Project

Name: Aris Podotas

University: National and Kapodistrian University of Athens

Program: Data Science and Informaion Technologies

Specialization: Bioinformatics - Biomedical Data

Lesson: Clustering Algorithms

Date: November 2024

Contents

1 Preface

Before any of the further analysis a look at the files provided in the report are all the files that were originally sent with the description along with files from previous homework and a file named *main.m* that implements the Matlab code that generates all the data transformations, a folder named *Images* with all produced figures, a file named *process.mat* for saved final data after all transformations were considered, a file *tests.m* that has the trials done for the timing and comparison of the clustering algorithsm with eachother, a file named *clustered.mat* that hold the variables after the whole analysis including average times of algorithm executions, a file named *fuzzy_c_means.m* (Theodoridis *et. al.,* 2010) for the implementation of the Fuzzy algorithms, a file named *GMDAS.m* (Theodoridis *et. al.,* 2010) for the implementation of the probabilistic algorithms and a file named *report.pdf* that contains this report. It is recommended to read this report with the *main.m, tests.m* files open along side to look at the corresponding matlab code for each step. We operate starting from the *main.m* file for all data transformations but since the output of this file will be provided one can just go straight to the *tests.m* file.

The files will also contain comments explaining what goal each function has and the method used.

2 Feeling the data

2.1 Missing data

It is stated in the description of the project that: Only the pixels with nonzero class label will be taken into consideration in this project. This will be considered a form of missing data.

How will we handle this missing data? This dataset is adequately large for pruning of missing values to occur. A variable that copies the original data and is then filtered for missing data will be made. The missing data needs to be handled first in this analysis since it is explicitly stated that missing values will not be considered and thus any representation of the data before removing these values will be improper.

An interesting note is that the output is 13.908 data vectors. The full data had 22.500 which means that we had missing data for 8.592 feature vectors (zero label missing data).

Is this the only missing data in our sample? Not necessarily, because values for the features could contain **NaN** or **missing**. Once we remove the missing data we know we have (zero labels) we apply a check for other missing data and handle it. Our approach for other missing data should be different than before (zero label) since zero label missing data is explicitly stated for removal, however with other missing data values we can imploy other missing data handling. In this dataset there are no **NaN, missing** fields (see appropriate function in the *main.m* file). Should there have been we would know that the missing values would have been within one of the *spectral bands* and we could have handled it with substitution of the *mean* or *median* of the feature that was missing so as to not remove the whole data vector. Both the labels and the values were checked.

2.2 Data type

This segment is about the values our data takes (discreet or continuous). Actually we cannot look at the data itself for, large datasets like the one provided do not get visualized in the variable previewer. The way this will be overcome is by utilizing the knowledge of the dataset. We have a-priori knowledge for the features of the dataset due to the nature of the problem. Each feature takes continuous values within a *spectral band.*

