Results & Analysis
Version 1.0

Automatic prononciation evaluation

ECH-CHOUQI Rim

AALAOU Samia

BIOY Marianne

GUERY Samuel

YU Shaofeng

3IN

2015-2016

Sommaire

1 Features generation and first clustering

1.1 Mel features analysis

Results shown on the tables represents the rates of each category in each cluster (for example the rate of voiced phonemes in the cluster 1).

A first step was to use unsupervised k-means with 3 clusters :

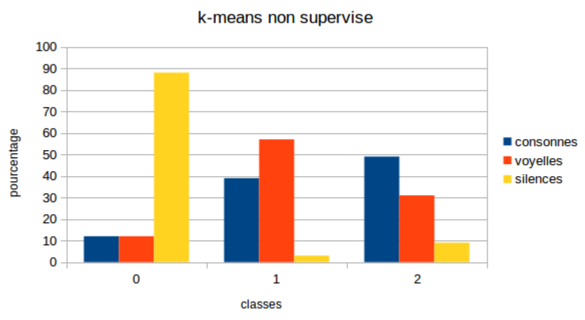


FIGURE 1.1 – unsupervised k-means on Mel parameters with 3 clusters

As nothing very meaningful shows up from this first clustering, we tried to use supervised k-means by previously initializing the centers of the clusters. To do so we choose as centers random windows containing a voiced/unvoiced/silence phoneme, these three centers allow us to initialize k-means.

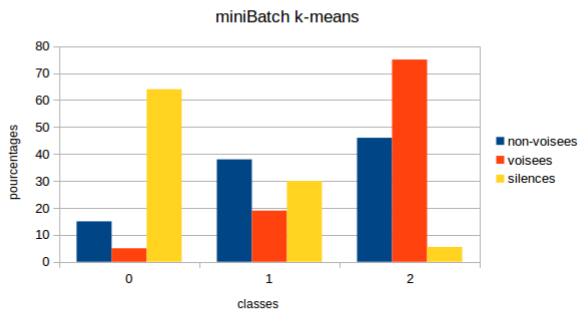


FIGURE 1.2 – Supervised K-means with three categories : voiced, unvoiced and silence

We notice that we have some better results : we can distinguish voiced phonemes in the third cluster.

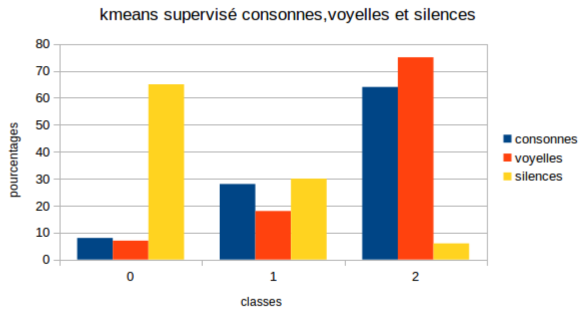


FIGURE 1.3 – Supervised K-means with three categories : consonants, vowels and silence

We notice that we can better distinguish silence in the first cluster. The results are surely better than those of a simple unsupervised K-means.

After this, we wanted to test what we can get if we proceed with 6 categories instead of 3. We choose the following categories : Implosives, nasals, fricatives, semiconsonants, vowels and silences :

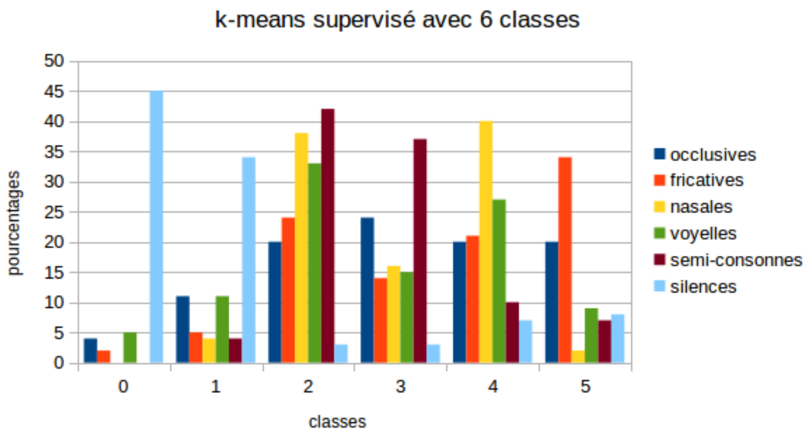


FIGURE 1.4 – Supervised K-means with 6 categories : nasals, fricatives...

We also tried the affinity propagation algorithm but it generated way too many clusters (above 80), as well as the spectral clustering algorithm that is too long in the execution.

In addition to displaying the rates of categories in the clusters, we tried to plot histograms that show the rate (or number) of each phoneme in each cluster :

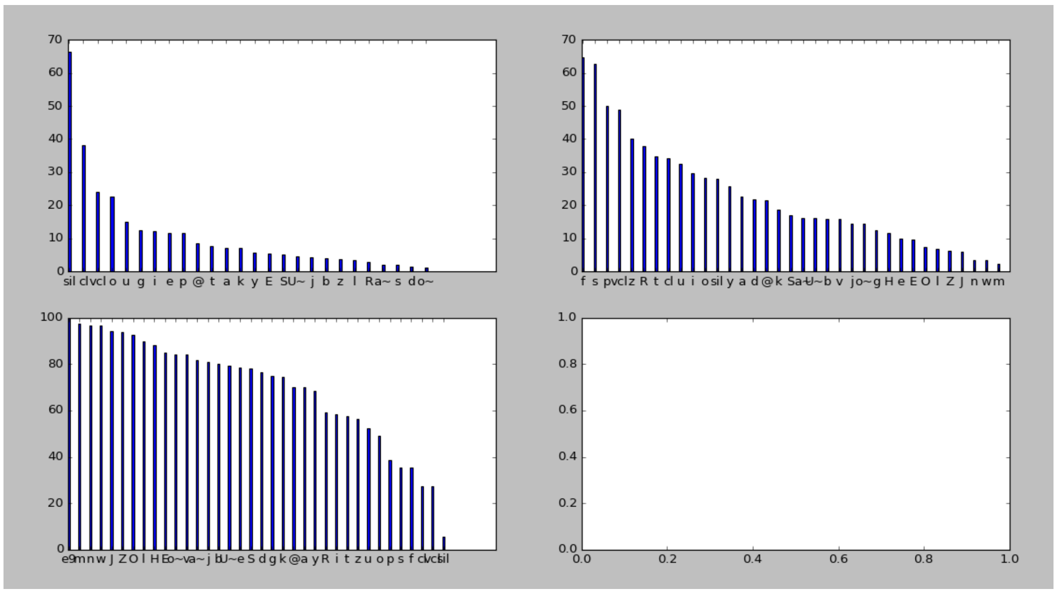


FIGURE 1.5 – k-means with 3 clusters histogram

1.2 Analysis with generated FBANK coefficients

1.2.1 FBANK generation

To generate this FBANK features, we use melspectrogram function from librosa. We first normalize the signal and then decompose it in the Mel scale thanks to the function. After that we apply a logarithm to the coefficients we got and we obtain this FBANK representation of the audio signal.

1.2.2 Analysis

Regarding the Fbank coefficients generated with python (on the file Bref80L4M01.wav), clustering with kmeans seems more efficient to separate voiced and unvoiced (initializing centers change nothing). Agglomerative clustering seems better to separate consonants and vowels. Though these separations are not very good.

