

Introduction général

Dans la nature humaine, parler est un besoin, l'un des plus élémentaires peut-être. Il en est le plus distinctif de par sa forme, son impact et surtout de par sa variabilité.

Les différents peuples communiquent dans leurs langues maternelles. Ils apprennent à leurs progénitures dès leurs naissances, les mots, les intonations, les phrases, les manières de dire les choses, de décrire les objets et ceci d'une part, pour uniformiser leur parler et d'autre part, pour développer leur présence et renforcer leur existence.

Les différentes langues se diffèrent, certes, par leur vocabulaire, grammaire et leur prononciation. Mais l'unicité du canal vocal et des différents articulateurs vocaux, les aligne à pied égal. Toutefois, ce moyen de communication peut être affecté de certaines anomalies que l'on définit par pathologie du langage. Celle-ci doit être traitée pour pouvoir communiquer. Généralement ces anomalies touchent surtout l'appareil articulaire par conséquent il nécessite différentes thérapies.

Le traitement de la parole est aujourd'hui une composante fondamentale des sciences de l'ingénieur. Situé au croisement du traitement du signal numérique et du traitement du langage.

Cette discipline scientifique a connu une expansion fulgurante liée au développement des moyens et des techniques de télécommunication.

L'importance particulière du traitement de la parole dans ce cadre plus générale s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine.

L'extraordinaire singularité de cette science qui la différencie fondamentalement des autres composantes du traitement de l'information, tient sans aucun doute un rôle fascinant que joue le cerveau humain à la fois dans la production et dans la compréhension de la parole

et à l'étendue des fonctions qu'il met, inconsciemment en œuvre, pour y parvenir de façons pratiquement instantanée.

Notre étude s'intègre dans un cadre générale du développement des systèmes de reconnaissance des phonèmes par classification bayésienne pour cela on a expose quelques méthodes de classification et présente cette classification à l'aide de l'algorithme k-means et viterbi

Notre projet est par conséquent organisé de la manière suivante :

- Le premier chapitre consacré aux généralités sur le signal de parole ;
- Le deuxième chapitre résume les notions théoriques de probabilité et des modèles Gaussien ainsi que, les différentes étapes du fonctionnement des algorithmes entraînés et expérimentés;
- Le troisième chapitre expose les expériences réalisées, ainsi que les résultats obtenus et leurs interprétations

Nous terminerons Cette mémoire par une conclusion qui résume les résultats obtenus au cours de notre travail.

1.1 Introduction

La parole est la capacité de l'être humain de communiquer la pensée par l'intermédiaire de sons articulés. Dû à son importance, la parole a préoccupé depuis toujours les scientifiques. Ainsi quelques-unes des sciences qui se préoccupent de l'étude de la parole ont déjà des centaines d'années.

Dans ce chapitre on va voir une approche générale sur l'identité du signal de parole commençant par tout ce qui concerne sa production y compris l'architecture de l'appareil vocal, le mécanisme de la phonation et la classification des phonèmes seront aussi abordé dans le contenu de ce chapitre. Nous allons aussi voir le niveau acoustique du signal de parole particulièrement son spectre sa fréquence fondamentale son énergie....etc.

1.2 Production de Parole

D'un point de vue mathématique, il est difficile de modéliser le signal de parole, compte tenu de sa variabilité.

Le signal de parole $s(n)$ est le résultat de l'excitation $u(n)$ du conduit vocal par un train d'impulsion ou un bruit avec un gain G donnant lieu respectivement aux sons voisés et non voisés (la figure 1). Dans le cas des sons voisés, l'excitation est une vibration périodique des cordes vocales suite à la pression exercée par l'air provenant de l'appareil respiratoire. [20]

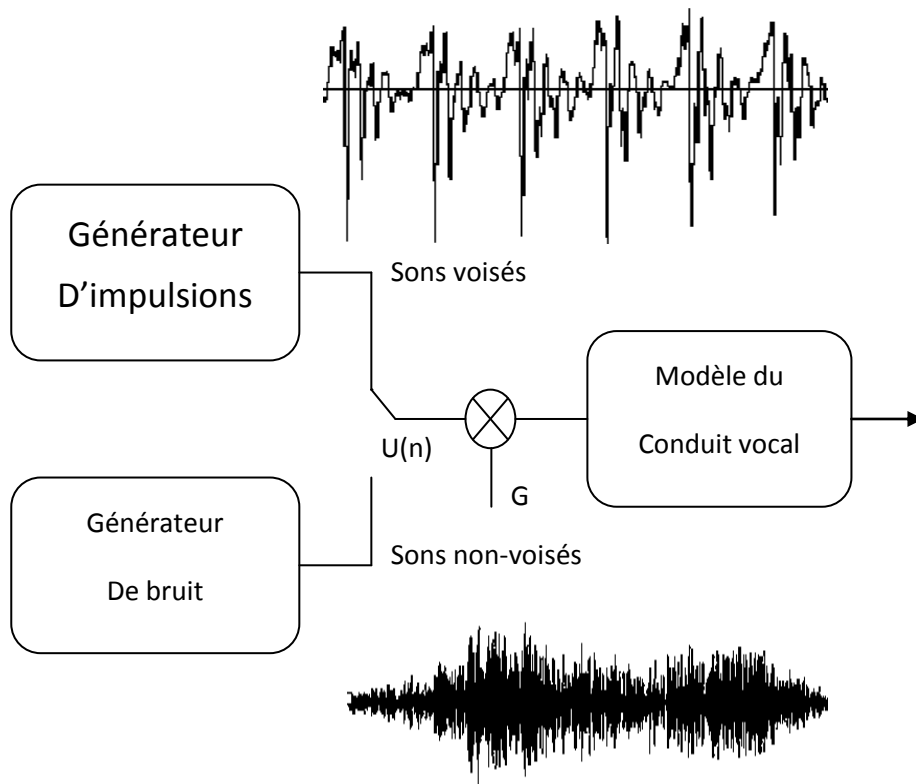


Figure (1.1) : Modèle Simple de Production de Parole

1.2.1 Architecture de l'Appareil Vocal

L'appareil vocal, ou système phonatoire, comprend quatre éléments fondamentaux fonctionnant en étroite synergie pour produire des signaux acoustiques. Ce sont, dans l'ordre où ils s'élaborent

- La soufflerie
- Le vibrateur
- Le corps sonore
- Le système articulateur

La soufflerie est constituée d'un réservoir d'air, les poumons, actionnés par les muscles du thorax et de l'abdomen, et d'un tube, la trachée artère, qui conduit l'air aux cordes vocales ; le vibrateur est le larynx, qui engendre les ondes aériennes; le corps sonore est constitué d'un

ensemble complexe de résonateurs, dont le pharynx et la bouche sont les principaux; le système articulateur, enfin, se compose d'éléments fixes et mobiles permettant de modifier largement la forme de l'onde laryngée. Tous ces éléments sont placés sous la dépendance étroite du système nerveux central, qui en assure le synchronisme et la coordination [1].

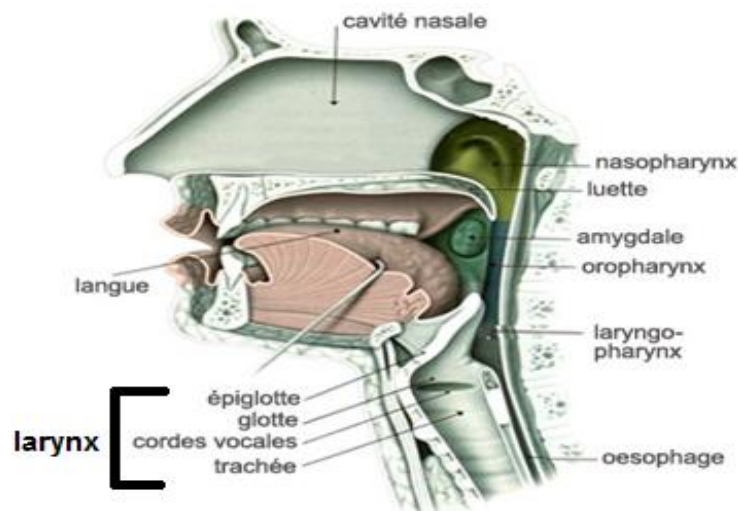


Figure (1.2) : Appareil Vocal

1.2.1.1 Soufflerie et Vibration

L'air est la matière première de la voix. Si le fonctionnement de notre appareil vocal est souvent comparé à celui d'un instrument de musique, il doit être décrit comme celui d'un instrument à vent. En effet, en expulsant l'air pulmonaire à travers la trachée, le système respiratoire joue le rôle d'une soufflerie. Il s'agit du « souffle phonatoire » produit, soit par l'abaissement de la cage thoracique, soit dans le cadre de la projection vocale par l'action des muscles abdominaux [3]

1.2.1.2 Larynx

Le larynx constitue l'extrémité supérieure de la trachée artère, situé à la hauteur de la sixième vertèbre cervicale (chez l'adulte). C'est un assemblage de cartilages articulés, reliés entre eux par des ligaments et des muscles (dont les cordes vocales), l'ensemble étant tapissé d'une muqueuse [1].

1.2.1.3 Corps Sonore

Les résonateurs du système phonatoire sont, pour l'essentiel, responsables du timbre de la voix. Leur originalité par rapport aux caisses de résonance des instruments de musique traditionnels est leur faculté de changer, grâce à un réseau musculaire dense et élaboré, — dans de larges proportions, et très rapidement — de forme et de volume, assurant ainsi au son vocal une variété acoustique sans équivalent.

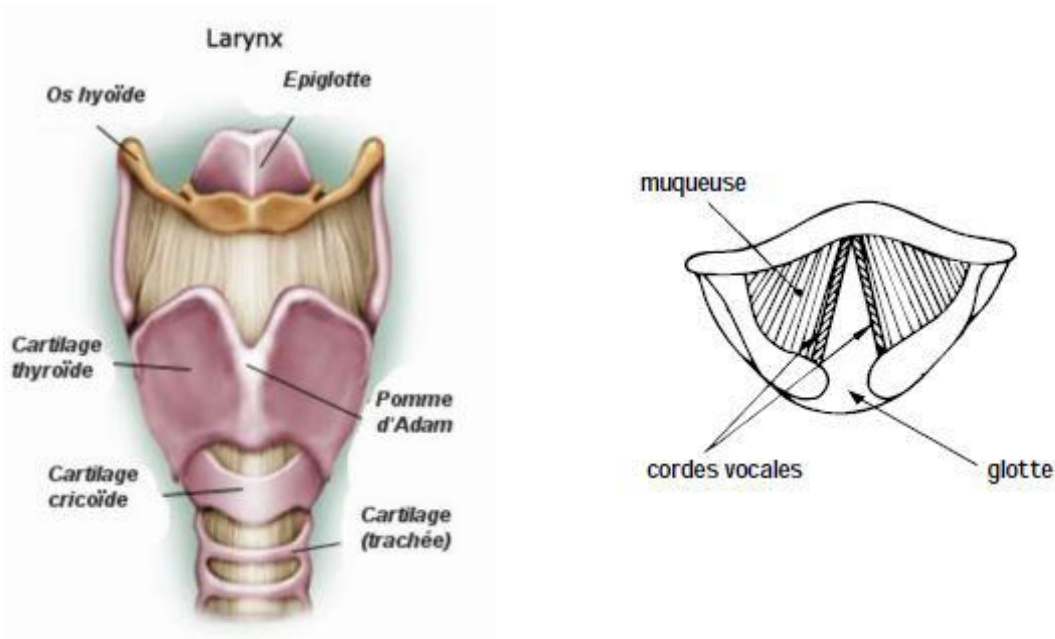


Figure (1.3) : Vue Schématique Antérieure du Larynx.

Les résonateurs sont au nombre de cinq : le pharynx, la cavité buccale divisée en deux, la cavité labiale, et la cavité nasale. Tous communiquent entre eux par des ouvertures de tailles réglables. Tapissés de muqueuse, ils sont peu amortis.

