




OCTOBER THE 19TH, 2015

CLUSTERING AND FACTOR ANALYSIS

INTRODUCTION TO MACHINE LEARNING

EROL KAZANCLI
MARÍA LEYVA
DANIEL SIMÓN



Contents

Introduction to the problem	3
The datasets	3
The clustering algorithms.....	3
K-means.....	3
Bisecting K-means	3
Fuzzy C Means	4
Measuring the results	4
Rand Index and Adjusted Rand Index	4
Davies Bouldin Index	4
F-Measure	4
Purity	5
Visualization	5
Selecting the most important attributes (Principal Components).....	5
3D graphs	5
Confusion matrix	6
Analysed datasets	7
Pyma_diabetes	7
K-means.....	7
Bisecting K-means	7
Fuzzy C-means	7
Conclusions	8
Iris.....	8
K-means.....	8
Bisecting K-means	9
Fuzzy C-means	9
Conclusion	10
Vehicle	11
K-means.....	11
Bisecting K-means	11
Fuzzy C-means.....	11
Conclusions	12
The code	12
Structure.....	12
How to execute it	15
Problems found and improvements proposed	15

Distribution of work 16

Bibliography 16

Introduction to the problem

We need to implement and analyse different clustering algorithms applied to different datasets. Also, we need to visualize and measure this algorithms performance.

In order to explain the problem, we are going to introduce how are the datasets we're working with, the algorithms used and the measure techniques we have applied.

The datasets

We are going to work with .arff (Attribute-Relation File Format) files (Weka, 2015).

For the clustering algorithms we are going to use, we are going to need datasets that only contain continuous attributes.

Also, we must normalize the values of these attributes, in order to treat all attributes equally.

The files we have selected for this (from the datasets provided by our professor at the Racó):

1. Pyma_diabetes.arff
2. Iris.arff
3. Vehicle.arff

The clustering algorithms

K-means

Our goal is to divide our data in K groups (clusters) so that every data point in each cluster is more similar to the other points in the same cluster than it is to the points in other clusters.

If we consider similarity as the Euclidean distance between two points, we must find k groups of points very similar (or close) among them.

To do this, we follow this steps:

1. Selection of seeds: one random seed is picked, and in proportion with K new seeds are added in a fashion that these new seeds are the furthest to the previous seeds. This continues until K seeds are picked. This process is done five times in order to find better seeds. They are taken as the centroid of the k clusters.
2. We assign each point in the dataset to the closest cluster. This means, we assign it to the cluster with the closest centroid.
3. We recalculate the cluster centroid, based in the points that belong to the cluster in this point.
4. We repeat steps 2 (now the closest centroid may be other) and 3 until any point changes membership.

For choosing the number of clusters, we are getting the real number of classes we have in the read data file. (Salamó, Lecture 2. Introduction to unsupervised learning and Cluster Analysis, 2015)

Bisecting K-means

This is a variation of K-Means, which follows the next steps: (Salamó, Lecture 2. Introduction to unsupervised learning and Cluster Analysis, 2015)

1. Choose a cluster (right now, we are choosing the largest one).
2. From that cluster, find two subclusters using traditional K-means.
3. Repeat 2 a given n times, and choose the one with biggest overall similarity.

4. Repeat 2 and 3 until k clusters are formed.

Fuzzy C Means

When we talk about fuzzy clustering we mean that there is no strict belonging to a cluster, but a membership value to one or more clusters. A data point can belong 80% to cluster 1 and 20% to cluster 2. This membership value is used to assign a point to a cluster. (Salamó, Lecture 2. Introduction to unsupervised learning and Cluster Analysis, 2015)

Measuring the results

Rand Index and Adjusted Rand Index

Both Rand Index and Adjusted Rand Index try to measure the degree of agreement between the actual cluster of the data and the results obtained by our algorithm. (Salamó, 2015)

Being:

- a= number of pairs of points which belong to the same cluster in P and in G
- b= number of pairs of points which belong to the same cluster in P and but to different clusters in G
- c= number of pairs of points which belong to different clusters in P and but to the same in G
- d= number of pairs of points which belong to different clusters both in P and in G

$$RI(P, G) = \frac{a + d}{a + b + c + d}$$

The Adjusted Rand Index is calculated using the Rand Index and the Expected Rand Index.

$$ARI = \frac{RI - E(RI)}{1 - E(RI)}$$

Davies Bouldin Index

According to the Matlab Documentation:

“The Davies-Bouldin criterion is based on a ratio of within-cluster and between-cluster distances. The Davies-Bouldin index is defined as

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \{D_{i,j}\}$$

where $D_{i,j}$ is the within-to-between cluster distance ratio for the i th and j th clusters. In mathematical terms,

$$D_{i,j} = \frac{(\bar{d}_i + \bar{d}_j)}{d_{i,j}}$$

is the average distance between each point in the i th cluster and the centroid of the i th cluster. \bar{d}_j is the average distance between each point in the j th cluster and the centroid of the j th cluster. $d_{i,j}$ is the Euclidean distance between the centroids of the i th and j th clusters.

