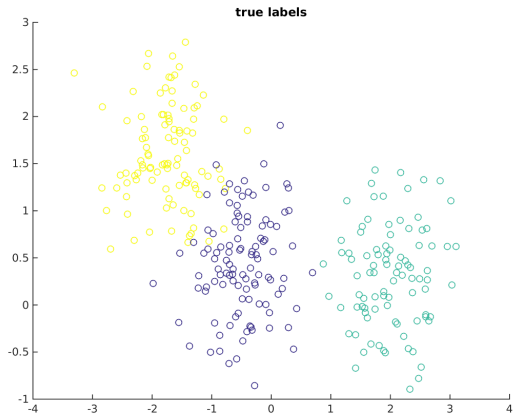
Hint: You can use MATLAB's ≪Run Section≫ option to do the iterations. To mark a section, place a double percent sign (%%) before and after the code you want to run. Also, if you use the command `figure(1);` before you make the plot, the program will replace the previous plot with the new one without creating a new window.

* **Bonus**: write code that checks when the algorithm converged instead of plotting it and making the comparison yourself.

convergence after 8 iterations

1.7 Compare the clusters to the `true_labels`

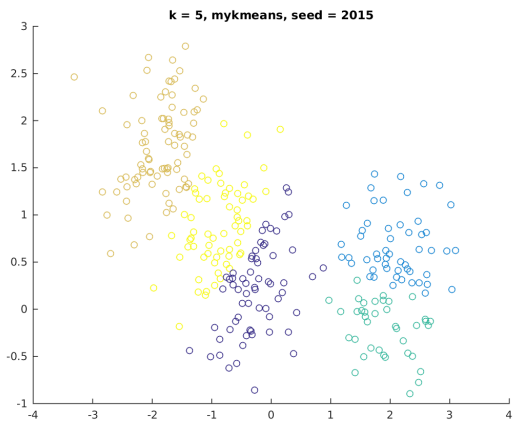
In a separate figure, make another scatter plot with the `true_labels` instead of the results from the clustering. Note that the colors can be different because the number of each cluster is arbitrary. Is the rightmost cluster the same in both plots (apart from the color, of course)? Do you spot any differences of the labels in the other two clusters? If so, why does the k-means algorithm converge to this different solution?



The rightmost cluster is identical. The other two are mixed in the true labels, but the k-means algorithm can only have strict boundaries between the clusters and therefore the points from cluster 1 that are closer to the center of cluster 2 are labelled as 2 and vice versa.

1.8 What happens when k is too large?

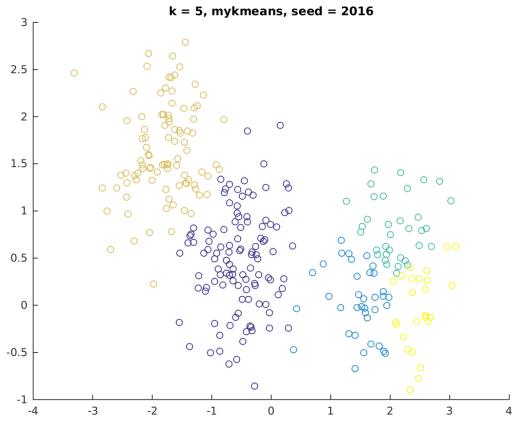
Repeat all the previous steps with $k = 5$ (remember to initialize the random number generator again using `rng(2015)`). How many iterations did it take the algorithm to converge this time? Compare the clusters to the true labels again and describe what you see.



Converged after 22 iterations. Two of the clusters were split into two sub-clusters.