It is generally a good idea to view the distributions of the data and the features. All histograms have been made in a folder (Images) created by the *main.m* file. All zero label fields have been removed.
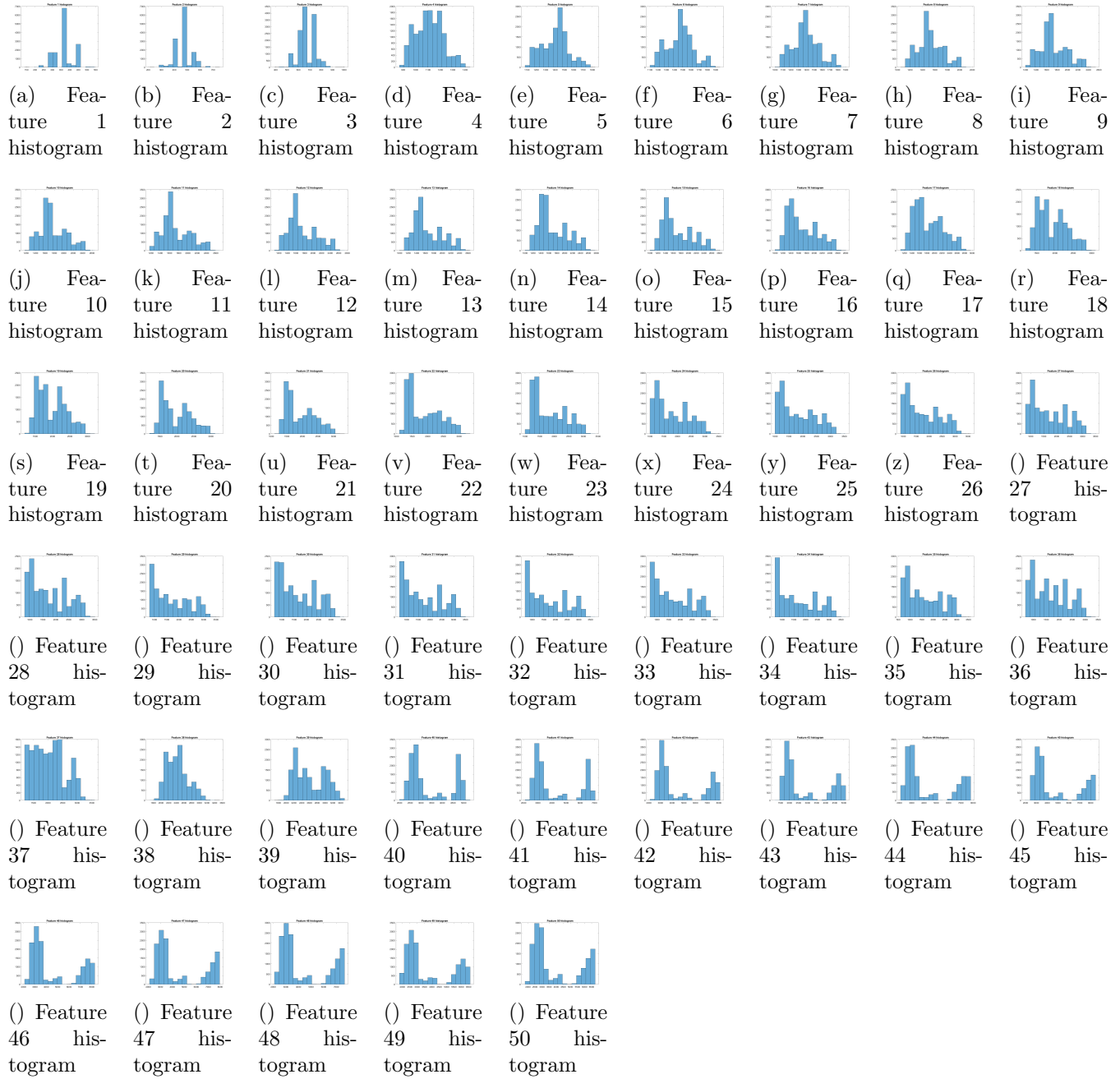
(a) Feature 1 histogram
(b) Feature 2 histogram
(c) Feature 3 histogram
(d) Feature 4 histogram
(e) Feature 5 histogram
(f) Feature 6 histogram
(g) Feature 7 histogram
(h) Feature 8 histogram
(i) Feature 9 histogram

(j) Feature 10 histogram
(k) Feature 11 histogram
(l) Feature 12 histogram
(m) Feature 13 histogram
(n) Feature 14 histogram
(o) Feature 15 histogram
(p) Feature 16 histogram
(q) Feature 17 histogram
(r) Feature 18 histogram

(s) Feature 19 histogram
(t) Feature 20 histogram
(u) Feature 21 histogram
(v) Feature 22 histogram
(w) Feature 23 histogram
(x) Feature 24 histogram
(y) Feature 25 histogram
(z) Feature 26 histogram
() Feature 27 histogram

() Feature 28 histogram
() Feature 29 histogram
() Feature 30 histogram
() Feature 31 histogram
() Feature 32 histogram
() Feature 33 histogram
() Feature 34 histogram
() Feature 35 histogram
() Feature 36 histogram

() Feature 37 histogram
() Feature 38 histogram
() Feature 39 histogram
() Feature 40 histogram
() Feature 41 histogram
() Feature 42 histogram
() Feature 43 histogram
() Feature 44 histogram
() Feature 45 histogram

() Feature 46 histogram
() Feature 47 histogram
() Feature 48 histogram
() Feature 49 histogram
() Feature 50 histogram

Figure 1: Histograms of all features without the zero label data

(a) Feature 51 histogram
(b) Feature 52 histogram
(c) Feature 53 histogram
(d) Feature 54 histogram
(e) Feature 55 histogram
(f) Feature 56 histogram
(g) Feature 57 histogram
(h) Feature 58 histogram
(i) Feature 59 histogram
(j) Feature 60 histogram
(k) Feature 61 histogram
(l) Feature 62 histogram
(m) Feature 63 histogram
(n) Feature 64 histogram
(o) Feature 65 histogram
(p) Feature 66 histogram
(q) Feature 67 histogram
(r) Feature 68 histogram
(s) Feature 69 histogram
(t) Feature 70 histogram
(u) Feature 71 histogram
(v) Feature 72 histogram
(w) Feature 73 histogram
(x) Feature 74 histogram
(y) Feature 75 histogram
(z) Feature 76 histogram
() Feature 77 histogram
() Feature 78 histogram
() Feature 79 histogram
() Feature 80 histogram
() Feature 81 histogram
() Feature 82 histogram
() Feature 83 histogram
() Feature 84 histogram
() Feature 85 histogram
() Feature 86 histogram
() Feature 87 histogram
() Feature 88 histogram
() Feature 89 histogram
() Feature 90 histogram
() Feature 91 histogram
() Feature 92 histogram
() Feature 93 histogram
() Feature 94 histogram
() Feature 95 histogram
() Feature 96 histogram
() Feature 97 histogram
() Feature 98 histogram
() Feature 99 histogram
() Feature 100 histogram