The maximum value of $D_{i,j}$ represents the worst-case within-to-between cluster ratio for cluster i . The optimal clustering solution has the smallest Davies-Bouldin index value.” (MathWorks, 2015)

F-Measure

F-measure is a measure of accuracy. It is calculated in function to precision and recall. We have calculated F-measure for every cluster created in each dataset. (Wikipedia, 2015)

$$Fmeasure = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{tp}{tp + fp}, \quad recall = \frac{tp}{tp + fn}$$

- Tp(true positives): number of points that were classified to this cluster and actually belonged to this cluster
- Fp(false positives): number of points that were classified to this cluster but didn't really belong to it.
- Tn(true negatives): number of points that were not classified to this cluster and actually did not belong to this cluster.
- Fn(false negatives): number of points that were not classified to this cluster but actually did belong to this cluster.

Purity

The purity is defined as the ration between the dominant class in the cluster π_i and the size of cluster π_i , being n_{ij} the number of members of the class j in the cluster i . (Salamó, 2015)

$$Purity(\pi_i) = \frac{1}{n_i} \max_j(n_{ij})$$

Visualization

Selecting the most important attributes (Principal Components)

When we want to represent a dataset, we have to reduce the number of attributes, since it is impossible to represent in space a point with four or more dimensions or attributes.

Our intention is to represent our data in 3D graphs, so we have to reduce our attributes to three.

In order to do this, we have implemented the Principal Component Analysis algorithm.

The steps are the following:

1. Substract the mean from the data
2. Calculate the covariance matrix
3. Calculate eigenvectors and eigenvalues of the covariance matrix
4. Order eigenvectors by eigenvalue, highest to lowest, and get the biggest three (new feature vector).
5. From this, get the new dataset.

3D graphs

Once we have reduced the attributes of the dataset to three, we can plot it using matlab function *scatter3*, with the three principal components being X, Y and Z. If we give different colors to the different clusters, we will see something like the following picture.