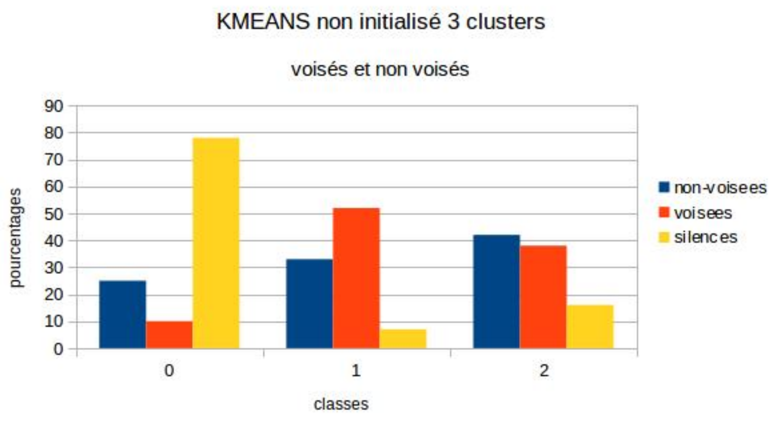


FIGURE 1.6 –

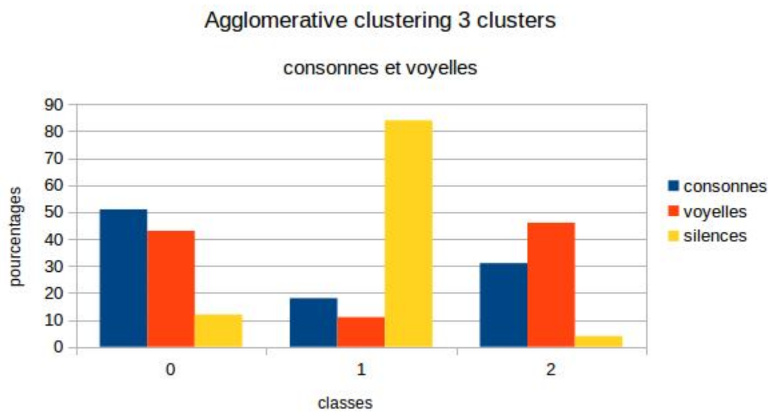
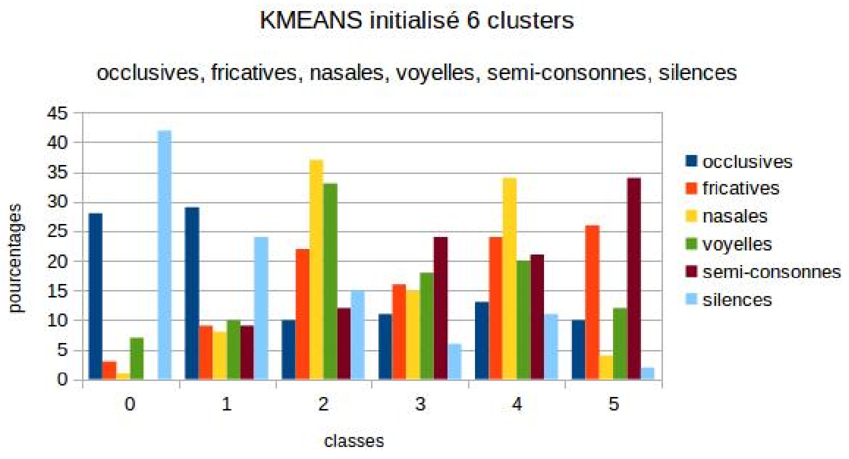
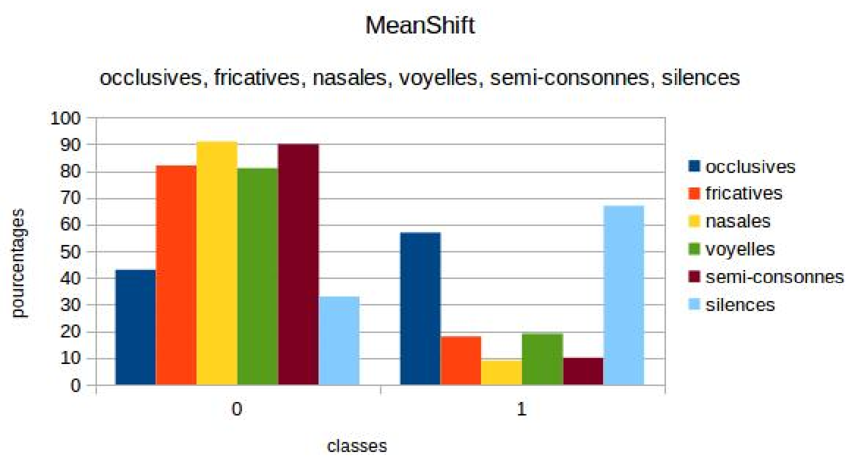


FIGURE 1.7 –

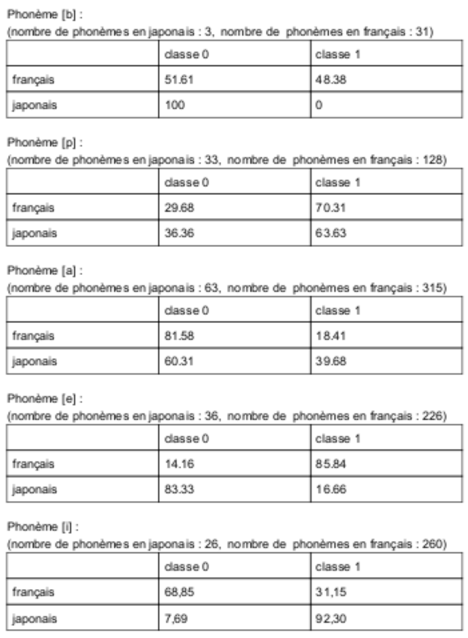
Considering 6 classes of phonemes (with k-means or an agglomerative clustering) ; we can notice a group between fricatives and half consonants, stop consonants and silences, nasals and vowels. Meanshift algorithm finds 2 classes and seems to associate silences with stop consonants, and the other classes together.





1.2.3 Work on Japanese corpus

Clustering (K-means) with only one phoneme, two speakers, one french and one Japanese. Here are clustering results on several phonemes :



Phonème [b] :
(nombre de phonèmes en japonais : 3, nombre de phonèmes en français : 31)

	classe 0	classe 1
français	51.61	48.38
japonais	100	0

Phonème [p] :
(nombre de phonèmes en japonais : 33, nombre de phonèmes en français : 128)

	classe 0	classe 1
français	29.68	70.31
japonais	36.36	63.63

Phonème [a] :
(nombre de phonèmes en japonais : 63, nombre de phonèmes en français : 315)

	classe 0	classe 1
français	81.58	18.41
japonais	60.31	39.68

Phonème [e] :
(nombre de phonèmes en japonais : 36, nombre de phonèmes en français : 226)

	classe 0	classe 1
français	14.16	85.84
japonais	83.33	16.66

Phonème [i] :
(nombre de phonèmes en japonais : 26, nombre de phonèmes en français : 260)

	classe 0	classe 1
français	68,85	31,15
japonais	7,69	92,30

FIGURE 1.8 –

For some phonemes (like [e] or [i]), we can distinguish the phonemes pronounced by the french speaker from those pronounced by the Japanese speaker, which is very interesting.
For others ([b] for instance), Japanese phonemes gather but with an important part of french phonemes (but in this case we don't have much Japanese data). Finally, with some kind of phonemes (like [a]), one cluster is much bigger than the other and results are not interesting at all (it is the case for silences, which seems logical).

1.2.4 Coefficients histograms

We drew histograms along each frequency bands of parameters matrix. However, regarding all transformations we made, we didn't get any distinctions between voiced and unvoiced phonemes (only silences distinguish better most of the time). Histograms we got more or less looks like this :

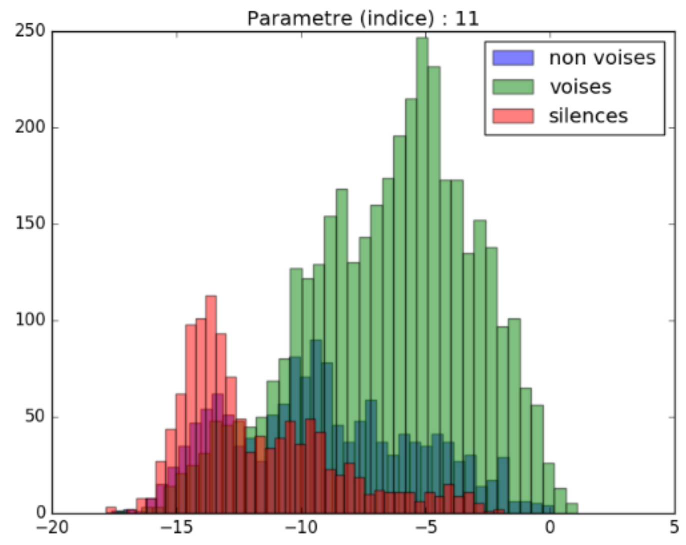


FIGURE 1.9 –

1.2.5 Use of more data

By using two files (Bref80L4M02 added), that is to say with more data, results are a bit better : the gap between voiced and unvoiced increase in the voiced class and in the unvoiced class. Using more than 2 audio files was not possible without more computational power, but for a start those two files seem to show the tendency which is that results should be improved if we use more data.