Figure 2: Histograms of all features without the zero label data

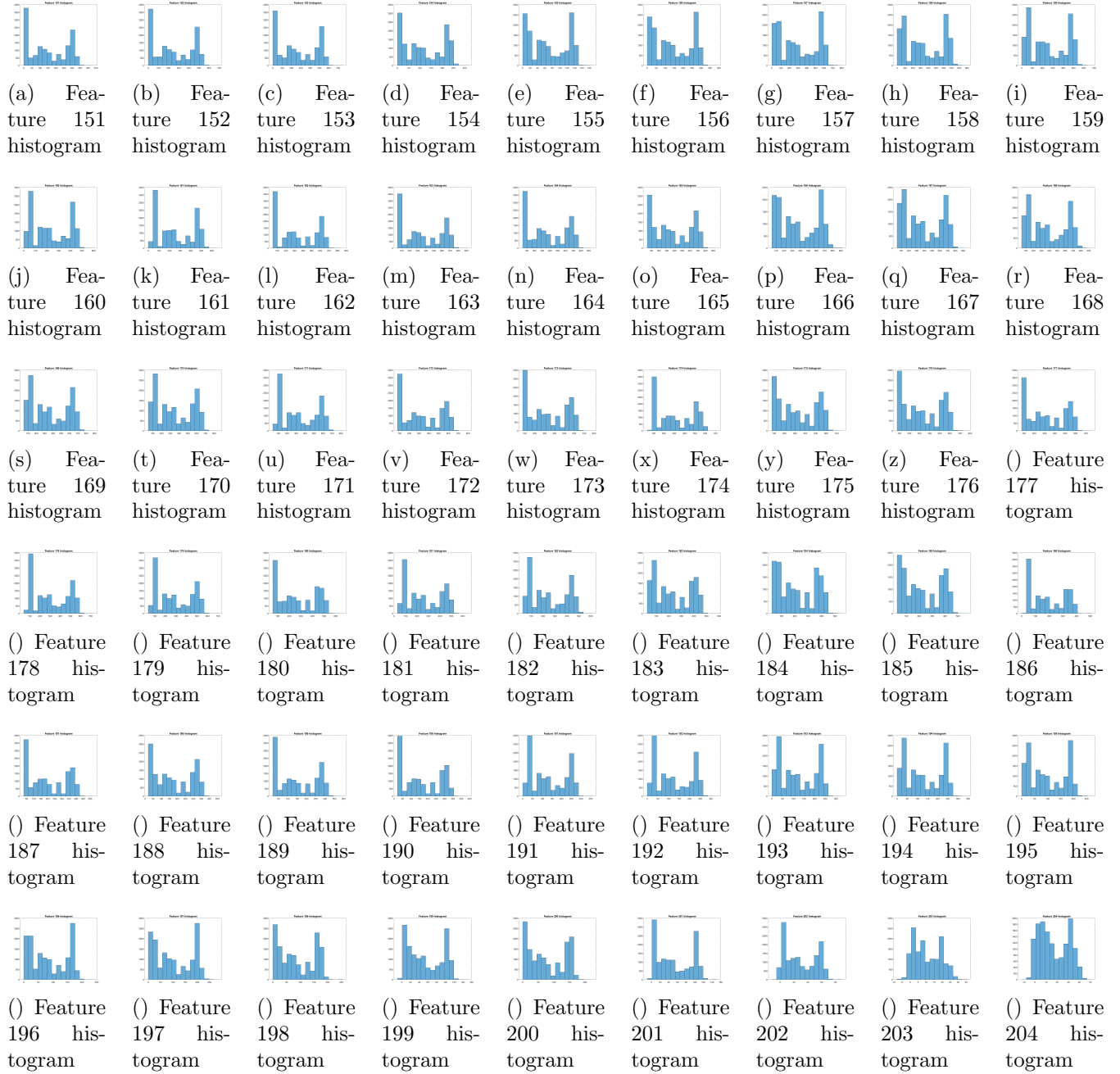Figure 3: Histograms of all features without the zero label data

Figure 4: Histograms of all features without the zero label data

The number of bins is determined using Sturge's rule

$$\text{Bins } b \text{ should be } b = \lceil log_2(N) + 1 \rceil, \quad \text{Sturges (1926)}$$

Actually the problem of representation plagues this dataset because none of the outputs would fit neatly into a format on the page so all outputs are only available withing the varibale browser in Matlab (for instance the mean).

From Figure 1 to 4 we see that there is a wide range of distributions within different features, most are not Gaussian. As a result the median of the features will be useful

and has been calculated before in the corresponding function in the *main.m* file. Despite this, from all the histograms one can see features that follow a similar distribution, and most features seem to be within some distribution grouping.