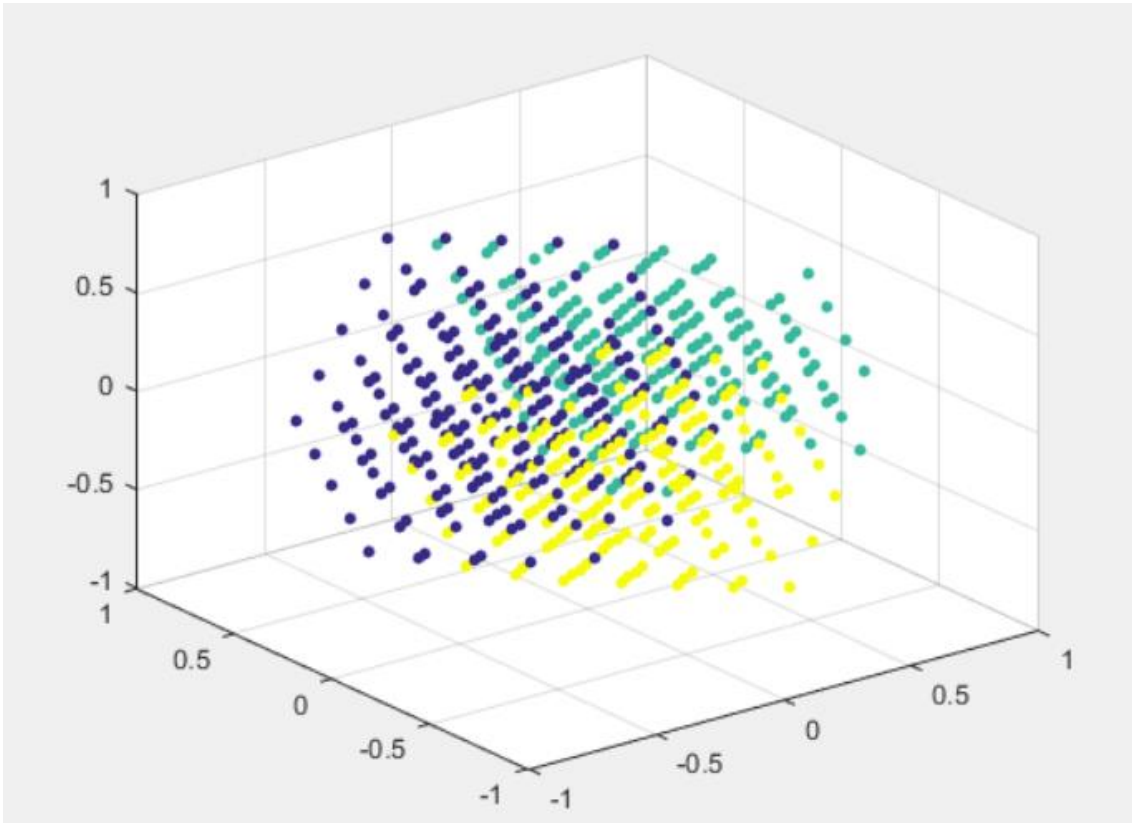


Figure 1: Example of 3d scattering

Confusion matrix

Another way of representing the data is a confusion matrix. This actually true positives, false positives, true negatives and false negatives as explained in F-Measure, but with all clusters at once.

	1	2	3
1	138	102	20
2	21	16	8
3	55	59	144

Figure 2: Confusion matrix example

	Cluster 1	Cluster 2	Cluster 3
Cluster 1	Tp(cluster 1)	Fp(cluster1,cluster2)	Fp(cluster1,cluster3)
Cluster 2	Fn(cluster 2, cluster 1)	Tp(cluster 2)	Fp(cluster2,cluster3)
Cluster 3	Fn(cluster 3, cluster 1)	Fn(cluster 3, cluster 2)	Tp(cluster 3)

Table 1: Confusion matrix explained

Translated for each cluster, it would be that:

	Cluster 1	Cluster 2	Cluster 3
Tp	138	16	144
Fp	102+20	102+8	20+8

Tn	16+8	138+144	16+144=
Fn	21+55	21+59	55+59

Table 2: Confusion matrix for each cluster

Analysed datasets

Pyma_diabetes

K-means

Rand index	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster1	F-measure cluster2	Purity cluster1	Purity cluster2
0.557	0.10238	0.70095	0.74877	0.31873	0.76	0.50373

Bisecting K-means

Rand index	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	Purity cluster 1	Purity cluster 2
0.557	0.10238	0.70095	0.74877	0.31873	0.76	0.50373

Fuzzy C-means

Rand index	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	Purity cluster 1	Purity cluster 2
0.558	0.11131	0.89023	0.73195	0.56733	0.6936	0.62388

Confusion Matrix - K Means			Confusion Matrix - Bisecting K Means			Confusion Matrix - Fuzzy C Means		
	1	2		1	2		1	2
1	380	120	1	380	120	1	351	149
2	135	133	2	135	133	2	108	160