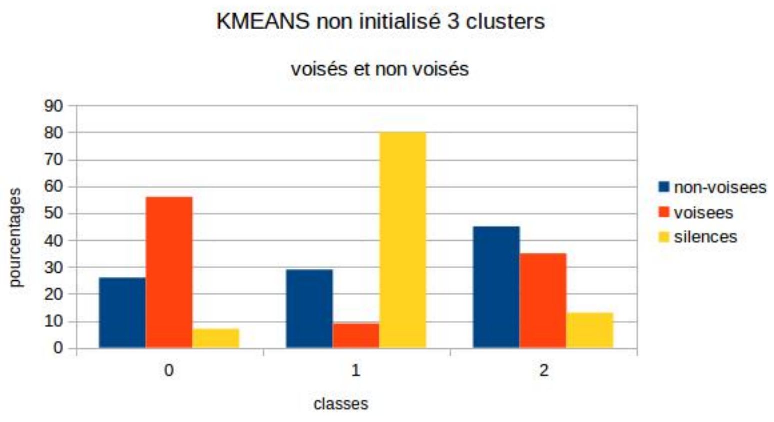


FIGURE 1.10 –

1.3 Given FBANK .mat Analysis

1.3.1 Unsupervised kmeans with 2 or 3 classes

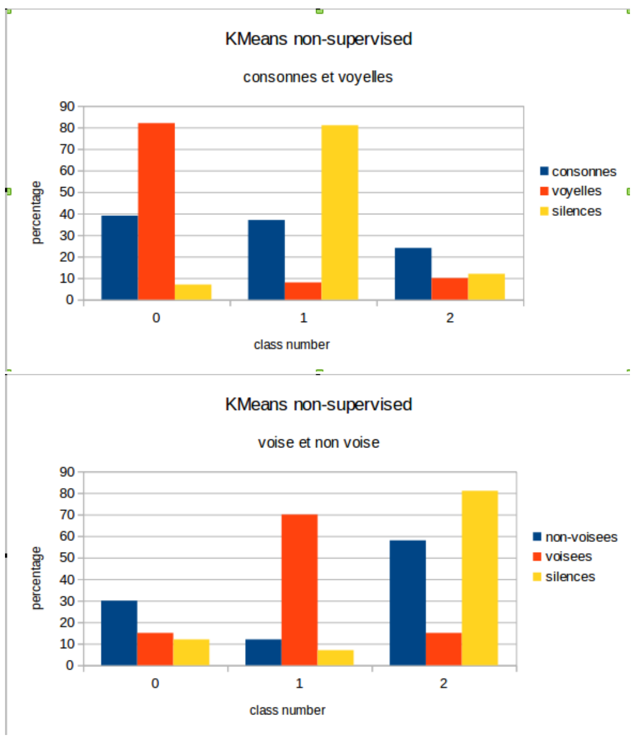


FIGURE 1.11 – Unsupervised K-means, 3 classes

With this algorithm, we classify pretty well the silence and the voiced sounds (or the vowels as well)

1.3.2 Unsupervised kmeans with 6 classes

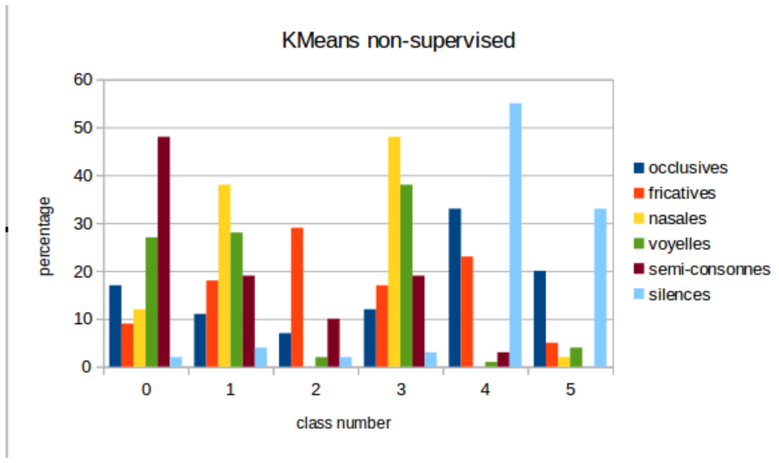


Figure 1.12 – Unsupervised K-means, 6 classes

For 6 clusters, we can group the semi-consonants, the silence and the nasals.

To find a better K, we can try Gap statistic algorithm. In the latter, we compute two values which are gap and s. We should choose the smallest value of K that would satisfy gap(K)> gap(K+1)-s(K+1)

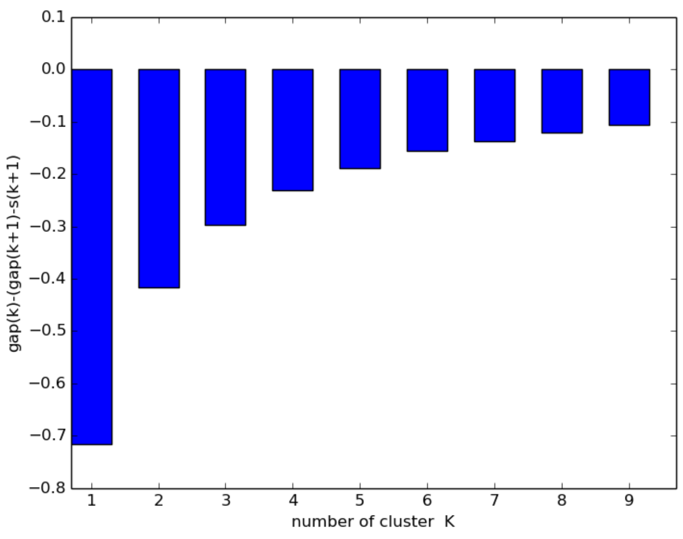


Figure 1.13 – Gap Statistic

In our case, we didn't find a small enough K that satisfies gap(K) >(gap(K+1)-s(K)).

1.4 Wavelets analysis

1.4.1 K-MEANS algorithm

The two figure below represent the results for the K-means algorithm, with the Morlet wavelet transform of the signal. In each figure, we show the bar chart for two matrices :

- The first is the concatenation of the lines corresponding to 20ms time windows, with a 10ms overlapping
- The second is made of the means of those lines

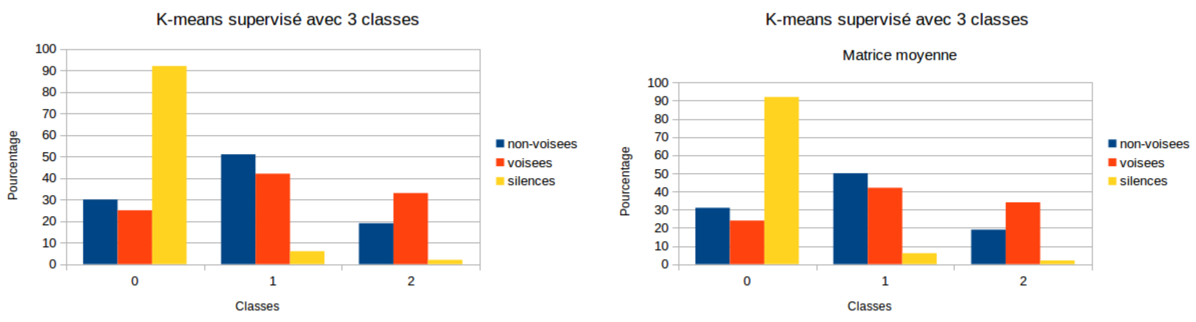


FIGURE 1.14 – K-means 3 classes initialized

We notice that we obtain the same kind of results for the two types of matrices. We will now focus on the the "mean" matrix, considering its reduced dimension.

1.4.2 K-means initialized

Other types of wavelet transforms (Paul and dog) have also been computed on the "mean" matrix. The figure below show the results of these two transforms for k-means initialized :

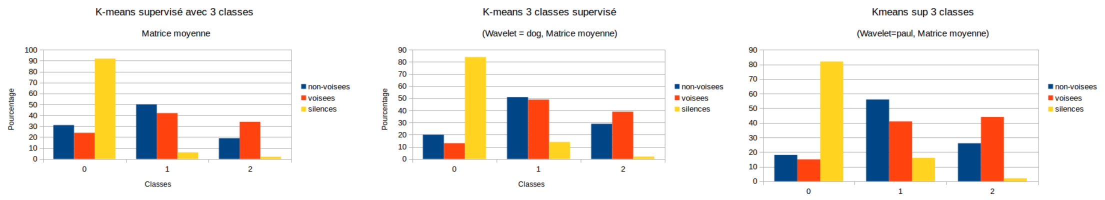


FIGURE 1.15 – wavelet mother Morlet / wavelet mother dog / wavelet mother Paul

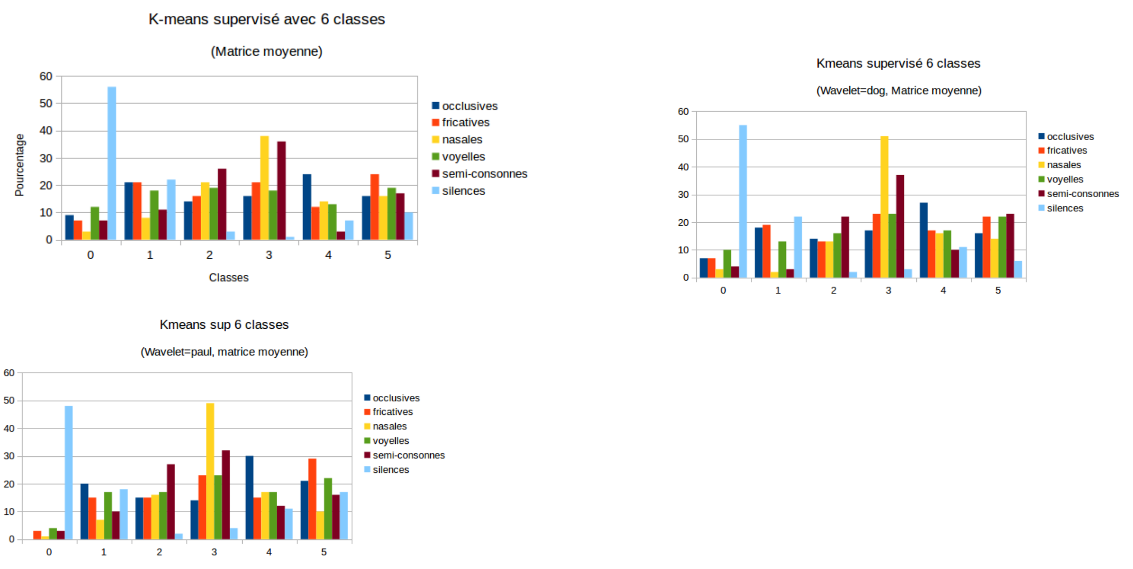


FIGURE 1.16 – wavelet mother Morlet / wavelet mother dog / wavelet mother Paul

he resulting classes are quite mixed up, onnly the 'silence' phones really stands alone. However, we can notice that there is a slightly better classification when using the wavelet mother Paul, it clusterizes better the voiced phones.