Les sinus de la face, contrairement à une croyance largement répandue, sont trop petits pour se comporter en résonateurs, et n'ont d'ailleurs aucune fonction reconnue dans la phonation [1].

1.2.1.4 Système Articulateur

Il intègre un ensemble d'organes mobiles, le voile du palais, la mâchoire inférieure (ou mandibule), la langue et les lèvres. Les mouvements de la mâchoire inférieure contribuent largement aux variations de volume de la bouche. La langue, reliée par sa base à l'os hyoïde, est extrêmement mobile, car commandée par dix-sept muscles (huit pairs et un impair) ; elle prend appui sur différents points du conduit pharyngo-buccal pour articuler les phonèmes. Ces points d'articulation sont [1] :

- les lèvres (articulations labiales ou bilabiales)
- les dents (articulations dentales)
- les alvéoles 16 (articulations alvéolaires)
- le palais dur, ou partie osseuse de la voûte (articulations palatales)
- le voile du palais ou « palais mou » (articulations vélares)
- la luette (articulations uvulaires)
- le pharynx (articulations pharyngales)
- la glotte (articulations glottales)

1.2.2 Mécanisme de Phonation

L'une de plus importantes caractéristiques du signal vocal est la nature de l'excitation. Il existe deux types élémentaires d'excitation qui produisent les sons voisés et non voisés.

1.2.2.1 Phonation de Sons Voisés

Les sons voisés sont produits à partir d'une excitation qui actionne sur le conduit vocal et qui consiste en une suite d'impulsions périodiques d'air fournies par le larynx. Les cordes vocales au début sont fermées. Sous la pression continue de l'air qui vient des poumons elles s'ouvrent graduellement délivrant cette énergie potentielle. Pendant cette ouverture la vitesse de l'air et l'énergie cinétique augmentent jusqu'à ce que la tension élastique des cordes vocales égale la force de séparation du courant d'air. A ce point l'ouverture de la glotte est maximale. L'énergie cinétique qui a été accumulée comme tension élastique dans les cordes vocales commence à rétrécir cette ouverture et de plus la force de Bernoulli accélère encore la

fermeture abrupte de la glotte [3]. Ce processus périodique est caractérisé par une fréquence propre à chaque personne, connue sous le nom de fréquence fondamentale (F_0) ou pitch et il donne la hauteur normale de la voix.

La fréquence fondamentale peut varier de 80 à 200 Hz pour une voix masculine, de 150 à 450 Hz pour une voix féminine et de 200 à 600 Hz pour une voix d'enfant [4].

Cette fréquence fondamentale peut varier suite à des facteurs liés au stress, intonation et émotions. Le timbre de la voix est déterminé par les amplitudes relatives des harmoniques du fondamental.

L'intensité du son émis est liée à la pression de l'air en amont de larynx. Tous ces aspects pour un son voisé peuvent être observés dans la figure 1.3.

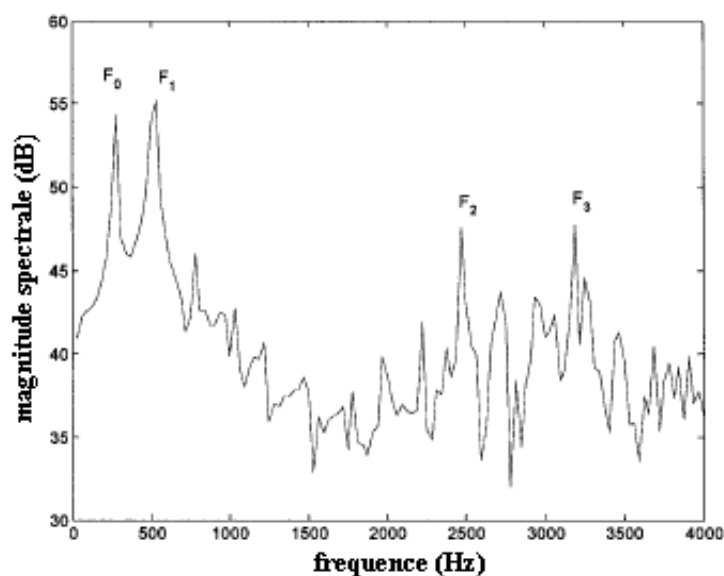


Figure (1.4) : Spectre du Son Voisé / i /

1.2.2.2 Phonation de Sons non Voisés

Les sons non voisés sont générés par le passage de l'air dans une constriction étroite situé en un point du conduit vocal. Ils sont générés sans l'apport du larynx et ne présentent pas de structure périodique [4]. Ces caractéristiques d'un son non voisé peuvent être observées dans la figure (1.4).

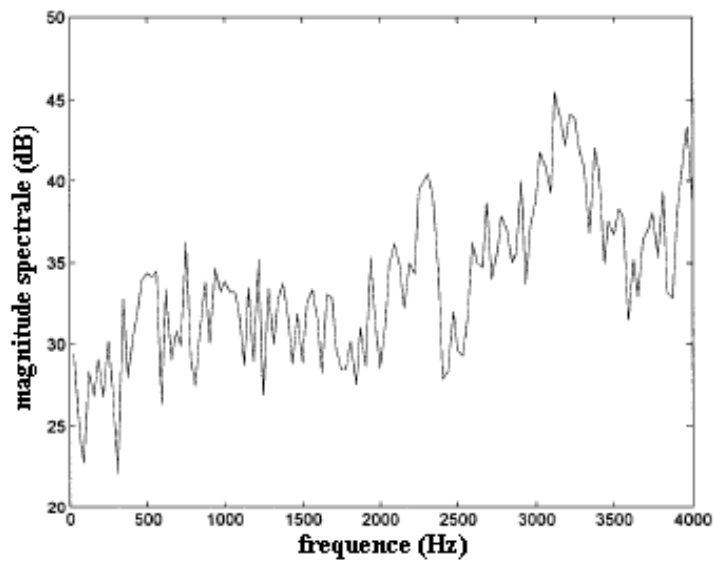


Figure (1.5) : Spectre du Son non Voisé / f /

1.2.3 Classification des Phonèmes

Il y a plusieurs façons de classier les phonèmes. Un phonème est stationnaire ou continuant si la configuration du conduit vocal ne change pas pendant la production du son. Un phonème est non continuant si pendant sa production il y a des changements dans la configuration du conduit vocal [6].

On peut grouper les 36 phonèmes de la langue française en classes et sous classes d'après le mode d'articulation de l'appareil de phonation (tableau I.1).

1.2.3.1 Voyelles

Les voyelles sont des sons voisés, continus, normalement avec la plus grande amplitude parmi tous les phonèmes et elles peuvent varier beaucoup en durée, entre 40 et 400 ms.

Les voyelles orales sont produites sans l'intervention de la cavité nasale pendant que pour les voyelles nasales, le conduit nasal est couplé à la cavité buccale et la production de son se fait par la bouche et par les narines en même temps. Les voyelles sont différenciées en trois

groupes d'après la position de la courbure de la langue et le degré de la constriction induit dans le conduit vocal.

Différenciées en trois groupes d'après la position de courbure de langue et le degré de constriction induit dans le conduit vocal.

L'analyse dans le domaine temporel et fréquentiel révèle plusieurs caractéristiques acoustiques qui aident à la classification de chaque son. L'analyse dans le domaine temporel montre que les voyelles sont de sons quasi périodiques dus à l'excitation.

Les voyelles peuvent être identifiées par les locations de leurs formants dans le domaine fréquentiel. La position des deux premiers formants est suffisante pour caractériser la majorité des voyelles, le troisième formant est nécessaire juste pour quelques-uns. La position de formants de fréquence plus élevée reste presque inchangée et n'apporte pas d'information utile pour l'identification.

Phonèmes								
Voyelles		Semi- Consonnes	Consonnes					
Orales	Nasales		Liquides	Nasales	Fricatives		Occlusives	
					Voisées	Non- Voisées	Voisées	Non- Voisées
i(I)	ẽ (IN) œ(UN) ã (AN) õ(ON)	j (Y) w(W) ɥ(UI)	l(L) r(R)	m(M) n(N) ɲ(GN)	v(V) z(Z) ʒ(J)	f(F) s(S) ʃ(CH)	b(B) d(D) g(G)	p(P) t(T) k(K)
e(E)								
ɛ(AI)								
a(A)								
ɔ(O)								
u(OU)								
y(U)								
ø(EU)								
œ(OE)								
ə(E)								
o(AU)								

Tableau (1.1) : Phonèmes de Langue Française.

1.2.3.2 Diphtongues

Les diphtongues impliquent un mouvement d'une voyelle initiale vers une autre voyelle finale. Donc les diphtongues sont essentiellement des sons non continus.

La différence entre une diphtongue et deux voyelles individuelles composantes est sa durée de transition est plus grande que la durée de chaque voyelle. De plus la voyelle initiale est plus longue que la voyelle finale. Dans la parole les deux voyelles composant une diphtongue peuvent ne pas être réalisées entièrement ce qui accentue l'idée de non-stationnarité qui caractérise les diphtongues.

1.2.3.3 Semi-Consonnes

Les semi -consonnes sont des sons non continus et voisés qui possèdent des caractéristiques spectrales semblable aux voyelles. On peut considérer que les semi-consonnes comme des sons transitoires qui s'approchent, puis s'éloignent d'une position cible. La durée des transitions est comparable à la durée transition vers une position cible.

1.2.3.4 Consonnes

Le conduit vocal est plus étroit pendant la production des consonnes par rapport aux voyelles. Les consonnes impliquent les deux formes d'excitation pour le conduit vocal et elles peuvent être voisées ou non voisées.

➤ Consonnes Fricatives

Les fricatives non voisées résultent d'une turbulence créée par le passage de l'air dans une constriction du conduit vocal qui peut se trouver près des lèvres pour les labiales, au milieu du conduit vocal pour les dentales et au fond du conduit vocal pour le palatales.

Dans ce cas la constriction cause une source de bruit et aussi divise le conduit vocal en deux cavités. La première cavité actionne comme une enceinte anti-résonante qui atténue les basses fréquences d'où la concentration de l'énergie vers les hautes fréquences dans le domaine spectral.

Pour les fricatives voisées l'excitation est mixte et à la source de bruit s'ajoutent les impulsions périodiques créées par la vibration de cordes vocales.

➤ *Consonnes Occlusives*

Les consonnes occlusives sont des combinaisons de sons voisés, non voisés et de courtes périodes de silence. Une forte pression d'air s'accumule avant une occlusion totale dans un point du conduit vocal qui est relâché après brusquement. Cette période d'occlusion s'appelle la phase de tenue.

Pour les occlusifs non voisés la phase de tenue est un silence et la période de friction qui suit est plus longue que pour les occlusives voisés. Pour les occlusives voisés, pendant la phase de tenue, un son de basse fréquence est émis par vibration des cordes vocales.

➤ *Consonnes Nasales*

Les consonnes nasales sont des sons continus et voisés. Les vibrations produites par les cordes vocales excitent le conduit vocal que cette fois est formé de la cavité nasale ouverte et la cavité buccale fermée. Même fermée, la cavité buccale est couplée à la cavité nasale et influence la production de sons comme une enceinte anti-résonante qui atténue certaines fréquences, en fonction du point où elle est fermée. Les formes d'onde des consonnes nasales ressemblent aux celles des voyelles mais sont normalement plus faibles en énergie due à la capacité réduite de la cavité nasale de radier des sons par rapport à la cavité buccale.

➤ *Consonnes Liquides*

Les consonnes liquides sont des sons non continus et voisés qui possèdent des caractéristiques spectrales similaires aux voyelles. Elles sont plus faibles en énergie due au fait que le conduit vocal est plus étroit pendant leur production.

1.3 Modélisation du Processus de Perception de Parole

La seconde modélisation consiste naturellement à considérer non plus la production mais la perception humaine [6]. Il s'agit donc de modéliser les caractéristiques de l'oreille humaine [7]. Ces modèles sont appelés modèles auditifs (Auditory Models) [8], [9], [10].