Illustration 1: Pima_diabetes confusion table

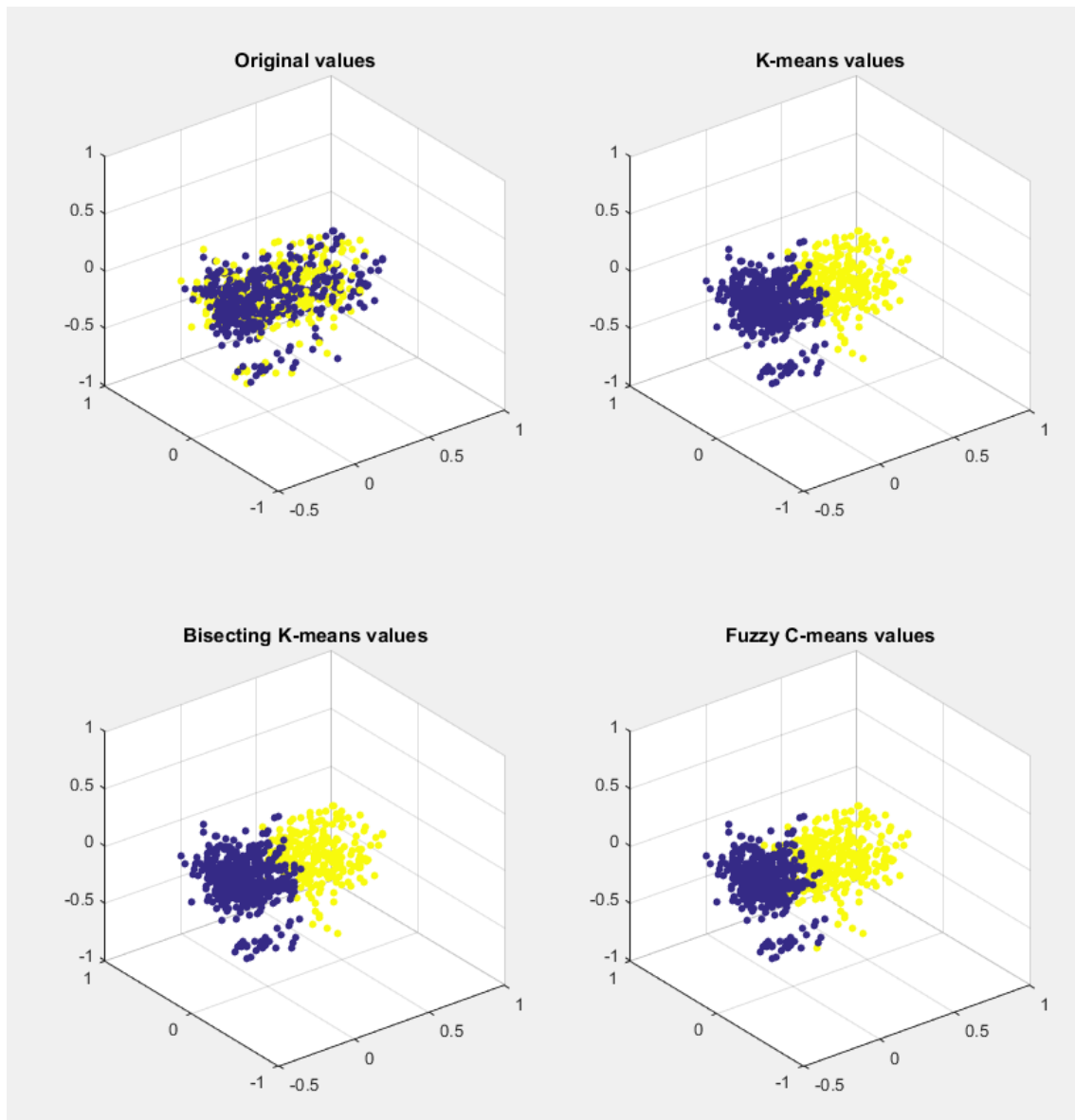


Figure 3: Pima_diabetes distribution

Conclusions

For this dataset we are not getting very good results. The Rand Index and Adjusted Rand Index are very low.

In the other hand, Davies-Bouldin Index is pretty high, so maybe the data it's not well distributed (another k value might be better). Actually, as we can see in our visualization, our distribution is more uniform than the real one.

Comparing the algorithms, K-means and bisecting K-means are getting the exact same results. Fuzzy C-Means, on the other side, gets better F-measures, and more uniform purities, but a higher Davies-Bouldin Index, so we cannot get a good conclusion.

Iris

K-means

Rand index	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster1	F-measure cluster2	F-measure cluster3	Purity cluster1	Purity cluster2	Purity cluster3
-------------------	----------------------------	-----------------------------	---------------------------	---------------------------	---------------------------	------------------------	------------------------	------------------------

0.87426	0.71325	0.77605	1	0.84074	0.8119	1	0.912	0.744
---------	---------	---------	---	---------	--------	---	-------	-------

Bisecting K-means

<i>Rand index</i>	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	F-measure cluster 3	Purity cluster 1	Purity cluster 2	Purity cluster 3
0.87311	0.71015	0.78113	1	0.83464	0.81481	0.884	0.768	0.884

Fuzzy C-means

<i>Rand index</i>	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	F-measure cluster 3	Purity cluster 1	Purity cluster 2	Purity cluster 3
0.87541	0.71493	0.77845	1	0.84112	0.8172	1	0.9	0.76

Confusion Matrix - K Means					Confusion Matrix - Bisecting K Means					Confusion Matrix - Fuzzy C Means				
	1	2	3			1	2	3			1	2	3	
1	50	0	0	0	1	50	0	0	0	1	50	0	0	0
2	0	47	3		2	0	47	3		2	0	45	5	
3	0	14	36		3	0	14	36		3	0	12	38	