1.4.3 K-means uninitialized

In using the Kmeans uninitialized algorithm (no centers initialization), we obtain the following results :

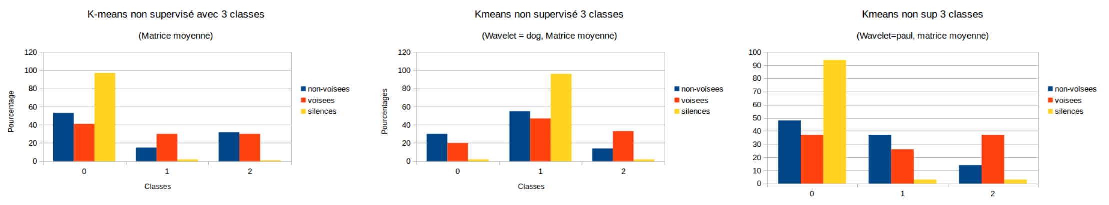


FIGURE 1.17 – wavelet mother Morlet / wavelet mother dog / wavelet mother Paul

As for the initialized algorithm, there is no distinct separation of the classes. The wavelet mother of type Paul gives also better classification results.

1.4.4 Agglomerative Clustering algorithm

the *Agglomerative Clustering* algorithm a also been tested on the wavelet transform matrix. The figures below represent the rates of the different sounds categories, in the different classes of the clustering :

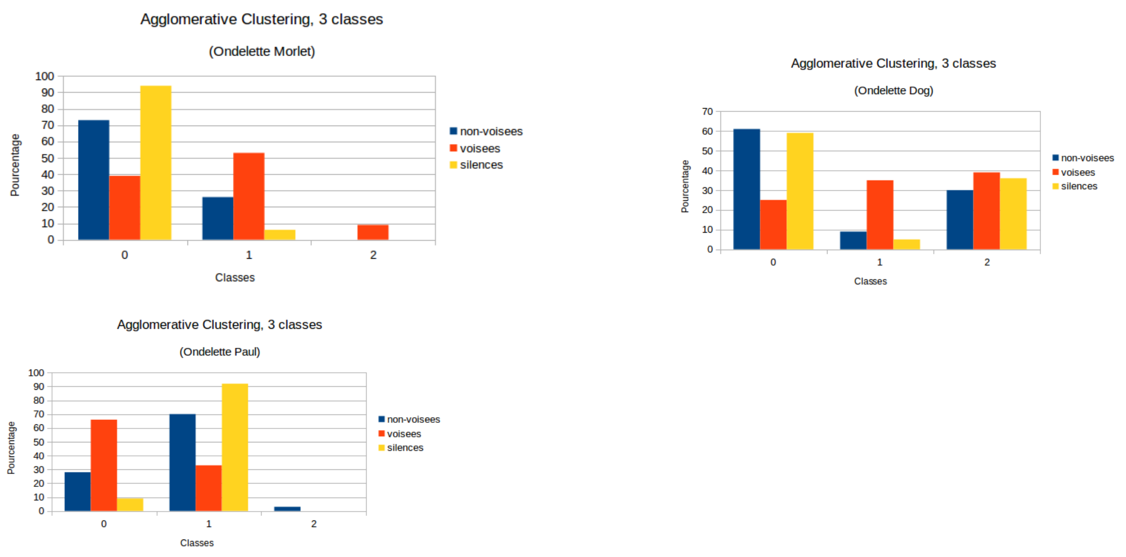


FIGURE 1.18 – Ondelette mère Morlet / Ondelette mère dog / Ondelette mère Paul

For the 3 classes, the voiced sounds are grouped in a single class using the wavelett mother Paul.

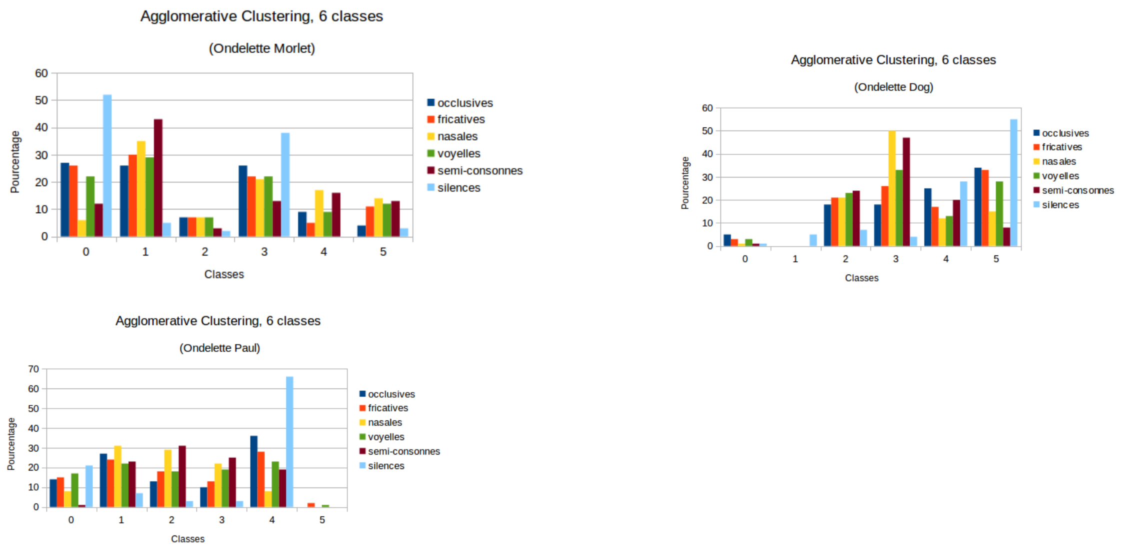


FIGURE 1.19 – Ondelette mère Morlet / Ondelette mère dog / Ondelette mère Paul

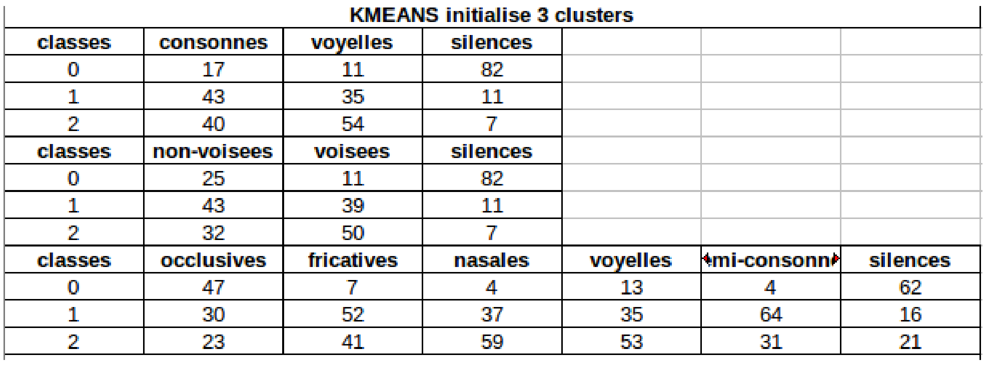
For the 6 classes clustering, if tou consider the Dog wavelett, the semiconsonnants and the nasals in one class, and the silences in an other. The Paul wavelet groups the silences and the occlusives in the same class.

1.5 Mel cepstral coefficients Analysis

Several methods of clustering have been tried : kmeans with centers initialized or not, agglomerative clustering and the meanShift algorithms. the first two clusterings have been established for 3 and 6 clusters, and the meanShift has produced 3 classes. We check the efficiency of the clusterizations by making appearing the rate of consonnants/vowels/silences, of phones voices/unvoices/silences and of phones categories (nasals, fricativs...) in each cluster. This all analysis has been done on the 13 and 40 coefficients Mel transform matrix. The results shown here are for the 13 coefficients matrix.

In first reading, we can sse that the kmeans initialized and uninitialized are almost the same, and this for the 3 or 6 clusters classification. The only remarkable result is that for 6 clusters, the semiconsonnnants are a little better recognized with the initialized algorithm.