Depuis de nombreuses années, des études ont été menées pour essayer de simuler l'oreille humaine [11]. Cependant, la réplique des différents phénomènes n'améliore pas systématiquement les performances des systèmes de reconnaissance [12].

Les principales étapes de la perception humaine sont représentées sur la partie gauche de la figure (1.6). On peut également trouver sur la partie droite de cette figure la transcription de ces étapes au domaine du traitement du signal ou de l'information.

On constate que les opérations de filtrage ou la mise en place d'échelle non linéaire (Mel ou Bark) sont des techniques largement utilisées dans les méthodes de codage traditionnelles. D'autres techniques, basées sur la connaissance du processus de perception, ont été développées pour la conception de modèles auditifs [13].

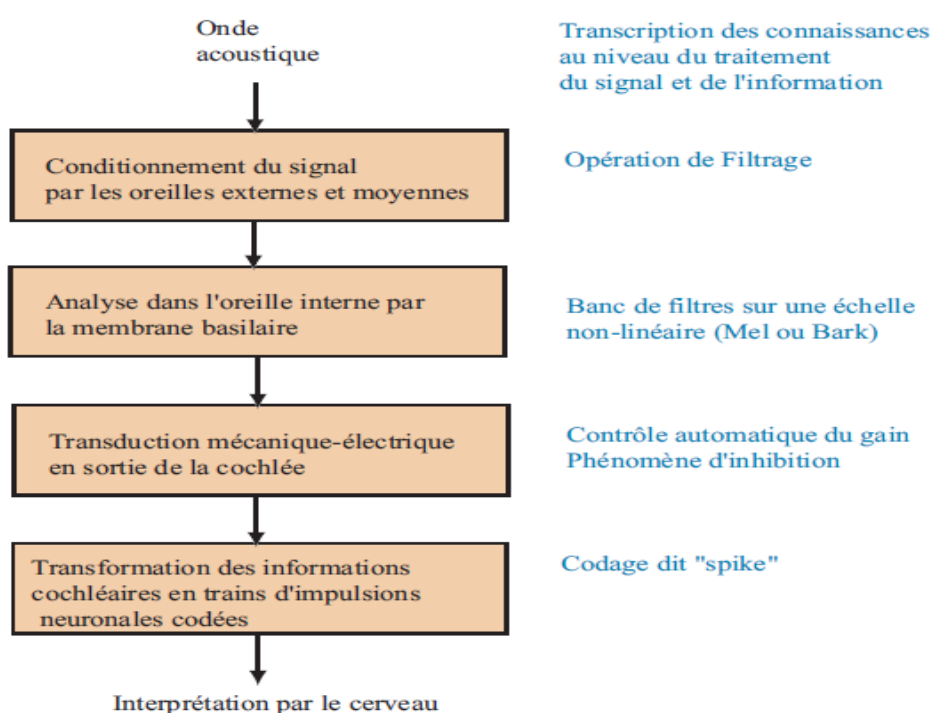


Figure (1.6) : Modèle de Perception Humaine

Malgré le développement de ces nombreux modèles, il s'avère difficile de trouver un compromis entre la modélisation fidèle du processus de perception humaine et une extraction de caractéristiques efficace. Hermansky [14], qui précise que la copie intégrale du processus de perception n'est sans doute pas la meilleure solution pour améliorer les scores de

reconnaissance, propose des voies intéressantes pour la conception d'extracteurs de caractéristiques. On peut citer comme exemples significatifs :

- Bandes critiques (banc de filtres).
- Echelle Mel (MFCC: Mel Frequency Cepstrum Coefficients) et Bark (PLP: Perceptual Linear Predictive) [15].
- Production de la parole (LPCC : Linear Predictive Cepstral Coding).

1.4 Analyse de Parole

L'analyse de la parole permet la mise en forme du signal mais aussi l'extraction de paramètres nécessaires pour les prochaines étapes telles que la reconnaissance.

Un des objectifs de cette analyse est d'obtenir une représentation compacte et informative du signal. Le signal de parole est un signal redondant et non stationnaire mais il peut être considéré comme localement stationnaire. L'analyse du signal de parole se fait pendant ces périodes stationnaires dont la durée varie de 10 à 30ms.

Cette durée correspond aussi à la durée de stabilité du modèle de production.

L'analyse de la parole consiste à effectuer des prétraitements, nécessaires pour la mise en forme du signal, tels que le découpage en fenêtres. La seconde étape de l'analyse est l'extraction de caractéristiques.

1.4.1 Prétraitements

Les prétraitements débutent par un échantillonnage des signaux (figure I.7), suivit d'une préaccentuation. Le signal $s(n)$ est divisé en fenêtres de longueur N (10-20ms). Le signal final $x(n)$ est obtenu par une multiplication du signal (n) par une fonction, ou encore fenêtre, de pondération non nulle $w(n)$:

$$X(n) = s(n) w(n) \quad (I.1)$$

La préaccentuation est un exemple d'utilisation de connaissances sur la perception humaine. Elle consiste en un filtrage du signal de parole par le filtre suivant :

$$Y(z) = (1 - a z^{-1}) X(z) \quad (I.2)$$

Le choix de la fenêtre est très important. Parmi les fenêtres utilisées, on peut citer les fenêtres de Hamming, Hanning, Blackman ou de Kaiser. Le choix se fait le plus souvent en fonction de l'application car les fenêtres présentent différentes atténuations à des fréquences bien précises.

Cependant, il faut noter que la plupart des systèmes sont directement conçus sur des fenêtres de Hamming.

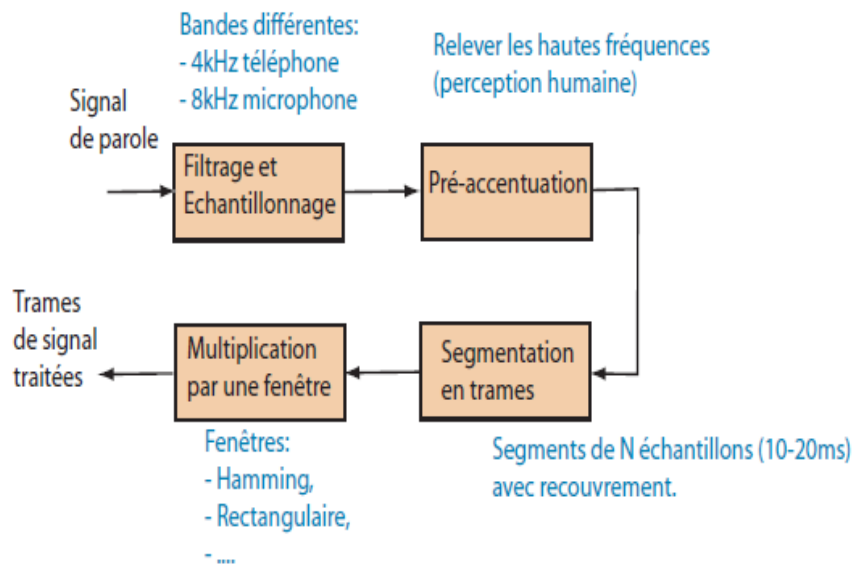


Figure (1.7) : Prétraitements Nécessaires

1.4.2 Extraction de Paramètres

L'extraction de paramètres est l'objet principal de l'analyse de la parole et c'est le passage obligé de toutes les applications en traitement de la parole (figure I.8).

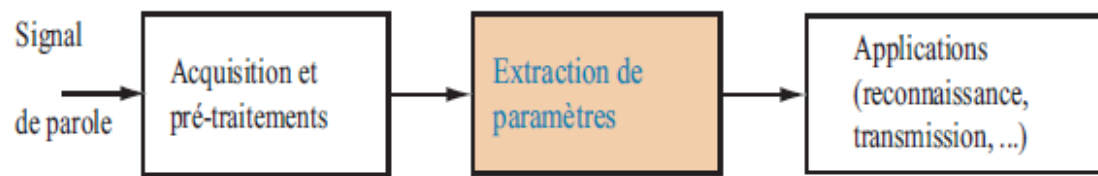


Figure (1.8) : Traitement de Parole

L'extraction de caractéristiques, avec ou sans apprentissage, est avant tout la recherche d'une représentation du signal de parole adaptée à l'application. Généralement, le vecteur caractéristique est formé de plusieurs caractéristiques telles que, par exemple :

- Vecteur code : LPCC, MFCC, PLP, ...
- Paramètres Δ et $\Delta\Delta$: dérivées premières et secondes du vecteur code. On cherche à modéliser la dynamique du signal de parole,
- Energie et sa dérivée,
- Le taux de passage par zéros (ZCR : Zero Cross Rate) et sa dérivée,
- La fréquence fondamentale F_0 (pitch).
- Le voisement.

Dans la suite, nous nous intéresserons principalement au vecteur code. En effet, une fois le vecteur code extrait, les autres caractéristiques peuvent être ajoutées pour former le vecteur caractéristique.

Le vecteur code est une représentation du signal de parole. Une des classifications possibles des méthodes de représentation des signaux de parole est la suivante :

- Méthode paramétrique : utilisation d'un modèle (LPC,...). [16]
- Estimation de paramètres à partir de méthodes non-paramétrique (MFCC,...). [17]
- Extracteur de paramètres à grands échelles en temps réel.

1.5 Synthèse de Parole

Dans le cas de la parole le terme «synthèse» implique une combinaison des sons et des bruits et de réalisé des mots ou des phrases.

L'objectif de la synthèse de la parole et de produire des sons de parole a partir d'une représentation phonétique du message. [16]

1.5.1 Techniques de Synthèse de Parole

Pour transmettre une information donnée entre un émetteur et un récepteur il faut utiliser des techniques qui permettent une représentation du signal sous forme réduit (codage) tel que, après transmission (stockage), on peut reconstituer le signal original (décodage) à partir de ces informations figure (1.9).

Le principe de synthétiseur est de créer une analogie avec l'appareil phonatoire humain. Les différentes techniques offertes par les synthétiseurs de la parole sont :

- Synthétiseur a canaux.
- Synthèse par formant.
- Synthèse basée sur la prédiction linéaire.

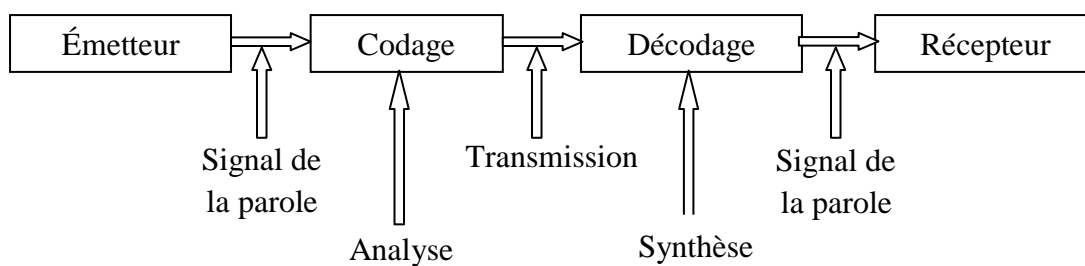


Figure (1.9) : Synthèse et Codage de Parole.

1.6 Reconnaissance de Parole

La reconnaissance automatique de la parole consiste à extraire l'information lexicale contenue dans un signal de parole (signal électrique obtenue à la sortie d'un microphone et typiquement échantillonné à 8khz dans le cas de ligne téléphoniques ou entre 10 et 16khz dans le cas de saisie par microphone). Bien que ceci soulève également le problème de la compréhension de la parole.

La compréhension de la parole est une étape supplémentaire qui consiste à exiger de la machine une action adéquate au contenu du message vocal [18].

Elle contient plusieurs système peut être classé on :

1.6.1 Reconnaissance de Mot Isolé

Elle est effectuée normalement au niveau acoustique. Les mots du vocabulaire constituent des références aux quelles un mot à reconnaître doit être comparé.