This function (call) will be commented out in the provided *main.m* file for time efficiency further down the analysis and since only one run of the function is required to prodce the above images so it can be omitted when viewing the report along side the *main.m* file.

2.3   Cross correlation

We actually cannot take the cross correlation terms using the *corrcoef* function in matlab for this dataset since the dimensions of the data are not congruent with the function. A sort of cross correlation will happen when we do the principaled component analysis.

Before moving to the Feature selection/transformation, out of curiosity we visualize the data using the indexes of the values. For details on exactly what is being represented read the corresponding *main.m* function. This function is very computationally demanding for all features so only the first feature was visualized since the rest would contain the same zero label data in a different range.
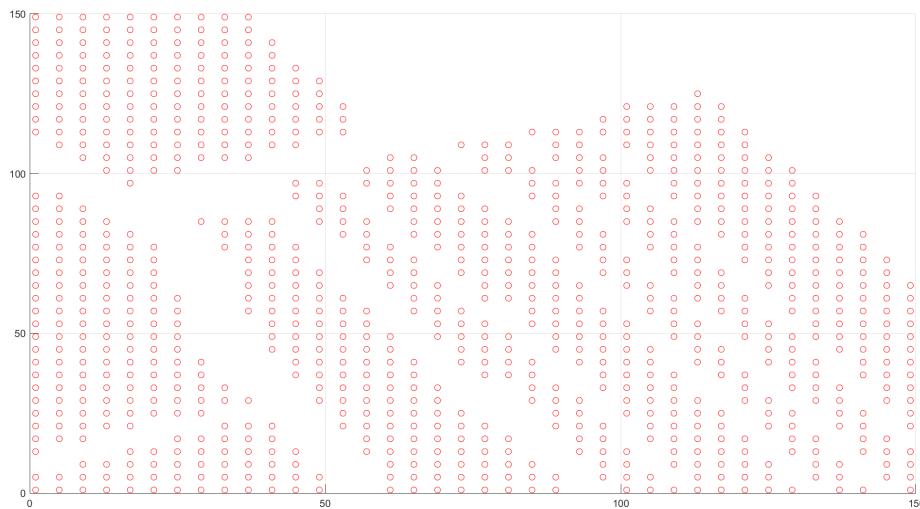

Figure 5: Overview of pixels with non zero label data

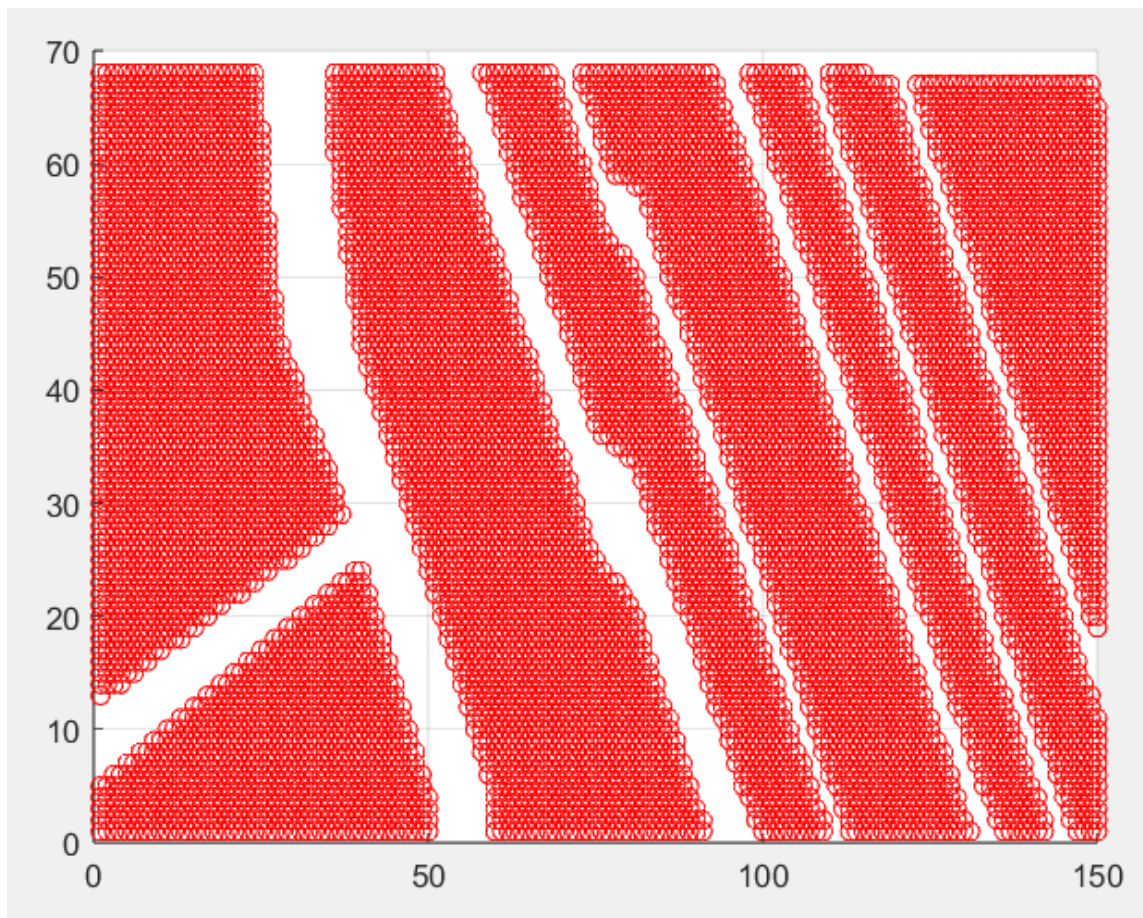and in a little higher resolution but stopped early.