Figure 4: Iris confusion matrix

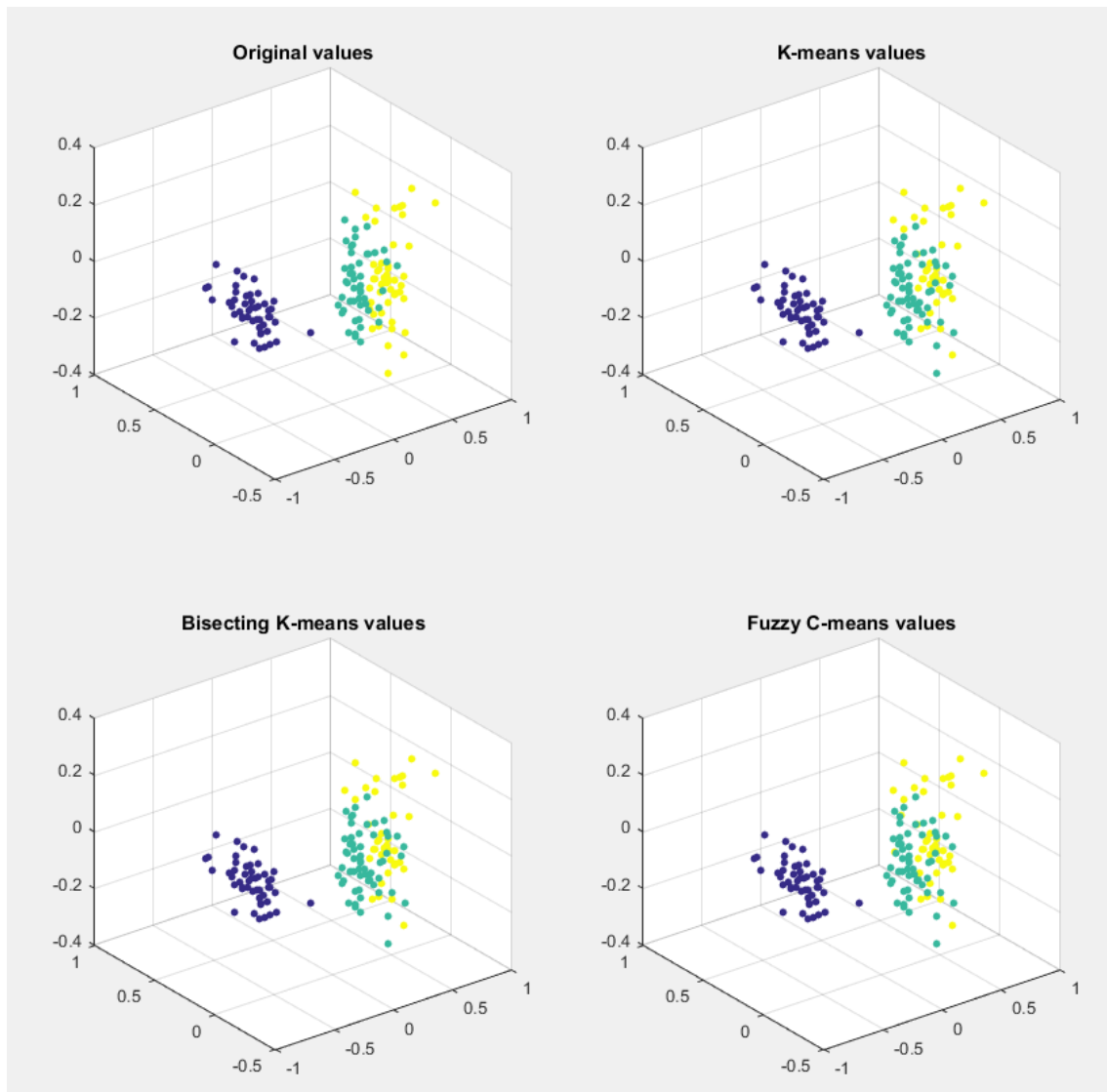


Figure 5: Iris distribution

Conclusion

With this dataset the three algorithms seem to work pretty well. We are getting the three Rand Indexes above 0.85 and the Adjusted Rand Indexes are bigger than 0.71, which is a very high measure.