Il faut donc définir d'une façon judicieuse une distance ou une mesure de dissemblance entre deux mots ; toute fois il est essentiel que cette distance soit peu sensible aux petites variations relatives des axes des temps. Apres l'alignement optimal des axes temporel, la référence la plus proche désigne le mot à identifier. Pour cela on utilise un algorithme basé sur la programmation dynamique appelé la DTW (Dynamique Time Warping).

1.6.2 Reconnaissance de Mots Enchainés

Le problème consiste à identifier une suite finie de mots prononcés d'une façon continue; ces mots appartiennent à un vocabulaire connu qui comporte un nombre limité de mots.

La méthode sera donc basée sur une comparaison globale de la suite à reconnaître avec des suit de référence.

1.6.3 Reconnaissance du Locuteur

C'est un terme générique pour discriminer parmi plusieurs personnes en fonction de leur voix (les émotions et les sons pathologiques : ce qui est utilisé dans notre projet) on distingue en général l'identification et la vérification du locuteur.

L'identification consiste à reconnaître un locuteur appartenant à une population de N locuteur.

La vérification consiste à accepter ou à refuser une identité proclamée par un locuteur ; on compare la distance entre son expression vocale et sa référence [18].

1.7 Description du Signal de Parole

1.7.1 Variabilité intra-locuteur

La variabilité intra-locuteur identifie les différences dans le signal produit par la même personne. Cette variabilité peut résulter de l'état physique ou moral du locuteur. Une maladie des voies respiratoires peut ainsi dégrader la qualité du signal de parole de manière à ce que celui-ci devienne totalement incompréhensible, même pour un être humain. L'humeur ou l'émotion du locuteur peut également influencer son rythme vocal, son intonation.[19]

1.7.2 Variabilité inter-locuteur

Cette variabilité est un phénomène majeur en reconnaissance de la parole. La nature de des différences entre interlocuteurs la plus part du temps est de nature physiologique telle que la forme et la longueur du conduit vocal ainsi la longueur de la corde vocale, ainsi elles sont liées à l'âge, l'accent régional, et le sexe. [19]

1.7.3 Redondance

Le signal vocal est caractérisé par une très grande redondance, condition nécessaire pour résister aux perturbations du milieu ambiant.

1.8 Conclusion

Dans ce chapitre nous avons décrit les principaux éléments caractérisant le signal de parole. Nous avons aussi abordé l'analyse et les différents types de reconnaissance de parole.

Dans le chapitre suivant nous allons présenter quelques définitions théoriques des probabilités, la loi normale et les modèles Gaussiens. Nous allons aussi décrire les trois algorithmes de classification (Kmeans, EM et Viterbi-EM) qui nous intéressent dans notre projet de fin d'études.

2.1 Introduction

Dans ce chapitre on va présenter quelques approches ou méthodes de classification et de décision s'appuyant sur la distance, le groupement et la probabilité.

Pour cela nous allons présenter quelques définitions théoriques des probabilités, la loi normale et les modèles Gaussiens pour la description des trois algorithmes de classification (Kmeans, EM et Viterbi-EM) que nous utiliserons dans notre projet de fin d'études.

2.2 Notions de Probabilité de Base

2.2.1 Définition de Probabilité

La probabilité est une fonction notée P qui attribue à tout événement 'A' une valeur $P(A)$ désignant la probabilité que A se réalise. Dans le cas général la probabilité $P(A)$ de l'évènement 'A' s'obtient en calculant le rapport du nombre m de cas favorables correspondant à l'évènement sur le nombre n de tous les cas possibles.

Une probabilité possède la propriété suivante : $0 \leq P(A) \leq 1$ pour tout événement A. [22]

$$P(A) = m/n \quad \dots 2.1$$

2.2.2 Probabilité Conditionnelle

Soient A, B deux événements, la probabilité que l'évènement A se réalise sachant que l'évènement B a eu lieu est définie par : [22]

$$P[A|B] = \frac{P[A \cap B]}{P[B]} \quad 2.2$$

$$\{A \text{ et } B\} = A \cap B = \textit{intersection de A et B} \quad 2.3$$

2.3 Classification des Données

La classification est un processus d'estimation de l'appartenance d'un événement ou d'un point x à une des classes $C_k \in \{C_1, C_2, \dots, C_k, \dots, C_K\}$ fondée sur les caractéristiques de l'événement ou sur les critères de décision (K est le nombre de classes). On distingue deux types de classifications. La classification supervisée se caractérise par la connaissance des classes priori, elles ont en général une sémantique associée, par contre pour la classification non supervisée les classes sont fondées sur la structure des données (les caractéristiques).

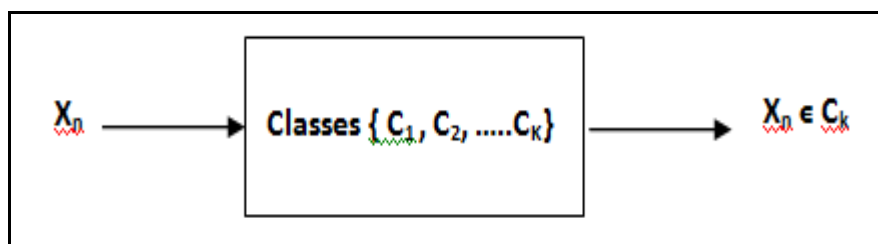


Figure 2.1 : Schéma Synoptique Représentant la Classification d'une Donnée.

2.3.1 Quelques Domaines de Classification

Reconnaissance de Formes

Généralement elle vise à reconnaître ou identifier certaines formes à partir des données brutes afin de prendre une décision dépendant de la catégorie attribuée à cette forme. Cette dernière peut s'agir d'une image (visage, empreinte digitale, rayon ...), sonore (reconnaissance de parole) ou reconnaissance des caractères manuscrits. L'empreinte digitale sert à identifier une personne grâce à son empreinte. Figure de 2.2 à 2.4 [21]

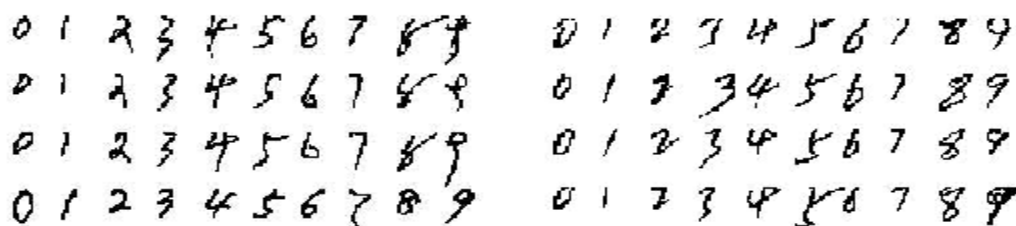


Figure (2.2) : Reconnaissance des Caractères Manuscrits



Figure (2.3) : Reconnaissance d'Empreintes Digitales

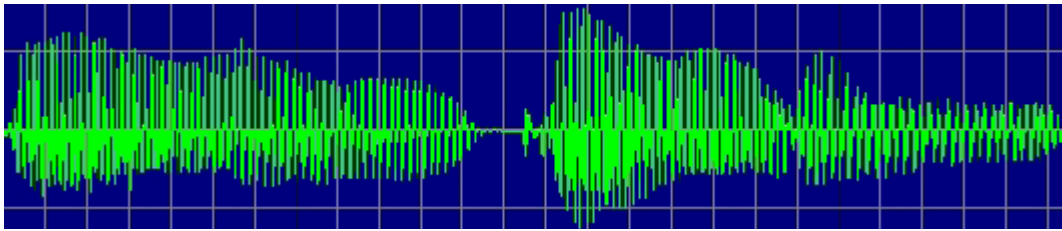


Figure (2.4) : Reconnaissance Vocale

2.4 Critères de Décisions de Classification

Parmi les critères de décisions utilisées lors d'une classification des caractéristiques (données) : qui définissent l'appartenance d'une forme (ou d'une donnée) à une classe avec un minimum de distance par la règle de décision euclidienne ou avec un maximum d'appartenance par la règle de décision Bayésienne qu'est un prolongement du théorème de Bayes, lui-même est une application des probabilités conditionnelles.

2.4.1 Critère de Distance

Nous indiquons deux critères parmi l'ensemble de critères :

2.4.1.1 Critère de Distance Euclidienne

C'est une distance métrique entre deux points $x_n (x_{n1}, x_{n2})$ et $\mu_k (\mu_{k1}, \mu_{k2})$. Elle est égale à la norme du vecteur qui les lie. La norme associée est appelée la norme Euclidienne qui est par définition calculée selon les équations suivantes: 2.4 à 2.6 (figure 2.5)

$$d_k(x_n, \mu_k) = \|x_n - \mu_k\|^2 \quad 2.4$$

$$d_k(x_n, \mu_k) = (x_n - \mu_k)^T (x_n - \mu_k) \quad 2.5$$

$$d_k(x_n, \mu_k) = |x_{n1} - \mu_{k1}| + |x_{n2} - \mu_{k2}| \quad 2.6$$

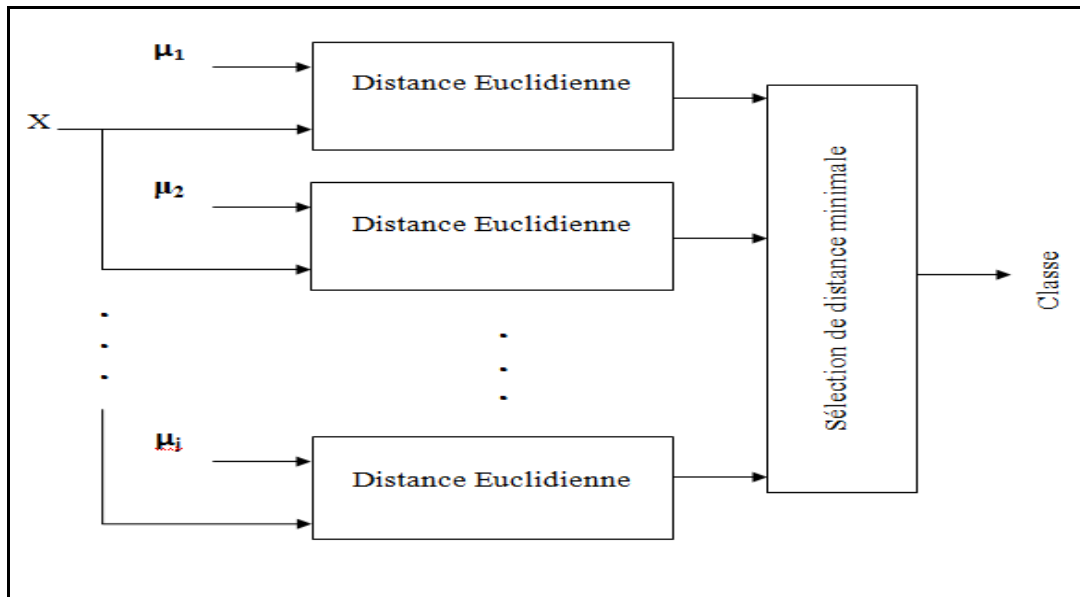


Figure (2.5) : Critère Euclidienne.