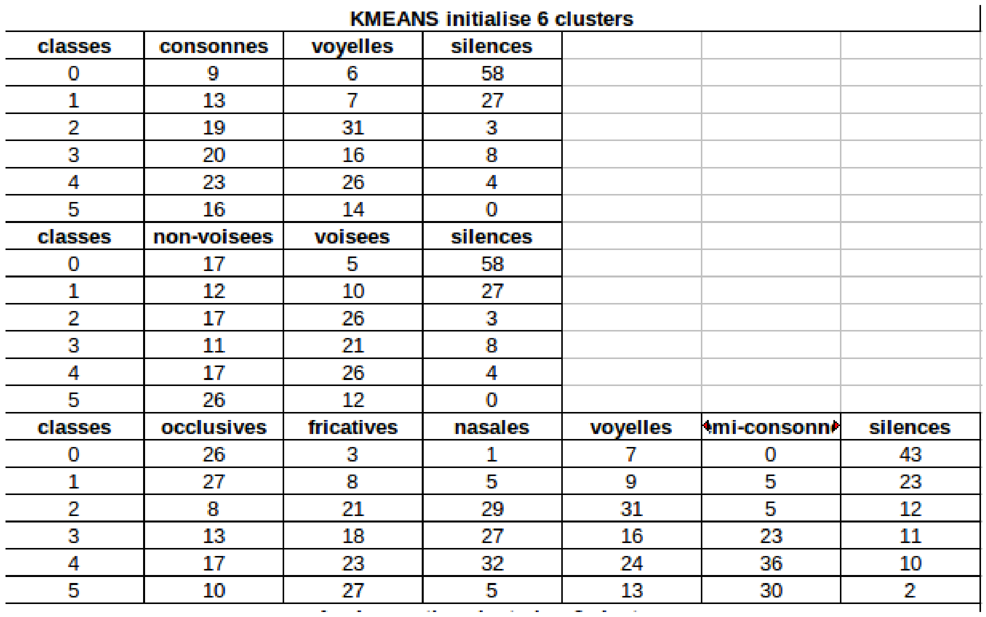
This is what we get for these two clusterings (those are the initialized results) :



KMEANS initialise 3 clusters

classes	consonnes	voyelles	silences			
0	17	11	82			
1	43	35	11			
2	40	54	7			
classes	non-voisees	voisees	silences			
0	25	11	82			
1	43	39	11			
2	32	50	7			
classes	occlusives	fricatives	nasales	voyelles	mi-consonn	silences
0	47	7	4	13	4	62
1	30	52	37	35	64	16
2	23	41	59	53	31	21

Figure 1.20 – kmeans for 3 clusters



KMEANS initialise 6 clusters

classes	consonnes	voyelles	silences			
0	9	6	58			
1	13	7	27			
2	19	31	3			
3	20	16	8			
4	23	26	4			
5	16	14	0			
classes	non-voisees	voisees	silences			
0	17	5	58			
1	12	10	27			
2	17	26	3			
3	11	21	8			
4	17	26	4			
5	26	12	0			
classes	occlusives	fricatives	nasales	voyelles	mi-consonn	silences
0	26	3	1	7	0	43
1	27	8	5	9	5	23
2	8	21	29	31	5	12
3	13	18	27	16	23	11
4	17	23	32	24	36	10
5	10	27	5	13	30	2

FIGURE 1.21 – kmeans for 6 clusters

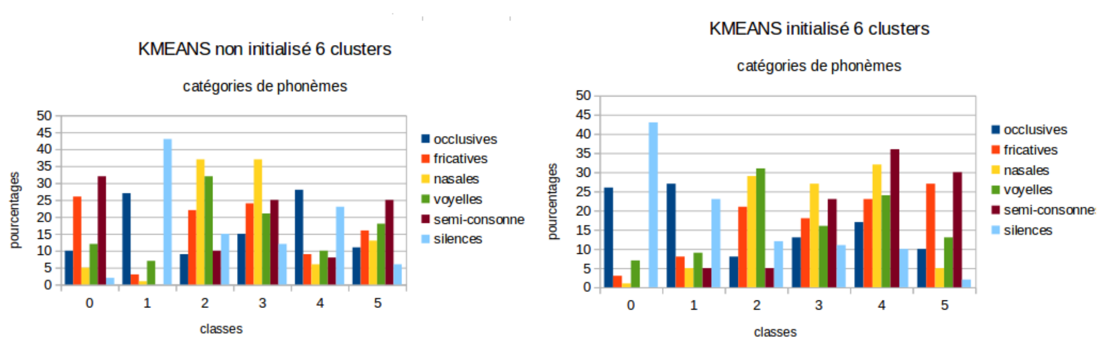
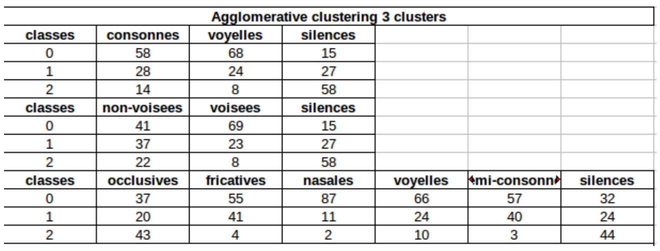


FIGURE 1.22 – categories of phones spotted out by the supervized and unsupervized kmeans

What emerges from these figure is that the voiced phones are well grouped together, and the vowels as well, for the initialized kmeans (more than 60% in one class, which is a better result than de silences). Besides, the occlusivs tend to be classified with the silences, and the semiconsonnants to be well recognized. The fricativs, the nasal consonnants and the vowels are grouped together.

The results of the agglomerative clustering are a lot better : for three classes, the vowels and the voiced phones are very well recognized, better than the silences. The nasal consonnants are grouped with the other phones, but are well recognized nontheless (they are grouped to more than 80% in the same class). The occlusivs are still seen as close to the silences :



Agglomerative clustering 3 clusters						
classes	consonnes	voyelles	silences			
0	58	68	15			
1	28	24	27			
2	14	8	58			
classes	non-voisees	voisees	silences			
0	41	69	15			
1	37	23	27			
2	22	8	58			
classes	occlusives	fricatives	nasales	voyelles	mi-consonn	silences
0	37	55	87	66	57	32
1	20	41	11	24	40	24
2	43	4	2	10	3	44

FIGURE 1.23 – results for 3 clusters of the agglomerative clustering

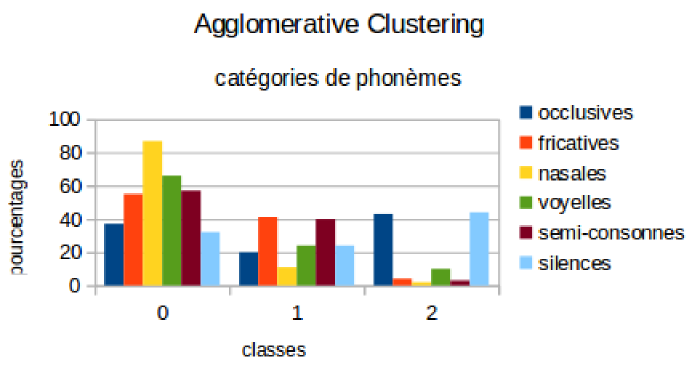


FIGURE 1.24 – categories of phones recognized by the agglomerative clustering

Finally, we can see that the MeanShift algorithm only identifies two classes, where the silences are opposed to the other phones. However, it can be noticed that the occlusiv phones are still strongly associated with the silences, and the nasal consonnants are better spotted out than the others :

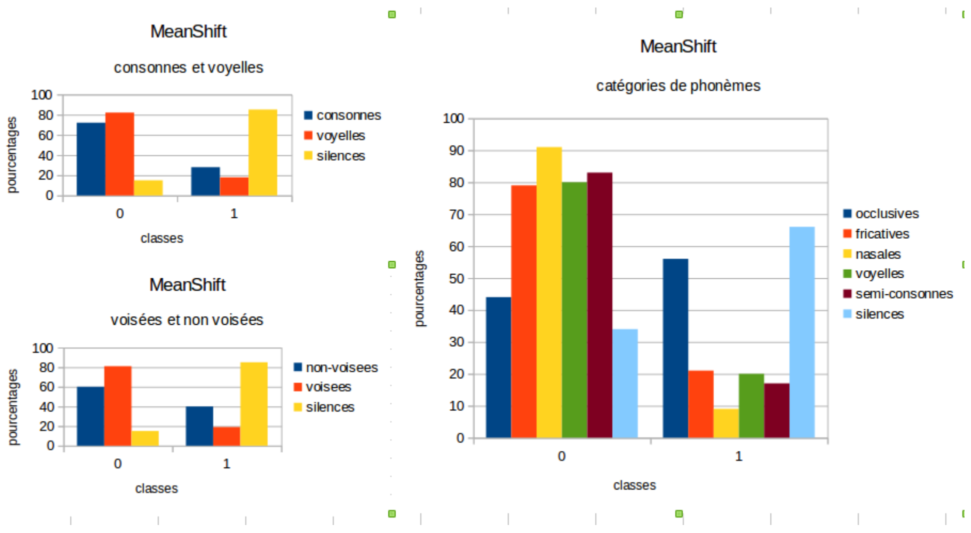


FIGURE 1.25 – bar chart of the results for MeanShift

There is the result of the clustering for each phone (percentage of each phone placed in each cluster for 6 clusters by the kmeans initialized an uninitialized, the agglomerative clustering, and the MeanShift algorithms) :