Figure 6: Overview of pixels with non zero label data with a higher resolution but smaller bounding area of the image (cropped due to time efficiency of the function)

Notice that this is the same image that was provided in the project description for the Salinas valley rotated. The function that produces this figure has been commented out for time complexity sake and since it only needs to be ran once to produce this output, thus it is unnecessary to re run when viewing this report along side the *main.m* file.

3   Feature selection/transformation

3.1   Selection

Representing multiple features with just one that correlates to a high degree would reduce the computational complexity of the data clustering to a significant degree. Simultaneously we have already removed a large set of the images because of the zero label pruning. Actually removing one feature removes more data from the sample than removing one data vector (since $150 \times 150 > 204$). All of this is to say that we should choose features to remove conservatively. The method of finding and removing features in previous exercises has been based on the cross correlation (2.3), where values less than a cutoff and over a second cutoff are removed. This wont be possible here since the cross correlation cannot be taken.

3.1.1 Principal Component Analysis

The way we will reduce dimensionality this time is via principaled component analysis. There is a builtin way of doing this provided with the project files. We will need to define the m variable input to the function which is the "number of the most significant principal components that are taken in to account". Before calling the function we will need to bring our data in to the form that the documentation says, that being "X: lxN matrix whose columns are the data vectors". In order to do this we will need to transform our $(M) \times (N) \times (L)$ matrix to a $((M \times N) \times L)$ matrix, essentially turning our pixel grid to a pixel line (this includes the labels if we don't plan on undoing the transformation afterwards).

To choose a value of m we should notice that the number of principal components is in the range of $[8, 204]$. If we pick a value $8 < a < 204$ that sufficiently explains the data (cutoff explain value) but might not be the optimal one then we can say that we do not need to consider the values $\in [a, 204]$ further. We will determine values of a to begin with and adjust accordingly until we end up with a qualitatively sufficient result.

We begin with a a of 50 and end up using a final value of 3. Actually this was apparent from the first trial since the output of the $pca_f un.m$ file is one that has "l-dimensional column vector, whose i^{th} element is the percentage of the total variance explained by the ith principal component", were we see that 3 components explain more than 99%.

After the PCA the data in the Y field looks as follows

./Images/PCA result.png

Figure 7: Overview of pixels with non zero label data

At this points we must mention that thougth the ground truch classes are 8 one could easily see 9 clusters in the image of the valey.

./Images/Salinas truth.png

Figure 8: Overview of pixels with non zero label data

It might be worth exploting the 9 clustering of the data, then conjoining two clsuters in the same ground truth class.

We mention this here since in the three dimensional representation result of the principal component analysis we can still see this 9 compact cluster form of the data.

3.2   Transformation

When we took the statistics of the features we end up with a $L \times 1$ dimensional vector with the value of each feature. In order to know if the data should be transformed we will look at the range of each feature tend compare it to the others. Particularly the maximum of all the feature maximums and the minimum of all the feature maximums, the maximum of all the feature minimums and the maximum of all the feature minimums.

Look to the corresponding function in the *main.m* file.

We see that the values range from (min(min)) −9 to (max(max)) 8374 with (max(min)) 2343 (min(max)) 35. We also see that the max and min median and mean values do not differ too greatly. The spatial resolution provided would change if the data was transformed, till now the dataset has been kept in the order (view the *main.m* file) of the original data to ensure compatibility when discussing results to the 8 ground truth classes. For these reasons we will not transform the data any further.

4 Selection of the clustering algorithm

Before moving to the criteria to select the clustering algorithms, let us notice that the *main.m* file has become crouded. We will take the final struct that we created and save it for importing without needing to use the whole pipeline made thus far. For this section the *tests.m* file will be considered.

4.1 Assumptions

There are two fields of assumptions to make, along the axis of accuracy and along the axis of time complexity.