2.4.1.2 Critère de Distance de Mahalanobis

Le calcul de distance est obtenu à partir de l'inverse de la matrice de covariance. La distance dite de Mahalanobis s'écrit alors, sous forme matricielle :

$$d(x_i, \mu_k) = (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \quad 2.7$$

Notons que la distance de Mahalanobis est similaire à la distance Euclidienne calculée dans variables décorrelées. (figure 2.6)

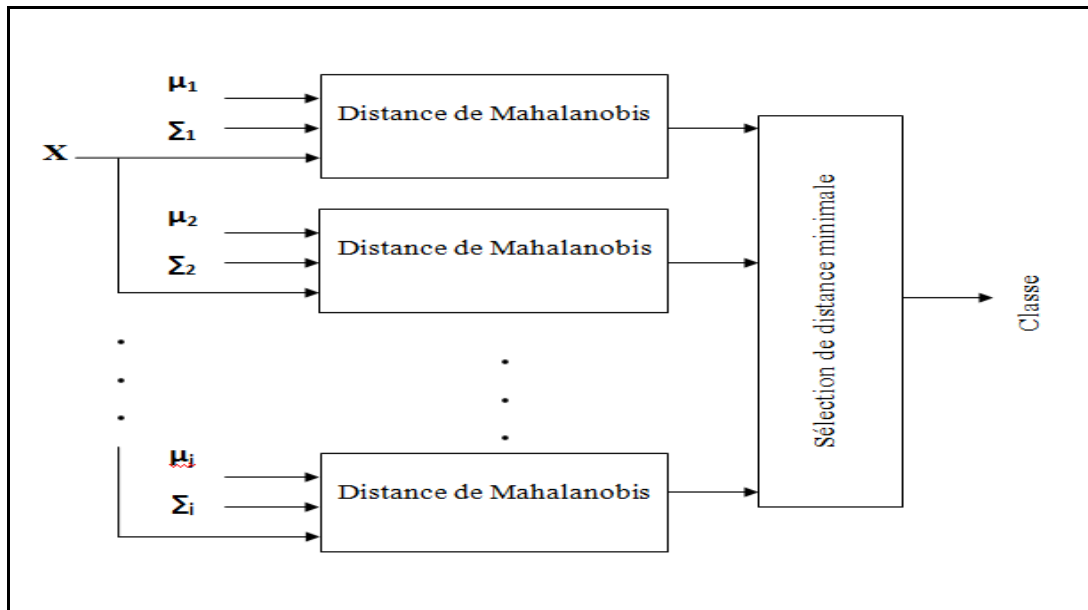


Figure (2.6) : Critère de Mahalanobis.

2.4.2 Critère de Bayes

On considérant X_n la nouvelle donnée à classer et l'ensemble $\{C_1, \dots, C_k\}$ des K différentes classes existantes. Le principe repose essentiellement sur la règle de Bayes. Il est illustré dans l'algorithme suivant :

- ✓ Calcul de la fonction de décision de la nouvelle donnée X_n par rapport à chaque classe existante;
- ✓ Décision finale : la donnée X_n est affectée à la classe C_k dont la fonction de décision est la plus grande selon l'inégalité suivante:

$$P(C_k | X_n) \geq P(C_l | X_n), \forall l \neq k \quad 2.8$$

2.5 Application de Règle Euclidienne

L'appartenance d'une nouvelle donnée X_n à une classe C_k de l'ensemble $\{C_1, \dots, C_K\}$ (des K différentes classes existantes) est basée sur la règle euclidienne illustrée dans l'algorithme suivant :

- ✓ Calcul de la fonction de décision de la nouvelle donnée X_n par rapport à chaque classe ;

$$P(C_k | X_n) = \frac{p(X_n | C_k) P(C_k)}{P(X_n)} \quad 2.9$$

- ✓ Décision finale : la donnée X_n est affectée à la classe C_k dont la fonction de décision est la plus petite selon l'inégalité suivante:

$$d_{k(x_n, \mu_k)} \leq d_{l(x_n, \mu_l)}, \quad \forall l \neq k \quad 2.10$$

2.6 Loi Normale

On dit qu'une variable X de "d" dimensions suit une loi normale (ou gaussienne) de paramètres μ et Σ , notée $N(\mu, \Sigma)$, lorsqu'elle prend ses valeurs dans R . Cette variable X oscille de façon symétrique autour de son vecteur moyen (μ) avec la densité de probabilité suivante: [22]

$$g(\mu, \Sigma)(x) = \frac{1}{\sqrt{2\pi}^d \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad 2.11$$

2.6.1 Définition des deux Paramètres μ et Σ

2.6.1.1 Vecteur Moyen μ

Le vecteur moyen μ représente l'axe de symétrie de la distribution normale. Il contient les valeurs moyennes de chaque dimension. Il est symbolisé par l'équation 2.12 :

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_i \end{pmatrix} \quad i=1, \dots, d \quad 2.12$$

$$\mu_i = E(X_i) \quad 2.13$$

2.6.1.2 Matrice de Covariance Σ

En théorie des probabilités et en statistique, la matrice de covariance (nommée matrice de covariance et présentée par l'équation (2.14) d'une série de n variables aléatoires X_1, \dots, X_n est la matrice carrée dont l'élément de la ligne i et de la colonne j désigne la covariance c_{ij} permettant d'évaluer le sens de variation des variables X_i et X_j ainsi, de qualifier la dépendance des variables X_i et X_j . Les éléments de la diagonale sont les variances c_{ii} .

La matrice Σ permet de quantifier la variation de chaque variable par rapport à elle-même et aussi par rapport à chacune des autres. Cette matrice est symétrique, étant donné la propriété que $c_{ij} = c_{ji}$.

$$\Sigma = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} \quad 2.14$$

Le calcul de chaque élément de covariance est réalisée par l'équation suivantes :

$$c_{ij} = E \left((X_i - \mu_i)^T (X_j - \mu_j) \right) \quad 2.15$$

2.6.1.3 Contours des Points d'une Distributions Gaussienne

Tous les points d'un contour quelconque d'une distribution Gaussienne ont la même valeur de densité de probabilité.

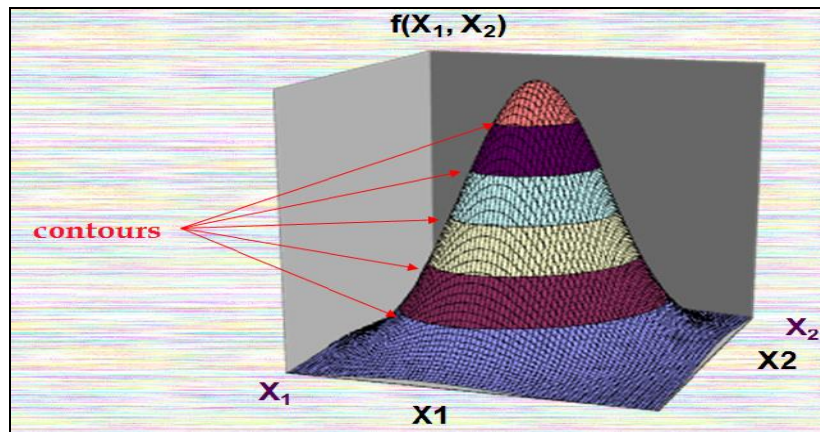


Figure (2.7): Contours d'une Distributions Gaussienne.

2.7 Vraisemblance selon un Modèle Gaussien

2.7.1 Vraisemblance

La vraisemblance d'un point x (dans un plan de deux dimensions) est la valeur de la fonction de densité de probabilité en ce point. Selon un modèle gaussien défini par un ensemble de paramètres $\Theta=(\mu,\Sigma)$, cette fonction est exprimée par l'équation 2.16 suivante :

$$P(x|\Theta) = \frac{1}{\sqrt{2\pi}^2 \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \quad 2.16$$

Pour simplifier et éviter les calculs des exponentiels il faut introduire la fonction logarithmique aux deux membres d'équation 2.17.

$$\log(P(x|\Theta)) = \log\left(\frac{1}{\sqrt{2\pi}^2 \sqrt{\det(\Sigma)}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}\right) \quad 2.17$$

Nous aurons l'équation correspondante au logarithme de vraisemblance suivante :

$$\log(P(x|\Theta)) = -\log(2\pi) - \frac{1}{2} \log(\det(\Sigma)) - \frac{1}{2} (x-\mu)^T \Sigma^{-1}(x-\mu) \quad 2.18$$

La fonction logarithmique est une fonction croissante monotone, ce qui implique que l'inégalité des vraisemblances reste valable pour les log-vraisemblances :

$$p(x|\Theta_1) > p(x|\Theta_2) \Leftrightarrow \log p(x|\Theta_1) > \log p(x|\Theta_2) \quad 2.19$$

Les log-vraisemblances peuvent être utilisées directement pour classifier des échantillons. Dans une perspective de classification, le calcul de logarithme de vraisemblance (équation 2.18) sera simplifié encore plus dans le cas où les deux matrices de covariances ($\Sigma_1 = \Sigma_2$) sont égales.

Les relations d'ordre restent valables si nous laissons tomber la division par 2 et les deux termes $-\log(2\pi)$ et $[\log(\det(\Sigma))]/2$. L'abandon du premier terme est justifié car c'est une constante indépendante. L'abandon du deuxième terme est démontré aussi dans le cas où les deux matrices de covariances des deux modèles Gaussien $N(\Theta_1)$ et $N(\Theta_2)$ sont indépendantes des classes (égales).

2.7.2 Vraisemblance Conjointe

La vraisemblance conjointe d'un ensemble X de N points indépendants distribués identiquement est le produit des N vraisemblances.

$$p(X|\Theta) = \prod_{i=1}^N p(x_i|\Theta) = \prod_{i=1}^N p(x_i|\mu, \Sigma) = \prod_{i=1}^N g(\mu, \Sigma)(x_i) \quad 2.20$$

On introduisant la fonction logarithmique aux deux membres de l'équation 2.20

Le produit se transforme en une sommation. [22]

$$p(X|\Theta) = \prod_{i=1}^N p(x_i|\Theta) \Leftrightarrow \log p(X|\Theta) = \sum_{i=1}^N \log p(x_i|\Theta) \quad 2.21$$

Les logarithmes des vraisemblances sont plus simples à calculer mais peuvent s'utiliser directement pour des tâches de classification.

Le logarithme de vraisemblance conjointe d'un échantillon est une sommation des logarithmes des vraisemblances de tous les points appartenant à l'échantillon.

2.8 Quelques Algorithmes de Classification

2.8.1 Algorithme des K-moyennes

L'algorithme des K-moyennes est un algorithme de classification non supervisée des données le plus simple et le plus utilisé.

L'idée principale de cet algorithme est de choisir aléatoirement un ensemble de K centres fixes a priori dans les régions de l'espace les plus peuplées et de chercher itérativement la partition optimale des données en k classes qui ne se chevauchent pas (k est un entier).

Chaque donnée est affectée au centre le plus proche par la règle de la distance minimale. Après l'affectation de toutes les données, pour chaque itération la moyenne de chaque groupe est recalculée, ces moyennes constituent les nouveaux centres représentants des groupes.