With F-Measure and Purity, we're getting a straight 100% with the first cluster, and between 0.74 and 0.91 the others, so we are classifying correctly almost more than $\frac{3}{4}$ of the data.

Although, we are still getting a Davies-Bouldin Index above 0.77, so maybe there is another better value for K that would get better results.

Comparing the algorithms, K-means and fuzzy C-means get purity and F-measures more different for the second and third cluster. We are getting a better purity value for the second cluster, but worse for the third. They both get a perfect F-measure and Purity for the first group. The Bisecting K-means, gets a slightly worse purity for the first and the second cluster, and a slightly better one for the third cluster.

The conclusion is that the three algorithms get very good results, but very similar, so it is difficult to choose one.

Vehicle

K-means

<i>Rand index</i>	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	F-measure cluster 3	F-measure cluster 4	Purity cluster 1	Purity cluster 2	Purity cluster 3	Purity cluster 4
0.60441	0.082265	3.2886	0.47121	0.2403	0.30837	0.0070796	0.53208	0.49862	0.4945	0.62211

Bisecting K-means

<i>Rand index</i>	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	F-measure cluster 3	F-measure cluster 4	Purity cluster 1	Purity cluster 2	Purity cluster 3	Purity cluster 4
0.62252	0.087177	15.816	0.46503	0.22765	0.38937	0.010714	0.55189	0.53456	0.44404	0.49447

Fuzzy C-means

<i>Rand index</i>	Adjusted Rand Index	Davies-Bouldin Index	F-measure cluster 1	F-measure cluster 2	F-measure cluster 3	F-measure cluster 4	Purity cluster 1	Purity cluster 2	Purity cluster 3	Purity cluster 4
0.65208	0.077524	1.4857	0.45187	0.16739	0.2226	0.22857	0.49623	0.47373	0.42477	0.37789

Confusion Matrix - K Means					Confusion Matrix - Bisecting K Means					Confusion Matrix - Fuzzy C Means				
	1	2	3	4		1	2	3	4		1	2	3	4
1	118	94	0	0	1	117	55	40	0	1	105	36	35	36
2	117	100	0	0	2	116	57	44	0	2	103	40	36	38
3	29	159	28	2	3	57	63	98	0	3	46	93	45	34
4	0	193	0	6	4	0	98	99	2	4	0	67	70	62