We will reference this in the respective chapter but since the algorithms are initialized with the correct number of clusters we should expect that most algorithms will perform well on accuracy metrics (at least the algorithms that rely on input for the number of clusters).

We know the clusters form hard clusters, algorithms that leave room to interpret the bounds of the clusters will perform worse or equal to hard clusterings depending on our interpretation.

4.2 Project goals

Since there is an explicit statement that we will want to focus on the difference between hierarchical and CFO clustering algorithms we must choose a few from each category.

4.3 Selections

Actually, we are explicitly told what algorithms to choose.

Table 1: Algorithms to use

CFO Clustering	Hierarchical Clustering
K-means	Complete-link
Fuzzy c-means	WPGMC
Possibilistic c-means	Ward algorithms
Probabilistic	

5 Execution of the algorithms

5.1 Cluster Representative Initialization

The initialization of the cluster representatives occurs in the corresponding function in the *main.m* file. The method employed is to initialize them all within the bounds of the maximum and the minimum value along each feature randomly.

$$\theta_i \in [\min_i, \max_i]$$

5.2 Number of clusters

We actually already know the number of clusters that should be found from the project description.
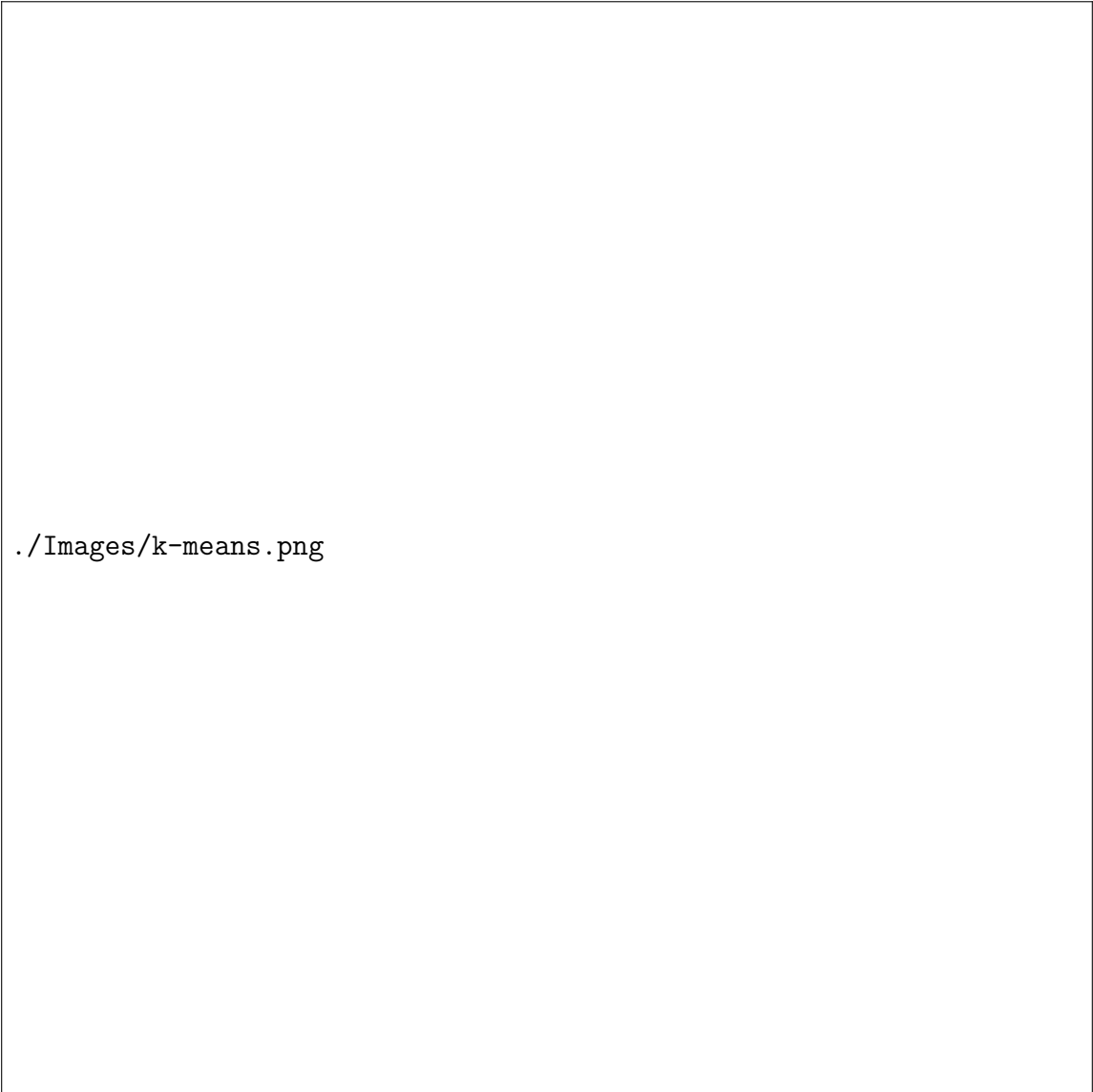
6 Results