La procédure itérative amène progressivement chaque point (donnée) à sa classe finale (localement stable). L'algorithme est arrêté à l'état de convergence où aucune donnée ne change de groupe et les centres deviennent localement stables. Chaque classe contient donc les données qui sont les plus proches à son centre. (figures 2.8 et 2.9)

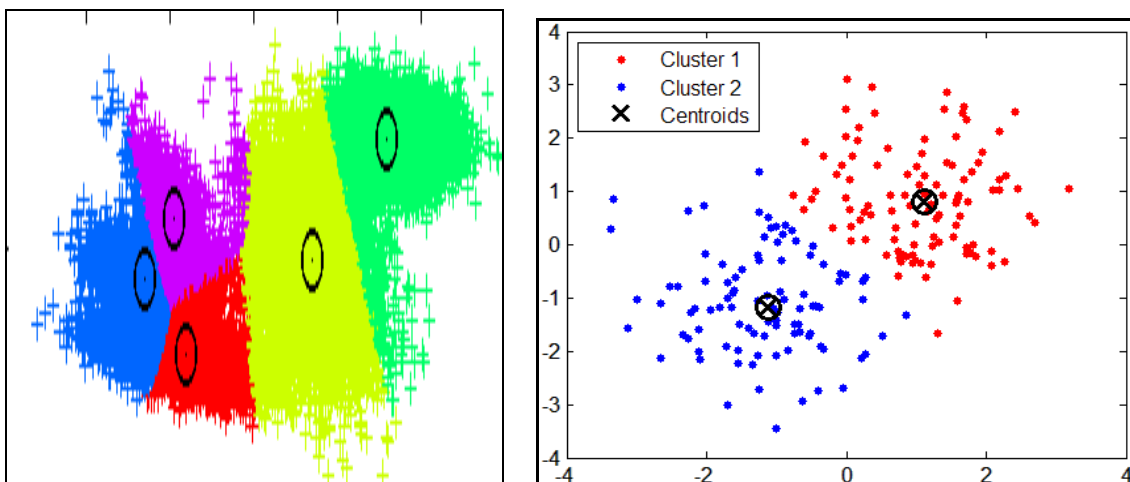


Figure (2.8): Deux Exemples de Classification par K_Means

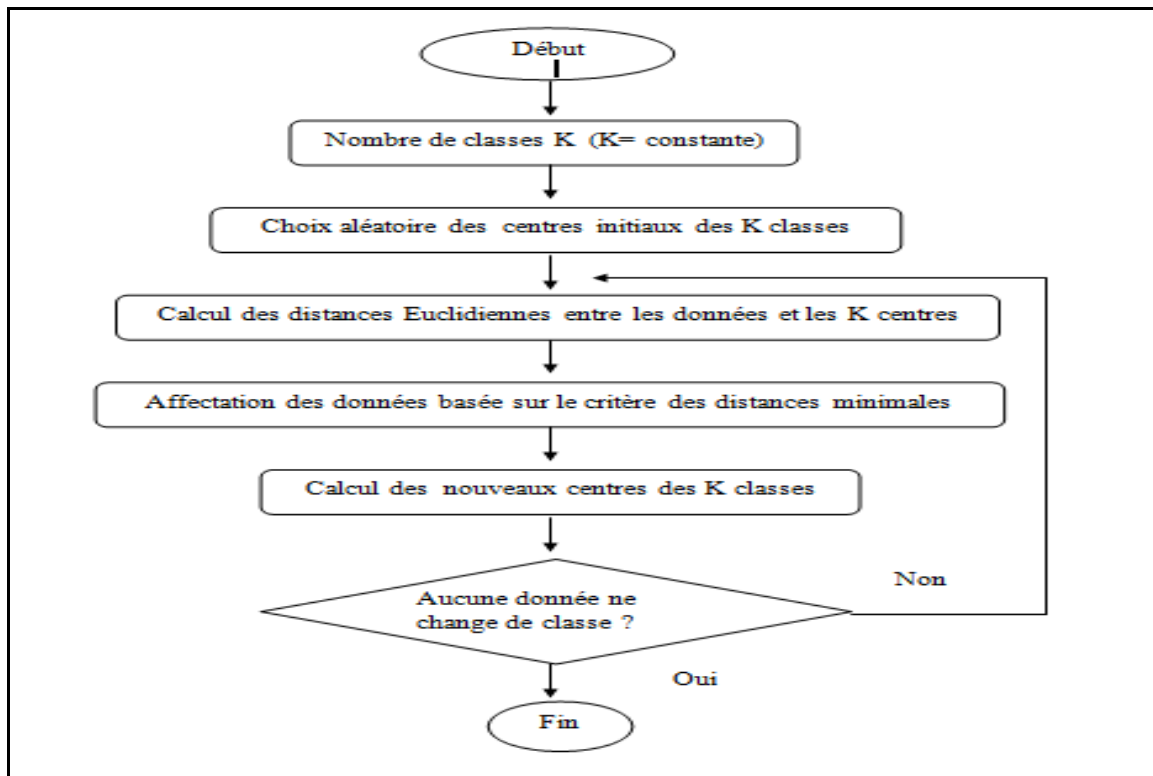


Figure (2.9): Organigramme de kmeans

Synthèse d'Algorithme :

L'algorithme de kmeans

- 1- Initialisation de nombre de classes k et le choix aléatoire** k centres initiaux $\mu_k, k=1, \dots, k$.
- 2- Calcul de distance euclidienne** carrée entre chaque point de données x_n et le centre μ_k ($k=1, \dots, k$) de la classe (k).
- 3- Affectation** de chaque point de données x_n au centre le plus proche μ_k de la classe C_k selon l'inégalité 2.9:
- 4- Calcul de k nouveaux représentants** qui sont égaux aux moyennes des k classes correspondantes.
- 5- Calcul de critère global (J)** représentant la distance euclidienne totale carrée entre les données et les centres auxquelles elles appartiennent. Ce critère (J) définit par l'équation 2.9 :

$$J = \sum_{k=1}^K \sum_{x_n \in C_k} d_k(x_n, \mu_k) \dots \dots \dots 2.10$$

- 6- Retour à l'étape 2** après chaque itération jusqu'à ce qu'il n'y ait plus de changement. A ce moment le critère (J) atteint un seuil minimal correspondant à la convergence.

2.8.2 Algorithme de Maximisation de Vraisemblance (EM)

L'algorithme EM est une approche générale qui fait un calcul itératif pour trouver des estimateurs du maximum de vraisemblance lorsque les données sont incomplètes. On l'appelle l'algorithme EM puisque chaque itération de l'algorithme consiste en une étape d'espérance et une étape de Maximisation.

L'étape d'estimation sert à avoir un ensemble Q de variables continues cachées, prenant des valeurs dans l'intervalle $[0; 1]$, qui donnent un étiquetage des données en indiquant dans quel proportion un point x_n appartient à la classe C_k . (C'est une classification molle, car un point peut appartenir, par exemple, à 60% à la classe 1 et à 40% à la classe 2).

$$P(C_k^{(\text{nouveau})} | x_n, \Theta^{(\text{nouveau})}) = \frac{P(C_k^{(\text{ancien})} | x_n, \Theta^{(\text{ancien})}) P(x_n | C_k^{(\text{ancien})}, \Theta^{(\text{ancien})})}{P(x_n | \Theta^{(\text{ancien})})} \quad 2.22$$

$$P(C_k^{(\text{nouveau})} | x_n, \Theta^{(\text{nouveau})}) = \frac{P(C_k^{(\text{ancien})} | \Theta^{(\text{ancien})}) P(x_n | \mu_k^{(\text{ancien})}, \Sigma_k^{(\text{ancien})})}{\sum_j P(C_j | \Theta^{(\text{ancien})}) P(x_n | \mu_j^{(\text{ancien})}, \Sigma_j^{(\text{ancien})})} \quad 2.23$$

L'étape de maximisation actualise les paramètres des k classes (moyennes, matrices de covariance) et les k probabilités a priori des classes comme suit :

$$\mu_k^{(\text{nouveau})} = \frac{\sum_{n=1}^N x_n P(C_k^{(\text{ancien})} | x_n, \Theta^{(\text{ancien})})}{\sum_{n=1}^N P(C_k^{(\text{ancien})} | x_n, \Theta^{(\text{ancien})})} \quad 2.24$$

$$\Sigma_k^{(\text{nouveau})} = \frac{\sum_{n=1}^N P(C_k^{(\text{ancien})} | x_n, \Theta^{(\text{ancien})}) (x_n - \mu_k^{(\text{nouveau})})(x_n - \mu_k^{(\text{nouveau})})^T}{\sum_{n=1}^N P(C_k^{(\text{ancien})} | x_n, \Theta^{(\text{ancien})})} \quad 2.25$$

Le calcul du critère global $\mathcal{L}(\Theta)$ se formalise comme suit :

$$\mathcal{L}(\Theta) = \log P((X) | \Theta) = \log \sum_Q P(X, Q | \Theta) \quad 2.26$$

$$\mathcal{L}(\Theta) = \log \sum_Q P(Q | X, \Theta) P(X | \Theta) \quad 2.27$$

$$\mathcal{L}(\Theta) = \log \sum_Q P(C_k | X, \Theta) P(X | \Theta) \quad 2.28$$

En introduisant l'inégalité de Jensen (équation 2.29) dans l'équation 2.28

$$\log \sum_j \lambda_j y_j \geq \sum_j \lambda_j \log y_j \quad \text{si} \quad \sum_j \lambda_j = 1 \quad 2.29$$

Nous obtenons les deux équations 2.30 et 2.31 suivantes :

$$\mathcal{L}(\theta) \approx \sum_{k=1}^K P(C_k|X, \theta) \log P(X|\theta) \quad 2.30$$

$$\mathcal{L}(\theta) \approx \sum_{k=1}^K \sum_{n=1}^N P(C_k|x_n, \theta) \log P(x_n|\theta) \quad 2.31$$

Synthèse de l'algorithme :

1-Etape d'Initialisation des k modèles Gaussiens $N_k(\theta_k)$ ($\theta_k = (\mu_k, \Sigma_k)$) et des k probabilités a priori des classes ($P(C_k) = 1/k$).

2- Etape d'Estimation des calculs des probabilités à posteriori pour chaque point des données x_n par rapport à chaque classe C_k .

$$P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})}) = \frac{P(C_k^{(\text{ancien})} | \theta^{(\text{ancien})}) P(x_n | \mu_k^{(\text{ancien})}, \Sigma_k^{(\text{ancien})})}{\sum_j P(C_j | \theta^{(\text{ancien})}) P(x_n | \mu_j^{(\text{ancien})}, \Sigma_j^{(\text{ancien})})} \quad 2.32$$

3- Etape de Maximisation actualise les paramètres des nouveaux représentants des k classes qui sont les k moyennes, k matrices de covariance et k probabilités à priori comme suit :

$$\mu_k^{(\text{nouveau})} = \frac{\sum_{n=1}^N x_n P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})})}{\sum_{n=1}^N P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})})} \quad 2.33$$

$$\Sigma_k^{(\text{nouveau})} = \frac{\sum_{n=1}^N P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})}) (x_n - \mu_k^{(\text{nouveau})})(x_n - \mu_k^{(\text{nouveau})})}{\sum_{n=1}^N P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})})} \quad 2.34$$

$$P(C_k^{(\text{nouveau})} | \theta^{(\text{nouveau})}) = \frac{1}{N} \sum_{n=1}^N P(C_k^{(\text{nouveau})} | x_n, \theta^{(\text{nouveau})}) \quad 2.35$$

Toutes les données participent à la mise à jour des trois paramètres. Leur participation est proportionnelle aux valeurs des probabilités à posteriori. (équations de 2.33 à 2.35)

4- Calcul de critère global $\mathcal{L}(\theta)$ à chaque nouvelle itération

$$\mathcal{L}(\theta) \approx \sum_{k=1}^K \sum_{n=1}^N P(C_k | x_n, \theta) \log P(x_n | \theta) \quad 3.36$$

Le critère $\mathcal{L}(\theta)$ final représente la limite inférieure pour la vraisemblance conjointe de l'ensemble des données par rapport à tous les modèles.

5- Retour à l'étape 2 jusqu'à ce que la valeur de critère global sera inférieure ou égale à un seuil choisi. Ce dernier critère est maximisé localement par l'algorithme.

2.8.3 Algorithme de Viterbi-EM

Même principe de fonctionnement que l'algorithme EM sauf dans les calculs des trois K paramètres des K classes (moyenne, matrice de covariance et probabilité à priori). Il contribue à avoir un ensemble Q de variables booléennes cachées qui donnent un étiquetage des données en prenant la valeur 1 (appartient) ou 0 (n'appartient pas) pour chaque classe C_k et chaque point x_n . La valeur de Q qui maximise $p(X;Q/\theta)$ nous indique précisément quel est le modèle le plus probable pour chaque point de l'ensemble X des données d'entraînement.

Donc, chaque point des données est assigné au cluster le plus probable pour lui C_k .

Synthèse de l'Algorithme :

1-Etape d'Initialisation des k : modèles Gaussiens $N_k(\theta_k)$ ($\theta_k = (\mu_k, \Sigma_k)$) et des k probabilités a priori des classes ($P(C_k) = 1/K$).

2- Etape d'Estimation : des calculs des probabilités à posteriori pour chaque point des données x_n par rapport à chaque classe C_k .