Figure 6: Vehicle confusion matrix

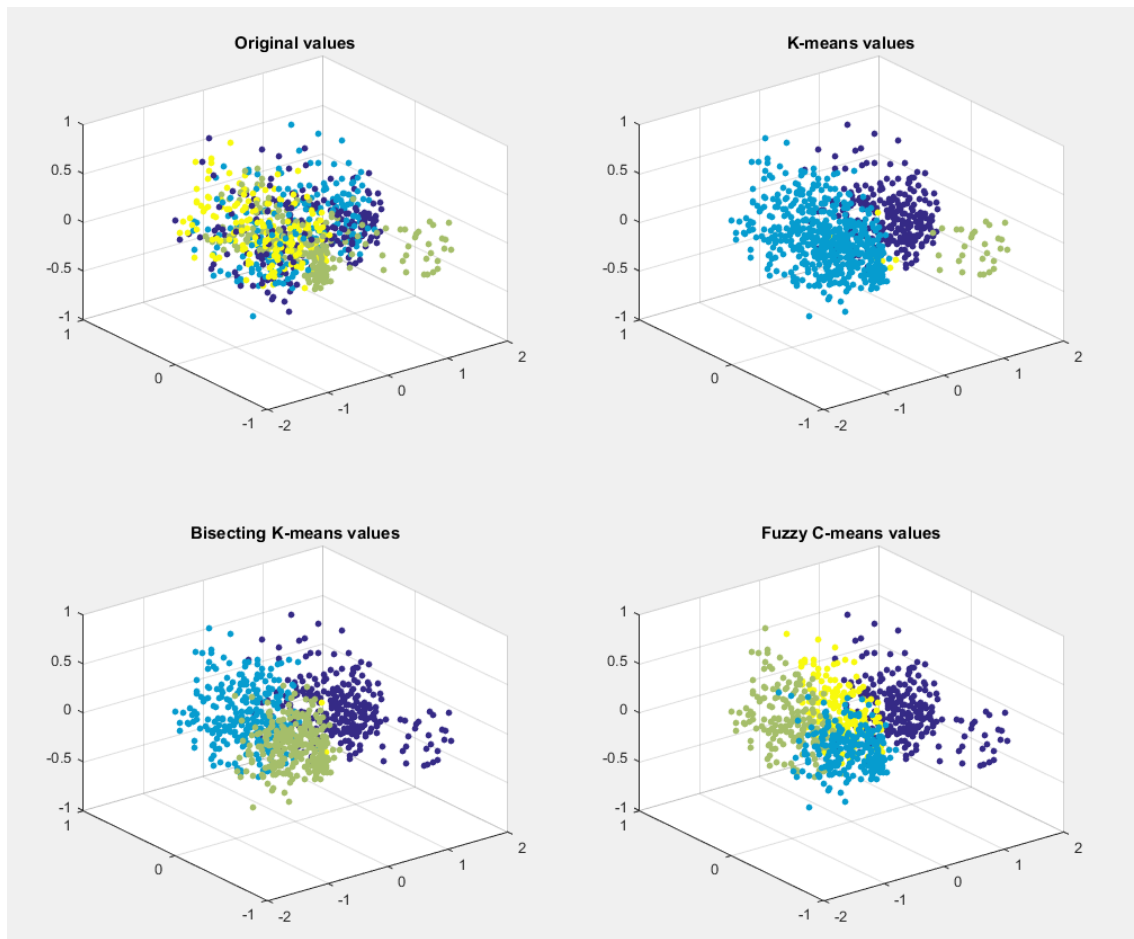


Figure 7: Vehicle distribution

Conclusions

This dataset obtains by far the worst result. It has a very high Davies-Bouldin Index and, though it is getting pretty high Rand Index and Adjusted Rand Index Values, it obtains very bad F-Measures and Purities. Again, the three methods work similarly, but this time, getting equally bad results.

The code

Structure

We have divided the code in different .m files, each one holding a function. The principal functions are the following:

Function	Input	Output	Description
<i>Arffparser</i>	Filename	Out cell	Given an .arff filename, it parses its data to a cell value
<i>Normalizer</i>	Data	Out matrix	Normalizes the data and transforms them into a matrix
<i>My_kmeans</i>	arr: normalised data numGroups: number of clusters	indexes: The number of cluster each data point belongs to after the algorithm ends. seeds: The centres of the clusters found after the algorithm ends, this is used	In this algorithm first seeds are randomly assigned. But since first seed values turned out to be important, we used an algorithm similar to kmeans++ to initialise our first seeds. After the first seeds are found the data points

		<p>later to calculate Davies-Bouldin index</p> <p>m: A cell containing all the clusters, which are arrays of data points</p> <p>WGV: Within group variance</p> <p>BGV: Between group variance</p>	<p>are assigned to the clusters according the minimum distance criterion. After the assignment the seeds are calculated again and the assignment is done according to the new seeds. This cycle continues until the memberships do not show any change or after a maximum number of iterations.</p>
<i>Bisecting_kmeans</i>	<p>params:</p> <p>arr: normalised data</p> <p>numGroups: number of clusters</p>	<p>indexes: The number of cluster each data point belongs to after the algorithm ends.</p> <p>seeds: The centres of the clusters found after the algorithm ends</p> <p>m: A cell containing all the clusters, which are arrays of data points</p>	<p>Implements bisecting k means according to Bisecting K-means section.</p> <p>It repeats step 2 n times</p>
<i>Fuzzycmeans</i>	<p>params:</p> <p>arr: normalised data</p> <p>numGroups: number of clusters</p>	<p>indexes: The number of cluster each data point belongs to after the algorithm ends.</p> <p>centroids: The centres of the clusters found after the algorithm ends, this is used later to calculate Davies-Bouldin index</p> <p>m: A cell containing all the clusters, which are arrays of data points</p> <p>WGV: Within group variance</p> <p>BGV: Between group variance</p>	<p>In this algorithm the initial membership values are assigned randomly to each data points. Initial centroids are found with these membership values. The distance for each data point to each centroid is calculated. Setting m= 1.5 new membership values and new centres are found. This cycle continues until a maximum number of iterations is reached or till the difference between two consecutive membership values do not change much, which means convergence to a local minimum.</p>
<i>Rand_index</i>	<p>indexes: The number of cluster each data point belongs to obtained by our algorithm.</p> <p>True_indexes: The number of cluster each data point belongs to according to the data file.</p>	<p>rand_index,</p> <p>adjusted_rand_index</p>	<p>Calculates rand index and adjusted rand index according to Rand Index and Adjusted Rand Index section.</p>
<i>F_measure</i>	<p>indexes: The number of cluster each data point</p>	<p>Fmeasure: matrix containing all the f-measures for all the clusters</p>	<p>Calculates f-measure according to F-Measure section.</p>