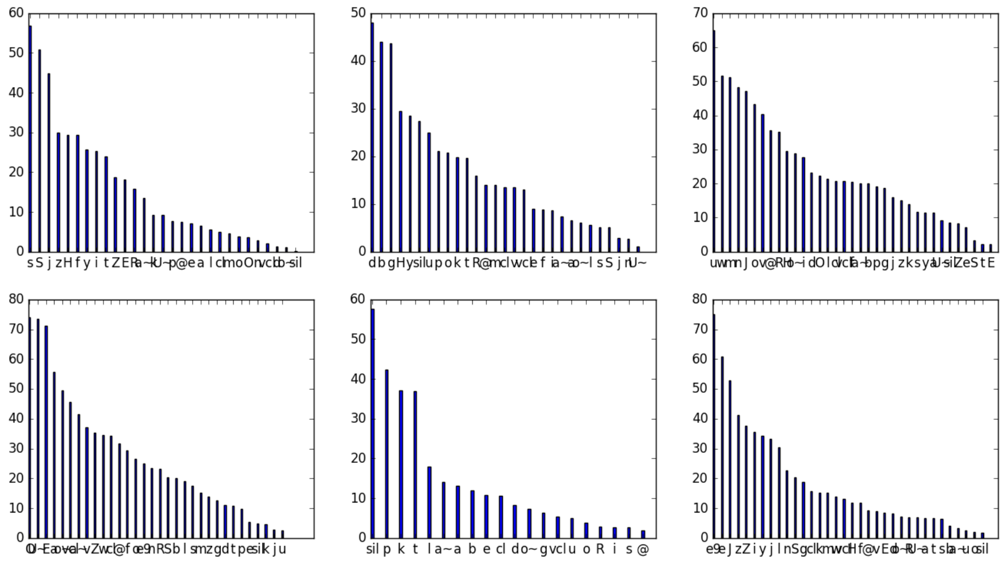


FIGURE 1.26 – kmeans uninitialized for 6 clusters

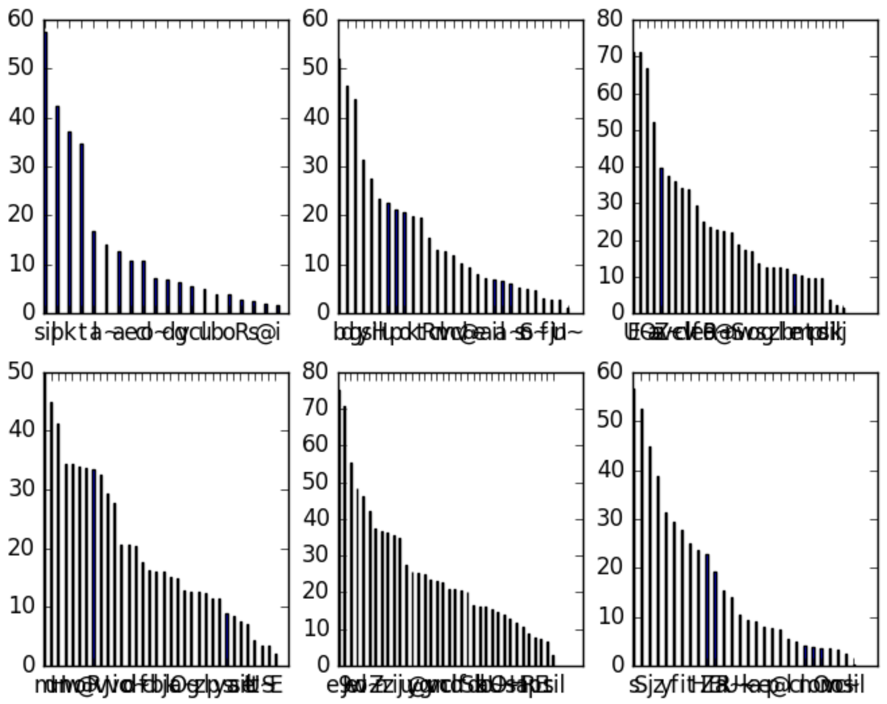


FIGURE 1.27 – kmeans initialized for 6 clusters

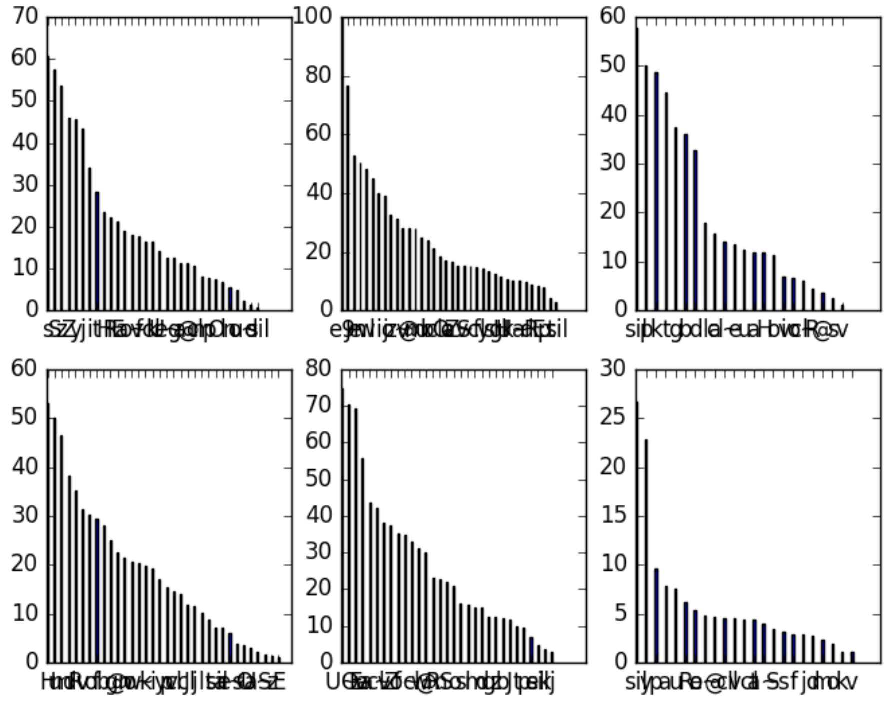


FIGURE 1.28 – agglomerative clustering for 6 clusters

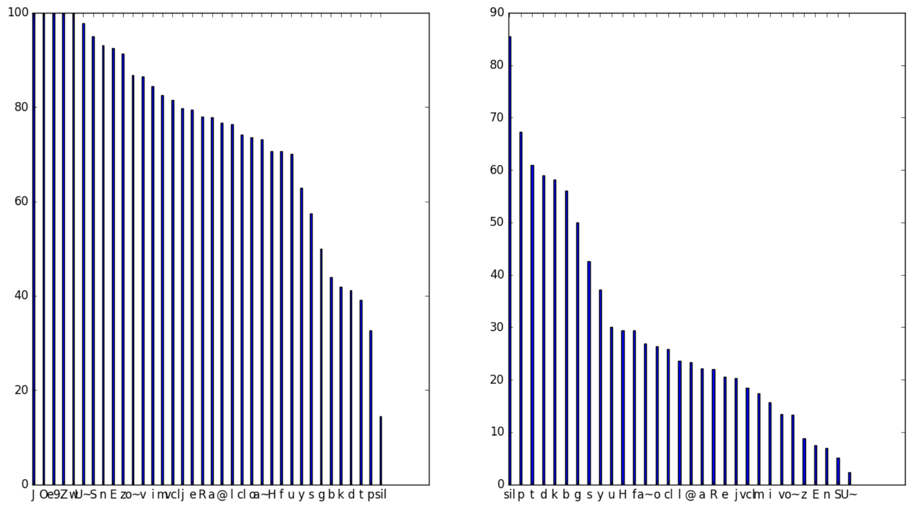


FIGURE 1.29 – meanShift

We can observe that some fricatives ([s]) and occlusives ([b] and [d]) are better discriminated (clusters 1 and 2), and that phones draw levels in each cluster, rather than just slopes. The 6 classes clustering seems to give slightly better results when you consider phones separately.

Conclusion

For the Mel coefficients, the result that is to be highlighted is the distinction of the voiced phones at 75% for the initialized minibatchKmeans algorithm.

What comes out of the mfcc parameters clustering are the agglomerative clustering results : it recognizes quite well the vowels and the voiced phones. What's more, for almost every clustering algorithm, a great part of the occlusivs are often seen as silences, for 2,3 or 6 classes. The nasal consonants are always well regrouped, but classed with the fricativs and the vowels.

Using the FBank coefficients generated by Matlab, we can group the silence and the voiced sounds (or the vowels) when using the K-means algorithm.