6.1 Times

Figure 9: Result of k-means algorithm

Table 2: Algorithms to use

Clustering	Average of 100 trials
K-means	
Fuzzy c-means	
Possibilistic c-means	
Probabilistic	
Complete-link	
WPGMC	
Ward algorithms	

6.2   Accuracy

./Images/k-means.png

Figure 10: Result of k-means algorithm

./Images/Fuzzy c-means.png

Figure 11: Result of Fuzzy c-means algorithm

./Images/Possibilistic c-means.png

Figure 12: Result of Possibilistic c-means algorithm

./Images/Probabilistic c-means.png

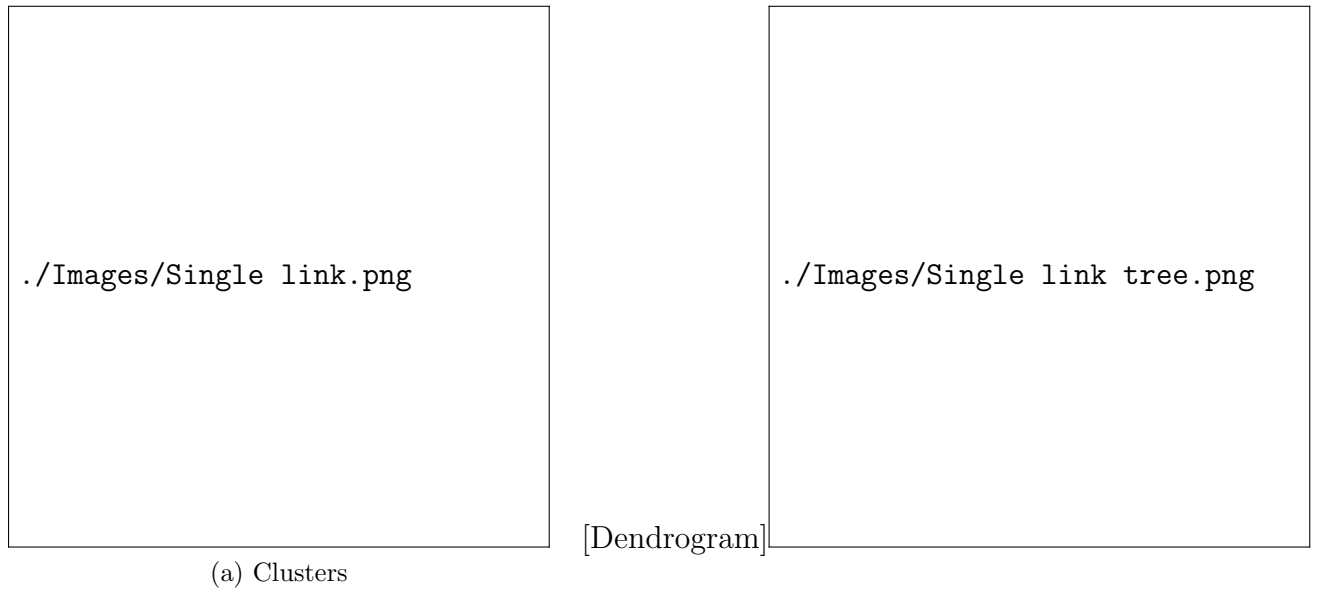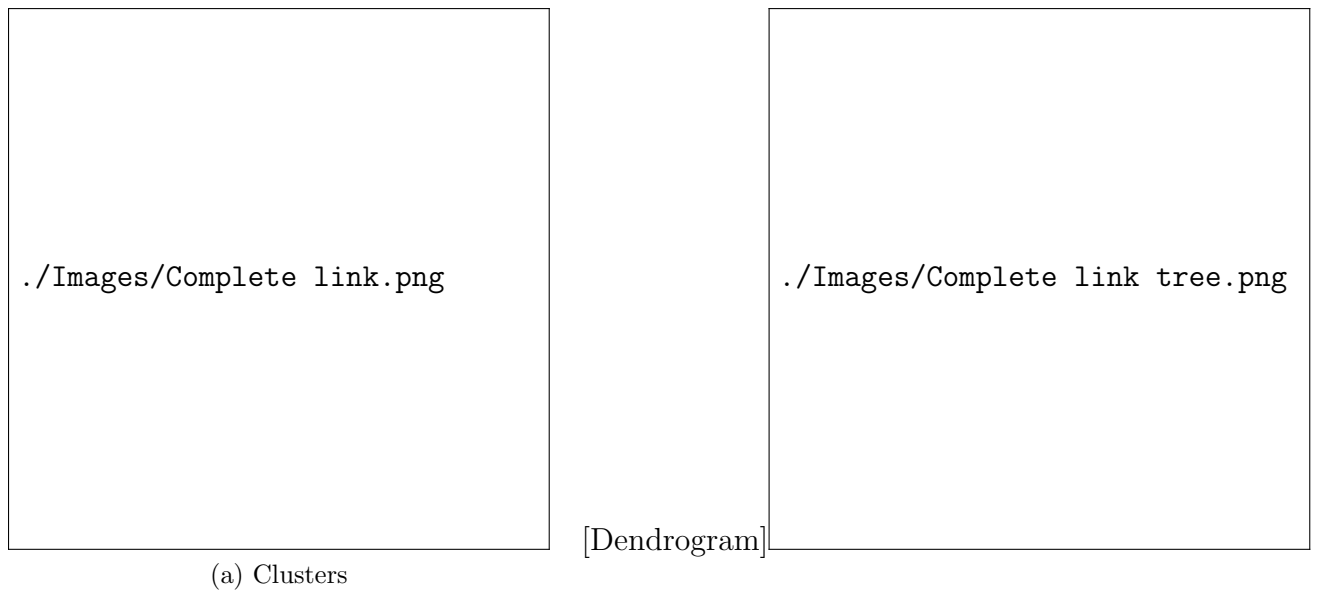Figure 13: Result of Probabilistic c-means algorithm