	belongs to obtained by our algorithm. True_indexes: The number of cluster each data point belongs to according to the data file.		
<i>Purity</i>	indexes: The number of cluster each data point belongs to obtained by our algorithm. True_indexes: The number of cluster each data point belongs to according to the data file.	Purity: matrix containing all the purities for all the clusters	Calculates purities according to Purity section.
<i>FindDaviesBouldinIndex</i>	m, centroids: seeds obtained by the algorithm numGroups	Db: Davies Bouldin index	Calculates Davies Bouldin Index according to Davies Bouldin Index section.
<i>Plot_results</i>	Filename: data file name, data_mat: normalized data matrix, true_indexes: indexes of clusters according to data file, indexes: indexes of clusters obtained by my_kmeans, fuzzy_indexes: indexes of clusters obtained by fuzzycmeans ,		Plots the results using 3D graphs and Confusion matrix. In order to plot it using 3D graphs, calculates Principal components with my_pca function.

	bisecting_indexes: : indexes of clusters obtained by bisecting_k means		
<i>My_pca</i>	Data: data matrix, P: number of attributes we want to get	feature_mat: feature matrix eigenvalues: eigenvalues obtained, result: data recalculated with n components, reconstructed_data: original data	Executes Principal Component Analysis according to Selecting the most important attributes (Principal Components) section.
<i>Clustering_analysis</i>	FileNames: cell array of strings containing filenames we want to analyse, numIterations: number of times we execute each algorithm. For this paper, we are doing it with 5.		Executes my_kmeans, bisecting_kmeans, fuzzycmeans numIterations times, and gets the mean of purities, f-measures, rand index, adjusted rand index and Davies-Bouldin index obtained in each iteration. It also plots the data of the last iteration using plot_results and writes the results of the validation functions into three files in the folder "results"

How to execute it

The best and direct way to execute the code is via the function clustering analysis.

For example, to obtain the data we've included in this paper, we ran this command:

```
clustering_analysis({'datasetsCBR/pima_diabetes.arff'; 'datasetsCBR/iris.arff'; 'datasetsCBR/vehicle.arff'}, 5)
```

Problems found and improvements proposed

The main problem we faced was the abandon of one of our members less than a week before the delivery of this work.

In the technical point of view, we have detected that some data may not be correctly classified, at least not in a very significant way with the attributes we have.

Also, we could have worked in the algorithms estimating the optimum value for k, in order to get better results, or trying many other clustering algorithms.

To conclude, if we have had more time, we could have analysed many more datasets, although we have chosen three quite representative ones, with three different number of clusters in order to cover as many scenarios as possible.

Distribution of work

TASK	AUTHOR
PARSER	Erol
NORMALIZATION	María
K-MEANS IMPLEMENTATION	Erol, Daniel
BISECTING K-MEANS IMPLEMENTATION	Erol, María
FUZZY C MEANS IMPLEMENTATION	Erol
PCA IMPLEMENTATION	María
F-MEASURE	María
RAND INDEX AND ADJUSTED RAND INDEX	María
PURITY	María
DAVIES-BOULDIN INDEX	Erol
VISUALIZATION: 3D PLOTS	María
VISUALIZATION: CONFUSION MATRIX	Erol
DOCUMENTATION	María (90%), Erol(10%)

Bibliography

Bishop, C. M. (2006). K-means Clustering. In *Pattern Recognition and Machine Learning* (pp. 424-427). Springer.

MathWorks. (2015, 10 25). *Davies Bouldin Evaluation*. Retrieved from <http://es.mathworks.com/help/stats/clustering.evaluation.daviesbouldinevaluation-class.html>

Salamó, M. (2015, 10). Lecture 2. Introduction to unsupervised learning and Cluster Analysis. *Introduction to Machine Learning*.

Salamó, M. (2015, 10). Lecture 3. Factor Analysis. *Introduction to Machine Learning*.

Weka. (2015, 10 25). *Weka - ARFF*. Retrieved from <https://weka.wikispaces.com/ARFF+%28book+version%29>

Wikipedia. (2015, 10 25). *Precision and recall*. Retrieved from https://en.wikipedia.org/wiki/Precision_and_recall