1.9 Importance of the initial choice of centroids

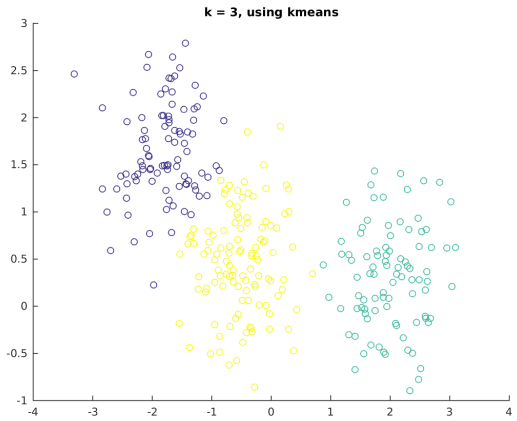
Repeat the same analysis but with another random seed – `rng(2016)`. Did the algorithm converge to the same 5 clusters? Compare again the clusters to the true labels again and describe what you see.



Converged after 9 iterations. Now the rightmost cluster was split into 3, and the two others are like in the $k = 3$ case.

1.10 Using the built-in MATLAB function

Use the *Statistics and Machine Learning Toolbox* implementation of `kmeans(X, k)`, with $k = 3$, and compare the clusters you get from it to the ones from Question 1.6.



The result is identical to the one in Question 1.6.

2 Hierarchical Clustering and Dendrograms

Load again the gene expression data from exercise 9 (`load ex9`). Make a `12x4297` matrix containing the expression data from the following strains: `strain02, strain03, strain06, strain08`. Note that clustering algorithms always treat the rows as the different data-points, so you should transpose the expression matrices.

2.1 Clustering using k-means

First, initialize the random number generator again using `rng(2015);`. Then, apply the function `kmeans(X, 4)` on the combined data matrix (`X`). Did the algorithm successfully identify the 4 groups?

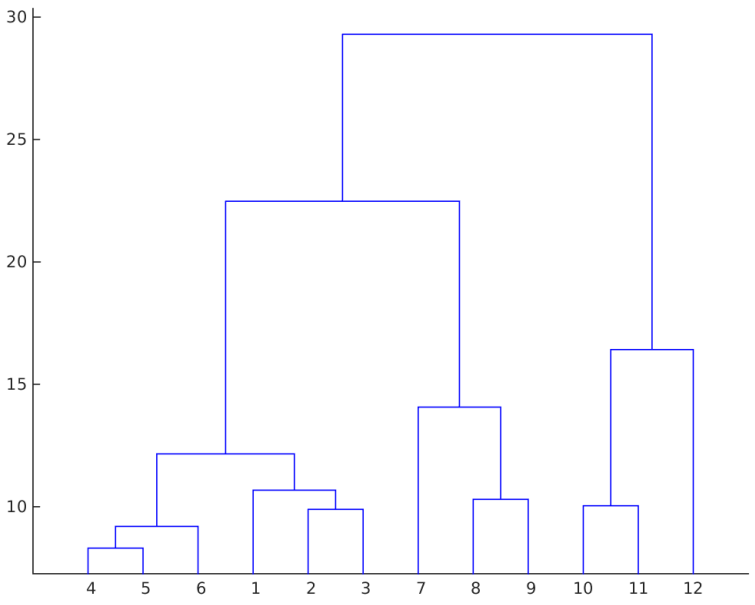
Hint: Note that now the data is not 2D (but rather 4297 dimensional), therefore you cannot visualize it in a scatter plot. Therefore, you must look at the cluster indices returned by k-means directly.

The cluster indices are: 2 2 2 2 2 2 4 1 1 3 3 3. This is not the answer we expect, since strains 02 and 03 merged into one cluster, and strain06 was split into two clusters.

2.2 Hierarchical clustering

Create a hierarchical cluster tree for the same data with `Z = linkage(X, 'average')`. This function returns a matrix `Z` with three columns which represents the tree. To visualize the tree use `dendrogram(Z)`.

Did the hierarchical clustering algorithm successfully identify the clusters? Among the four clusters (strains), which two are the closest? Which strain is the most distinct relative to the other 3?



Yes, the dendrogram shows that each triplet of samples clusters together before merging with the next cluster. Strain02 and strain03 are closer together with an average distance of about 12. Strain08 is the most distinct with an average distance of almost 30 from the other clusters.