3- Etape de Maximisation: actualise les paramètres des nouveaux représentants des k classes qui sont les k moyennes, k matrices de covariance et k probabilités à priori comme suit :

$$\mu_k^{(nouveau)} = \text{moyenne des points appartenant à } C_k^{(ancien)} \quad 2.37$$

$$\Sigma_k^{(nouveau)} = \text{variance des points appartenant à } C_k^{(ancien)} \quad 2.38$$

$$P(C_k^{(nouveau)} | \Theta^{(nouveau)}) = \frac{\text{nombre de points d'entraînement appartenant à } C_k^{(ancien)}}{\text{nombre totale de points d'entraînement}} \quad 2.39$$

4- Calcul de critère global : $\mathcal{L}(\Theta)$ à chaque nouvelle itération. Ce critère représente la vraisemblance conjointe des données par rapport aux modèles auxquels elles appartiennent.

$$\mathcal{L}(\Theta) = \sum_{\mathbf{x}} P(\mathbf{x} | \Theta) \quad 2.40$$

$$\mathcal{L}(\Theta) = \sum_{\varphi} \sum_{\mathbf{x}} P(\mathbf{x}, \varphi | \Theta) \quad 2.41$$

$$\mathcal{L}(\Theta) = \sum_{k=1}^K \sum_{x_n \in C_k} \log P(x_n | \Theta_k) \quad 2.43$$

5- Retour à l'étape 2 : jusqu'à ce que : il n'y ait plus de changement. Ce critère est maximisé localement par l'algorithme.

2.9 Conclusion

Dans ce chapitre nous avons présenté quelques notions de base sur les probabilités et les modèles Gaussiennes. Nous avons détaillé les trois algorithmes de classification utilisés dans la partie expérimentale.

Dans le chapitre suivant nous allons exposer les expériences réalisées ainsi que les résultats obtenus et leurs interprétations.

3.1 Introduction

Déférentes techniques supervisées et non supervisées serrent aux classifications des attributs selon des procédures itératives. Les phases et les résultats obtenus à la convergence de ces techniques sont dépendantes aux leurs phases initialisations. Nous allons essayer de détailler quelques notions de classification vu au chapitre précédent et ceci en les appliquant sur une base de donnée qui sera définie dans ce qui suit

3.2 Représentation de langage MATLAB [28]

MATLAB est une application scientifique interactive orientée au calcul vectoriel et matriciel avec une puissante librairie de visualisation. Vous pouvez résoudre des problèmes de calcul très complexes d'une façon simple et rapide comparée aux langages de programmation traditionnels du type C. Le nom MATLAB est dérivé de l'anglais MATrix LABORatory.

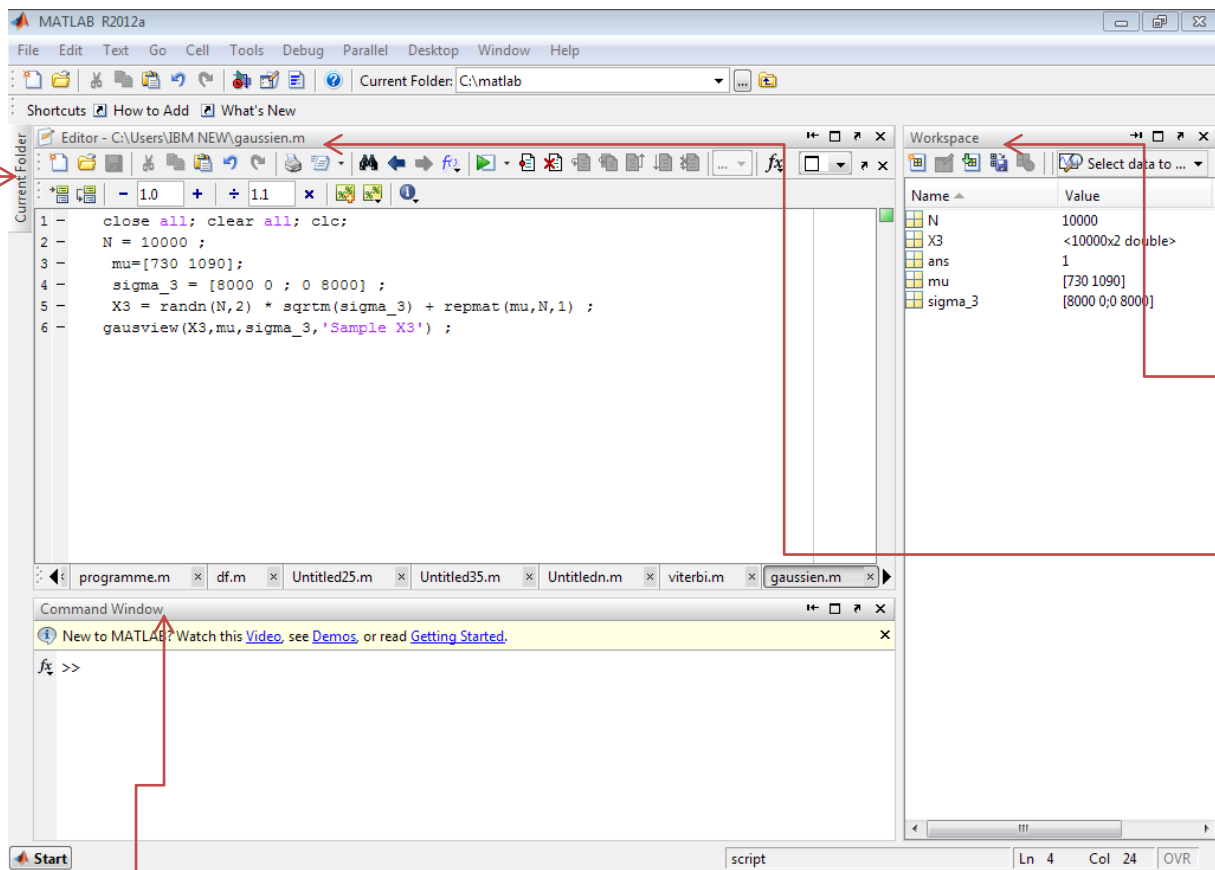


figure (3.0) : représentation des fenêtres de langage MATAB

- 1 → la fenêtre « Command Window » c'est pour voir les résultats
- 2 → la fenêtre « Current folder » pour insérer la base de données dans notre cas si le fichier "vowels.mat"
- 3 → la fenêtre « Workspace »
- 4 → la fenêtre « Editor » cette fenêtre pour écrire le programme

3.2.1 Programmes en Matlab : les fichiers M [29]

Un programme est une suite d'instructions enregistrées dans un fichier.

En Matlab, ces fichiers sont appelés fichiers M et ont tous l'extension '*.m'.

Un exemple de nom de fichier M est donc : 'toto'. Ces fichiers peuvent être créés avec le logiciel Matlab en sélectionnant "Nouveau fichier" dans le menu déroulant "Fichier". Ne pas oublier d'enregistrer le fichier après toute modification ! L'exécution du programme se fait en tapant le nom du fichier sans l'extension *.m dans la fenêtre de commande.

3.3 Base de données utilisée

Comme cité à l'introduction nous avons pris une base de données qui a été réalisée à l'université de Tugraz << Autriche >>. Cette dernière représente les pitch et le premier des voyelles suivantes /a/, /e/, /i/, /o/ et /y/. Elles sont composées de plusieurs occurrences tels que

- Le phonème /a/ est représenté par 5000 occurrences
- Le phonème /e/ est représenté par 6000 occurrences
- Le phonème /i/ est représenté par 5000 occurrences
- Le phonème /o/ est représenté par 3000 occurrences
- Le phonème /y/ est représenté par 1000 occurrences

L'ensemble de la base de données sera représenté par 20000 occurrences.

3.4 Nos Expériences

3.4.1 Expérience N°1 : sur la fonction de densité de probabilité (FDP)

Nous avons généré le nuage de points et tracé les contours d'un ensemble aléatoire \mathbf{X} de N points $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$; de 2 dimensions avec $N = 10000$, provenant d'un processus gaussien bidimensionnel ayant une moyenne :

$$\mu = \begin{pmatrix} 730 \\ 1090 \end{pmatrix}$$

et une matrice de covariance

1- 8000 pour les deux dimensions de chaque donnée de l'échantillon X_1 :

$$\Sigma_1 = \begin{bmatrix} 8000 & 0 \\ 0 & 8000 \end{bmatrix}$$

2- Exprimé comme une matrice de covariance diagonale (échantillon X_2) :

$$\Sigma_2 = \begin{bmatrix} 8000 & 0 \\ 0 & 18500 \end{bmatrix}$$

3- Exprimé comme une matrice de covariance pleine (échantillon X_3) :

$$\Sigma_3 = \begin{bmatrix} 8000 & 8400 \\ 8400 & 18500 \end{bmatrix}$$

L'exécution du programme a permis de générer trois nuages de points X_1 , X_2 , X_3 ont le même vecteur moyen μ et trois matrices de covariance précédentes Σ_1 , Σ_2 , Σ_3 , et de tracer leurs contours des points.

Nos exposons par la suite les figures représentant les nuages et les contours de points X_1 , X_2 , X_3 .

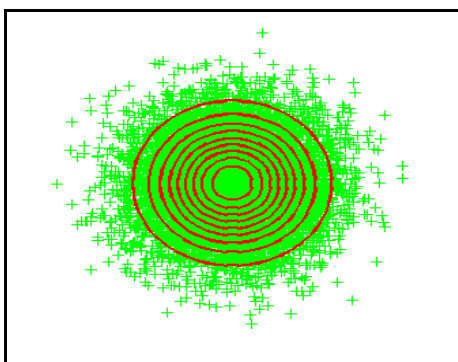


Figure (3.1) : Contours de Distribution des Points X_1

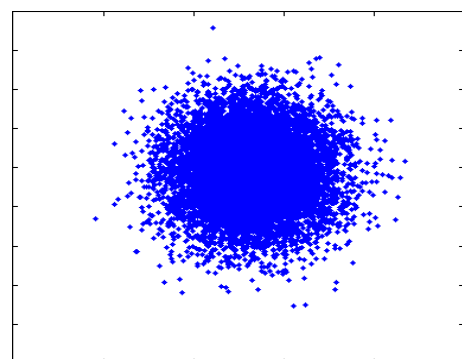


Figure (3.2) : Nuage de Points X_1

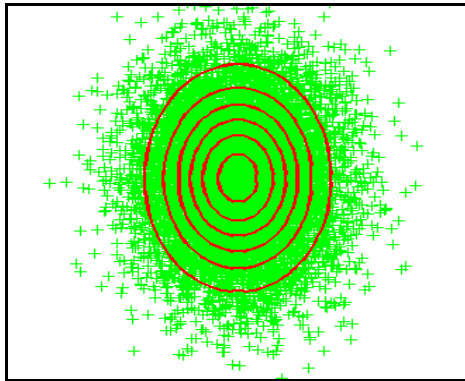


Figure (3.3) : Contours de Distribution des points X_2

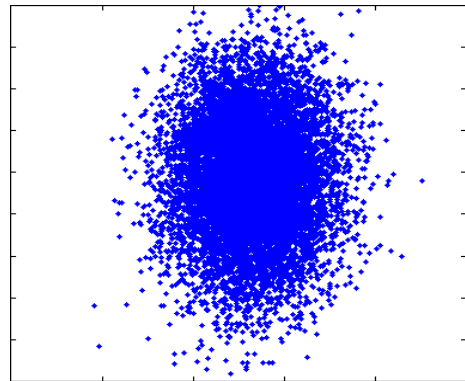


Figure (3.4): Nuage de Points X_2

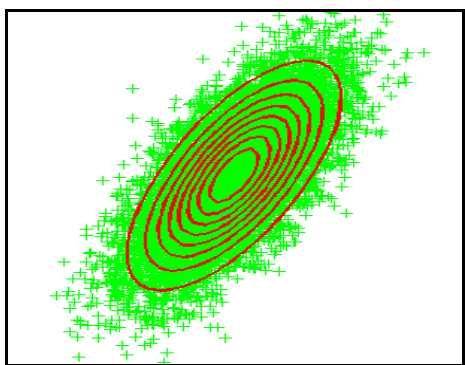


Figure (3.5) : Contours de Distribution des points X_3

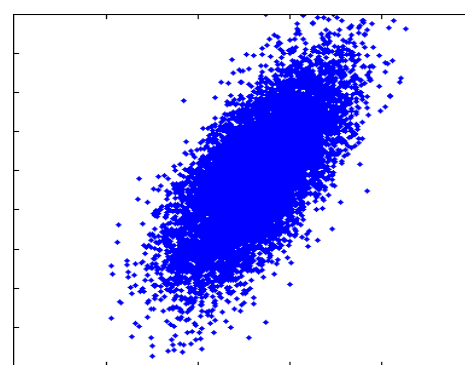


Figure (3.6) : Nuage de Points X_3

D'après les figures 3.1 à 3.6 nous observons que les vecteurs des données forment des contours et des nuages de points :