Regarding the wavelet transform, the use of the Paul mother wavelet in the supervised K-means algorithm gives some quite good results : For 3 clusters, we can group the silence and the non-voiced sounds in two separate classes, and for 6 clusters, we can separate the silence and the nasals as well. For the agglomerative clustering, it is the Morlet mother wavelet that gives the best classification for 6 clusters : we can separate the silence, the nasals and the sem-consonants. For 3 clusters, the Paul mother wavelet is the only one to be able to separate the voiced sounds in a separate class.

2 Activation maps Clustering

2.1 Activation maps Clustering Implementation

2.1.1 Data

The layers of the system are the following :

— conv1 : First convolution layer,

— conv2 : Second convolution layer,

— mp2 : Max-polling layer,

— dense1 : Dense and last layer.

For each layer, two sets of activation maps (FR and JA) are given, each set has 4 categories (correct_OK, incorrect_OK, correct_pasOK and incorrect_pasOK), each category has 70 examples : 35 for each phoneme, and each example has 256 maps for conv2 and 128 for the other layers.

2.1.2 Main idea

As the goal is to compare activation maps of different examples, according to various criteria (described below), we looked for a way to gather all the examples of an activation map to one manipulable matrix.

Therefore, we defined a function called pretraitement_matrice that allows us to have a representation of the activation maps as a (number of maps per example)*(number of examples)*(dimensions of a map) shaped tensor. In this function, the activation maps are vectored, to gather them in a matrix able to be an input of the clustering algorithms.
The parameters of this function are the kind of examples we want to consider (the dictionary, the phonemes...), which makes it a very general and reusable function.

The next idea is to perform the clustering on each matrix of the tenser, the first matrix for example is the matrix that gathers the first activation map of all the examples. The first line of this matrix, for example, is the vectored first map of the first example.

2.1.3 Clustering

There are 5 types of clustering to be considered :

1. Clustering on the FR and JA maps for the R phone,
2. Clustering on the FR and JA maps for the V phone,
3. Clustering on the FR maps for both the V and the R phones,
4. Clustering on the JA maps, for the Correct/Incorrect categories, for the R phone,
5. Clustering on the JA maps, for the Correct/Incorrect categories, for the V phone

We performed each of theses clusterings with four different algorithms : k-means non initialized, k-means with initialized centers, DBSCAN, and MeanShift.

Therefore, for each layer of the neural network, there are 5 types of clusterings, run with 4 different algorithms.

2.2 Activation maps Clustering Results

After the running of the different clustering algorithms on the maps of each layer, the clustering rates are saved in a (.csv) file where the rates represent the percentage of having the first/second item (such as the French or Japanese 'R' for the first clustering, or the 'R' or 'V' french phone for the second type of clustering) in the class 0 (the percentages of the class 1 are complementary to the ones in the classe 0, therefore, we do not take that class into acount).

In parallel, for each of the algorithms, an indices matrix is computed and stored. This matrix is built as follows :

— each column represent a clustering,

— for each clustering, the indices of the maps giving a two-class-result are stored

This matrix allows to keep track of which maps were satisfactorily clusterized. For instance, it doesn't register the maps which were found empty, or, in the case of the DBSCAN and MeanShift algorithms, it stores the few maps that gave two classes during the clustering. When we study wether the clusterings were interesting or not, this indices matrix will help finding the right activation maps and the rates associated.

For each layer, a file named cartesbonclustering is generated as well. This file contains, for each type of clustering, and for each clustering algorithm used, the maps which are said to be well-clusterized ; i.e the maps whose percentages for a given class respect criteria, such as :

— the rate of one item must be greater than 50%, and the other smaller,

— the distance between the two rates must be greater than a fixed value (30 by default)

Those two conditions mean that the clustering identified well the two items and separated them in the two different classes. Thus, for this clustering, and this algorithm, the activation map discriminates well the data and is interesting to keep for the generation of the next layer of the neural network.

When all the files have been generated, and the lists of the well-clustered maps have been entirely set-up, we generate, for each layer, a graph representing the amount of well clustered maps per algorithm and per clustering, and an other, describing, for each activation map of the layer, the number of times it has been considered as well clustered.

2.2.1 Visualization & Figures

The figure below represents all the "well-clustered" activation maps for the layers : conv1, conv2 and mp2

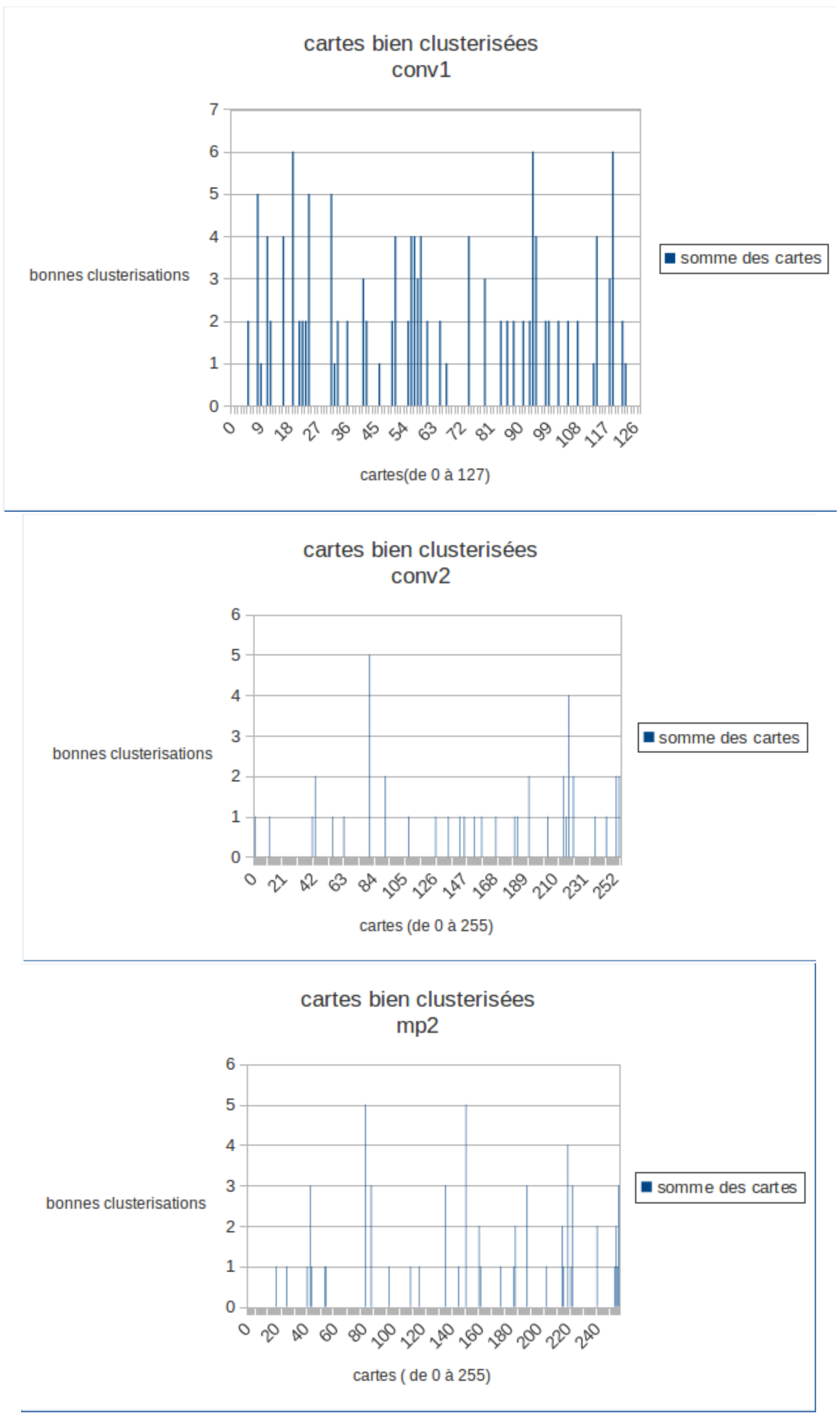


FIGURE 2.1 – Well-clustered maps for the conv1, conv2, and mp2 layers

The figure below represents the efficience of the different clustering algorithms, for each layer. the "1" tag refers to the "FR/JA R" clustering, the "1bis" to the "FR/JA v", "2" to "FR R/V", and "3" and "3bis" to "JA correct/incorrect R", and "JA correct/incorrect V" :