./Images/Single link.png

(a) Clusters

[Dendrogram] ./Images/Single link tree.png

Figure 14: Result of the Single link algorithm

./Images/Complete link.png

(a) Clusters

[Dendrogram] ./Images/Complete link tree.png

Figure 15: Result of the Complete link algorithm

(a) Clusters

[Dendrogram]

Figure 16: Result of WPGMC algorithm
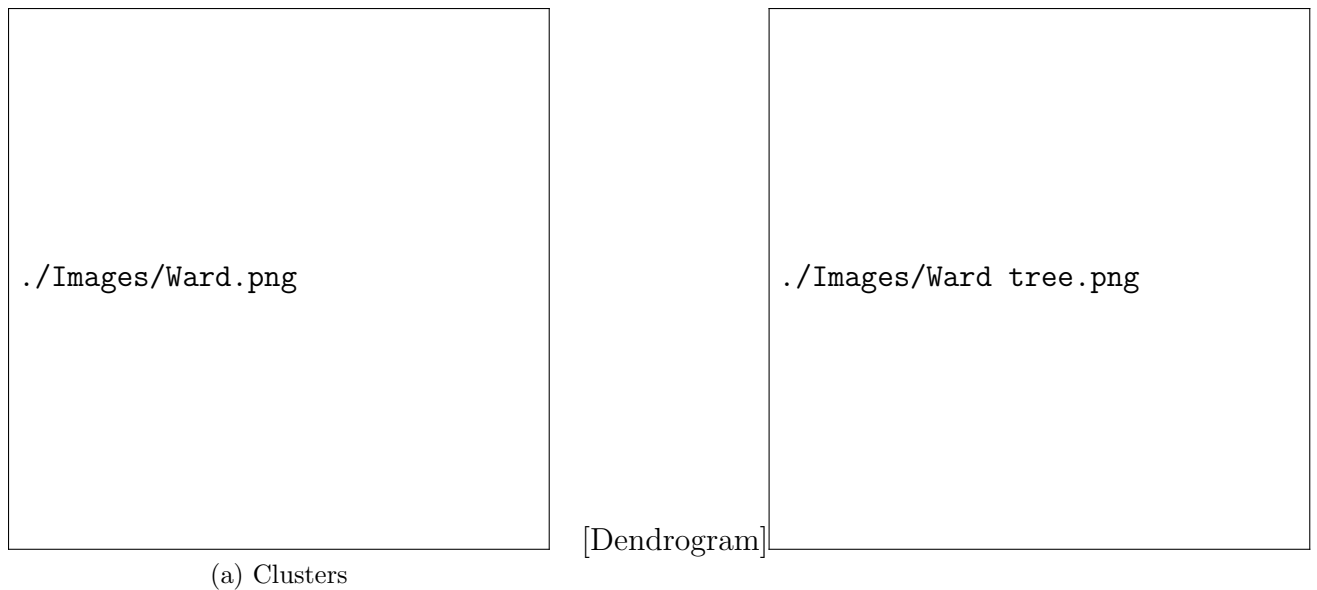

(a) Clusters

[Dendrogram]

Figure 17: Result of Ward algorithms

7   Citations

Sturges, H. A. (1926). The Choice of a Class Interval. Journal of the American Statistical Association, 21(153), 65–66. https://doi.org/10.1080/01621459.1926.10502161

Theodoridis, S., Pikrakis, A., Koutroumbas, K., Cavouras, D. (2010). Introduction to pattern recognition: a matlab approach. Academic Press.