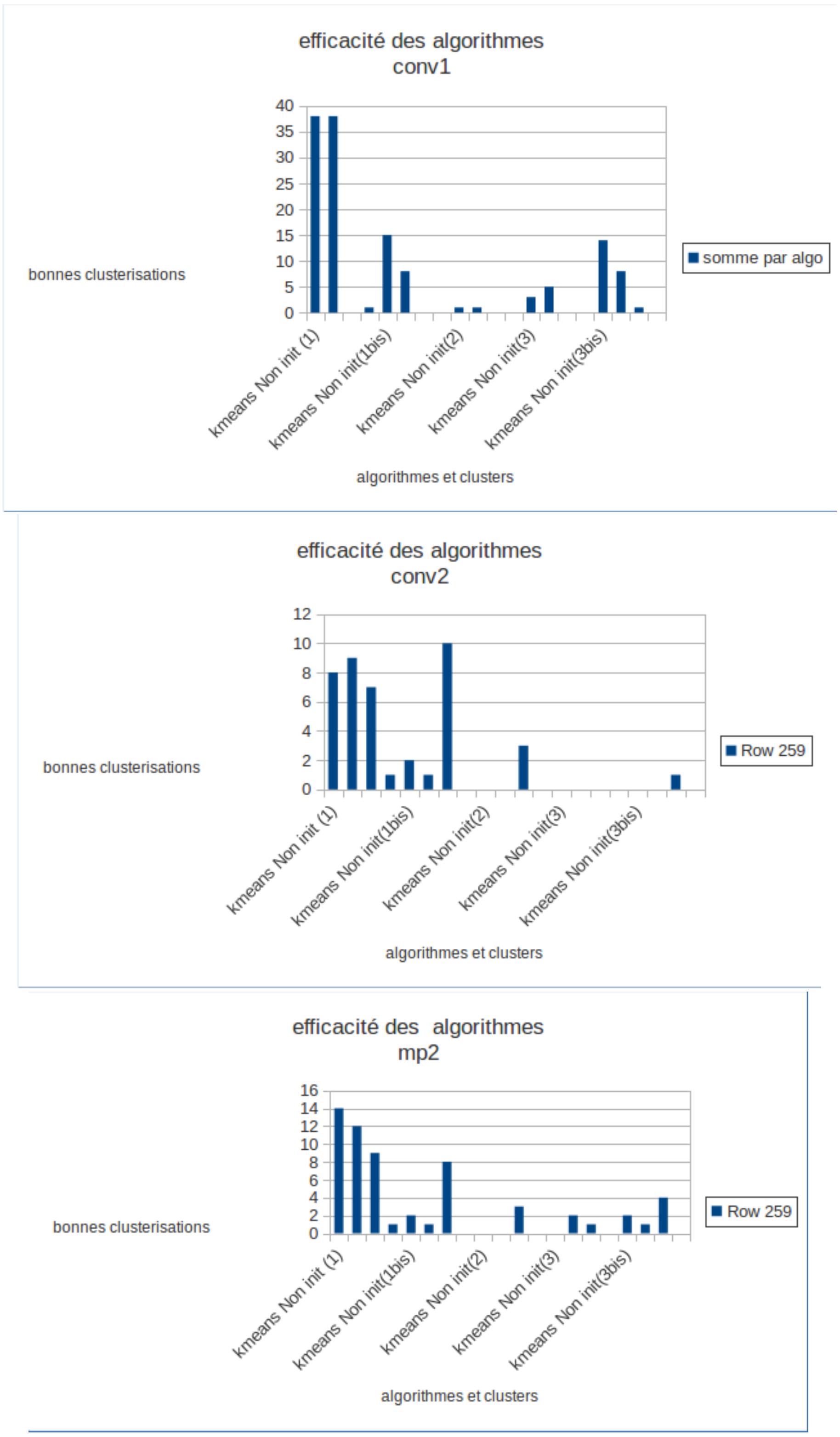
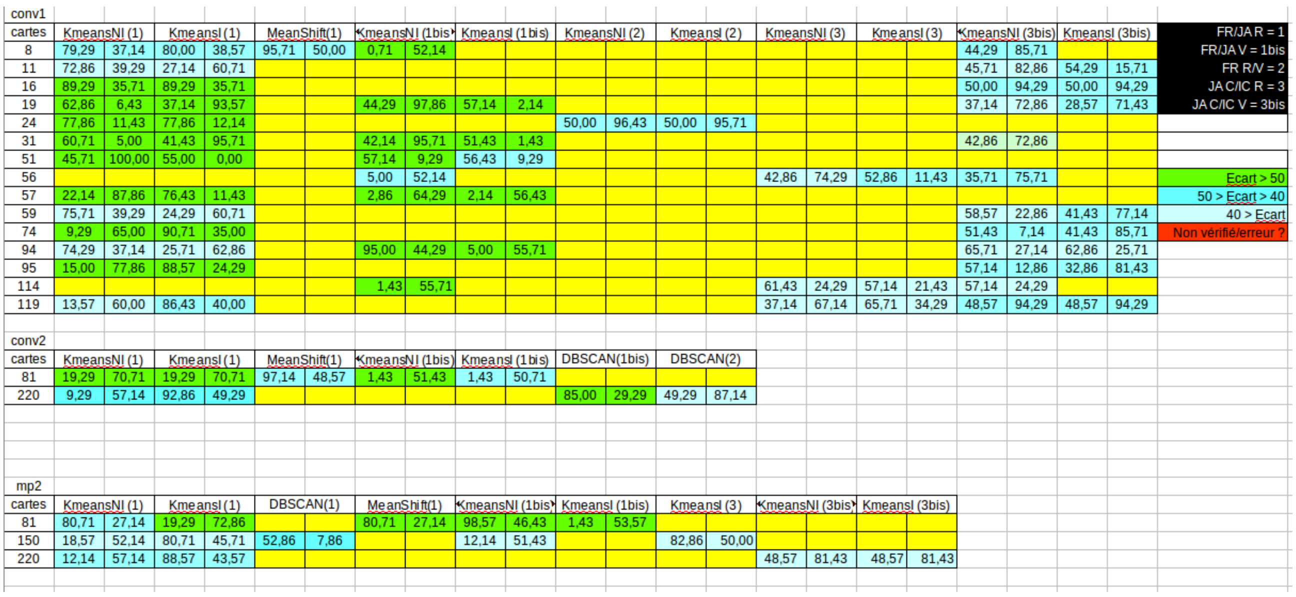


FIGURE 2.2 – Efficiency of clustering algorithms for the 3 layers

There is the table that recaps the maps that have been the most well clustered among the different algorithms and clusterings, for each of the three layers that gave results.

2.2. Activation maps Clustering Results



conv1

cartes	KmeansNI (1)		KmeansI (1)		MeanShift(1)		KmeansNI (1bis		KmeansI (1bis)		KmeansNI (2)		KmeansI (2)		KmeansNI (3)		KmeansI (3)		KmeansNI (3bis)		KmeansI (3bis)	
8	79,29	37,14	80,00	38,57	95,71	50,00	0,71	52,14											44,29	85,71		
11	72,86	39,29	27,14	60,71															45,71	82,86	54,29	15,71
16	89,29	35,71	89,29	35,71															50,00	94,29	50,00	94,29
19	62,86	6,43	37,14	93,57			44,29	97,86	57,14	2,14									37,14	72,86	28,57	71,43
24	77,86	11,43	77,86	12,14							50,00	96,43	50,00	95,71								
31	60,71	5,00	41,43	95,71			42,14	95,71	51,43	1,43									42,86	72,86		
51	45,71	100,00	55,00	0,00			57,14	9,29	56,43	9,29												
56							5,00	52,14							42,86	74,29	52,86	11,43	35,71	75,71		
57	22,14	87,86	76,43	11,43			2,86	64,29	2,14	56,43												
59	75,71	39,29	24,29	60,71															58,57	22,86	41,43	77,14
74	9,29	65,00	90,71	35,00															51,43	7,14	41,43	85,71
94	74,29	37,14	25,71	62,86			95,00	44,29	5,00	55,71									65,71	27,14	62,86	25,71
95	15,00	77,86	88,57	24,29															57,14	12,86	32,86	81,43
114							1,43	55,71							61,43	24,29	57,14	21,43	57,14	24,29		
119	13,57	60,00	86,43	40,00											37,14	67,14	65,71	34,29	48,57	94,29	48,57	94,29

FR/JA R = 1	
FR/JA V = 1bis	
FR R/V = 2	
JA C/IC R = 3	
JA C/IC V = 3bis	
🟩	Ecart > 50
🟦	50 > Ecart > 40
▨	40 > Ecart
🟥	Non vérifié/erreur ?

conv2

cartes	KmeansNI (1)		KmeansI (1)		MeanShift(1)		KmeansNI (1bis)		KmeansI (1bis)		DBSCAN(1bis)		DBSCAN(2)	
81	19,29	70,71	19,29	70,71	97,14	48,57	1,43	51,43	1,43	50,71				
220	9,29	57,14	92,86	49,29							85,00	29,29	49,29	87,14

mp2

cartes	KmeansNI (1)		KmeansI (1)		DBSCAN(1)		MeanShift(1)		KmeansNI (1bis		KmeansI (1bis)		KmeansI (3)		KmeansNI (3bis		KmeansI (3bis)	
81	80,71	27,14	19,29	72,86			80,71	27,14	98,57	46,43	1,43	53,57						
150	18,57	52,14	80,71	45,71	52,86	7,86			12,14	51,43			82,86	50,00				
220	12,14	57,14	88,57	43,57											48,57	81,43	48,57	81,43

Figure 2.3 – Recapitulative Table of the most well-clustered maps

We can observe that the clustering results on those maps is generally excellent, and that it is the same for conv2 and mp2, which makes sense.