- sphériques lorsque les variances des deux coordonnées de vecteurs de données sont égales et leurs covariances sont nulles ($c_{11} = c_{22}$ et $c_{12} = c_{21}=0$);
- elliptiques lorsque les variances des deux coordonnées de vecteurs de données sont différentes, par contre leurs covariances sont égales à zéro ($c_{11} \neq c_{22}$ et $c_{12} = c_{21}=0$). L'axe principal des ellipses correspond à la coordonnée qui a la plus grande variance et l'axe secondaire correspond à la deuxième coordonnée de plus faible variance. Les deux axes principal et secondaire des ellipses sont parallèles aux axes des abscisses et des ordonnées;

- elliptiques, mais les deux axes des ellipses ne sont pas parallèles aux deux axes d'abscisses et d'ordonnées par l'effet de la corrélation positif ou négatif entre les deux coordonnées ($c_{12} = c_{21} \neq 0$).

Interprétation :

Le nuage de points et les contours de la fonction de densité de probabilité correspondant à Σ_1 sont circulaires, parce que dans ce cas la première et la deuxième dimension des vecteurs sont indépendantes. En fait, elles ont une covariance nulle (l'équation 3.1):

$$E[(x_{d1} - \mu_{d1})^T (x_{d2} - \mu_{d2})] \quad 3.1$$

Elles sont orthogonales dans le sens statistique, ce qui se traduit par un sens géométrique (l'espérance est un produit scalaire de variables aléatoires ; un produit scalaire nul signifie l'orthogonalité). Intuitivement, une covariance nulle signifie que les deux dimensions ne partagent pas d'information : elles peuvent évoluer indépendamment de façon gaussienne le long de leurs axes respectifs. D'autre part, la variance est la même dans les deux dimensions, ce qui indique une dispersion des données équivalente le long des deux axes. D'où la masse circulaire de points de données et le nom de processus sphérique.

Le nuage de points et les contours de la fonction de densité de probabilité correspondant à Σ_2 sont elliptiques, avec les axes parallèles aux axes des abscisses et des ordonnées.

Ceci s'explique par le fait que la première et la deuxième dimension sont indépendantes (leur covariance est nulle), mais cette fois la variance est différente selon la dimension.

Pour Σ_3 , la covariance des deux dimensions n'est pas nulle, ce qui fait que les axes principaux des ellipses ne sont pas alignés avec l'axe des abscisses et l'axe des ordonnées.

x_{d1} , x_{d2} : Sont les deux composantes (coordonnées) du vecteur des données x .

μ_{d1} , μ_{d2} : Sont les deux composantes (coordonnées) du vecteur moyen.

3.4.2 Expérience N°2 : moyenne et variance d'un échantillon

Nous avons choisi l'ensemble de points \mathbf{X}_3 distribué selon le model gaussien $\mathcal{N}(\mu, \Sigma_3)$ de vecteur moyen μ et Σ_3 . Nous voulons estimer ces paramètres par trois différents échantillons appartenant à l'ensemble \mathbf{X}_3 (de valeur $N_1=10000$, $N_2=1000$, $N_3=100$) par le biais de deux estimateurs d'équations (3.3 et 3.4). Par la suite nous avons comparé et mesuré la distance euclidienne qui sépare les 2 vecteurs moyen (estimé et original) ainsi, nous avons comparé et mesuré la similarité entre les deux matrices (estimée et originale) par le calcul de la norme matricielle selon les deux équations d'erreurs (de 3.2 à 3.5) :

$$Err_{\mu} = \|\mu - \hat{\mu}\|^{\frac{1}{2}} \quad 3.2$$

$$Err_{\mu} = (\mu - \hat{\mu}_k)^T (\mu - \hat{\mu}_k)^{\frac{1}{2}} \quad 3.3$$

$$Err_{\Sigma_3} = \|\Sigma_3 - \hat{\Sigma}_3\|^2 \quad 3.4$$

$$Err_{\mu} = (\mu - \hat{\mu}_k)^T (\mu - \hat{\mu}_k) \quad 3.5$$

Nos exposons nos résultats dans le tableau 3.1.

Nombre de points	Err_{μ}	Err_{Σ_3}
$N_1=10000$	0.5024	524.8639
$N_2=1000$	16.7481	1650.2
$N_3=100$	30.6660	4381.7

Tableau (3.1) : Erreurs sur les Paramètres en Fonction de Nombre de Points.

Rappelons que :

- μ et Σ_3 : Sont les deux paramètres originaux du model gaussien $\mathcal{N}(\mu, \Sigma_3)$.
- $\hat{\mu}$ et $\hat{\Sigma}$: Sont les deux paramètres estimés du model gaussien $\mathcal{N}(\mu, \Sigma_3)$.
- Err_{μ} et Err_{Σ_3} : désignent les erreurs d'estimation de vecteur moyen et de la matrice de covariance.

D'après le tableau (3.1) nous remarquons que :

- ✓ Plus le nombre de points utilisé est grand, plus l'erreur d'estimation des deux paramètres diminue.
- ✓ L'erreur d'estimation de la matrice de covariance est très élevée que celle de vecteur moyen.
- ✓ Les erreurs d'estimation des paramètres changent après chaque nouvelle exécution du programme.

Interprétation :

L'exactitude des paramètres exige la totalité des points. Plus il y a de points, meilleures sont les estimations. Une estimation précise de vecteur moyen demande moins de points que celle d'une matrice de covariance, parce que le calcul d'estimation de vecteur moyen est plus simple par rapport au calcul d'estimation d'une matrice de covariance ce qui explique la différence des erreurs existante entre les deux paramètres.

Le changement des erreurs après une nouvelle exécution de notre programme est dû au fait que l'ensemble de points \mathbf{X}_3 change aléatoirement après chaque nouvelle exécution.

➤ La dispersion de nuage de points et leurs contours dépendent de la matrice de covariance ainsi que de l'axe de symétrie le vecteur moyen.

En général, pour toute technique de classification des formes basée sur les données, il est très important d'avoir suffisamment d'exemples d'entraînement pour pouvoir estimer des modèles précis des données.

3.4.3 Expérience N°3 : Vraisemblance d'un échantillon par rapport à un modèle gaussien

Soient les 4 modèles gaussiens $\{\mathcal{N}_1(\mu_1, \Sigma_1), \mathcal{N}_2(\mu_2, \Sigma_2), \mathcal{N}_3(\mu_3, \Sigma_3)$ et $\mathcal{N}_4(\mu_4, \Sigma_4)\}$.

$$\mathcal{N}_1 : \Theta_1 = \left(\begin{bmatrix} 730 \\ 1090 \end{bmatrix}, \begin{bmatrix} 8000 & 0 \\ 0 & 8000 \end{bmatrix} \right) \mathcal{N}_2 : \Theta_2 = \left(\begin{bmatrix} 730 \\ 1090 \end{bmatrix}, \begin{bmatrix} 8000 & 0 \\ 0 & 18500 \end{bmatrix} \right)$$

$$\mathcal{N}_3 : \Theta_3 = \left(\begin{bmatrix} 730 \\ 1090 \end{bmatrix}, \begin{bmatrix} 8000 & 8400 \\ 8400 & 18500 \end{bmatrix} \right) \mathcal{N}_4 : \Theta_4 = \left(\begin{bmatrix} 270 \\ 1690 \end{bmatrix}, \begin{bmatrix} 8000 & 8400 \\ 8400 & 18500 \end{bmatrix} \right)$$

X_3 est un échantillon de 10000 points son moyen μ_3 et son matrice de covariance Σ_3 .

Nous avons calculé les logarithmes des vraisemblances conjointes de l'échantillon X_3 et les quatre modèles Gaussien précédents. Les résultats obtenus sont présentés dans le tableau 3.2.

$\log P(x_3 \Theta_1)$	$\log P(x_3 \Theta_2)$	$\log P(x_3 \Theta_3)$	$\log P(x_3 \Theta_4)$
-1.2495e+005	-1.2244e+005	-1.1915e+005	-8.5880e+005

Tableau (3.2) : Logarithme des Vraisemblances Conjointes Obtenues.

Nous avons représenté graphiquement chaque modèle Gaussien en compagnie du nuage de point X_3 dans un plan et dans l'espace. Nous exposons leurs figures (de 3.7 à 3.14).

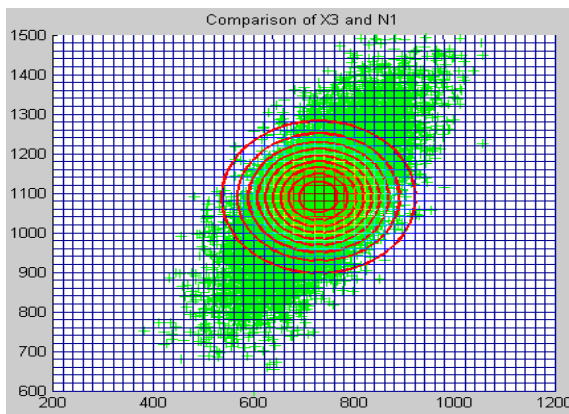


Figure 3.7 : X_3 et le Modèle \mathcal{N}_1 dans \mathcal{R}^2

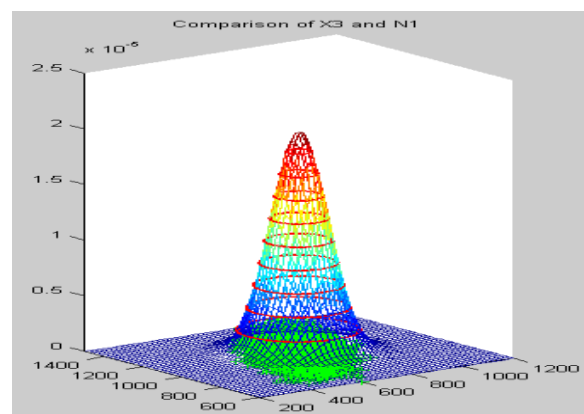


Figure 3.8 : X_3 le Modèle \mathcal{N}_1 dans \mathcal{R}^3

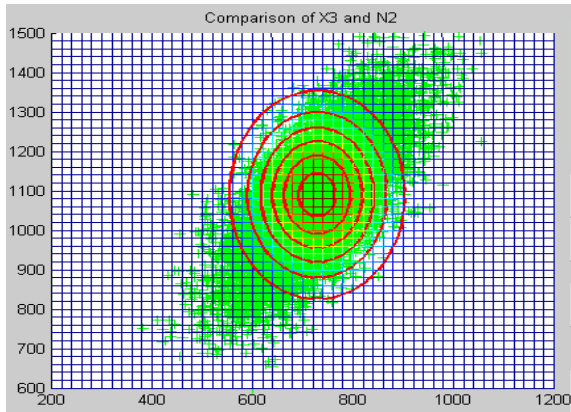


Figure (3.9) : X_3 et le Modèle N_2 dans R^2

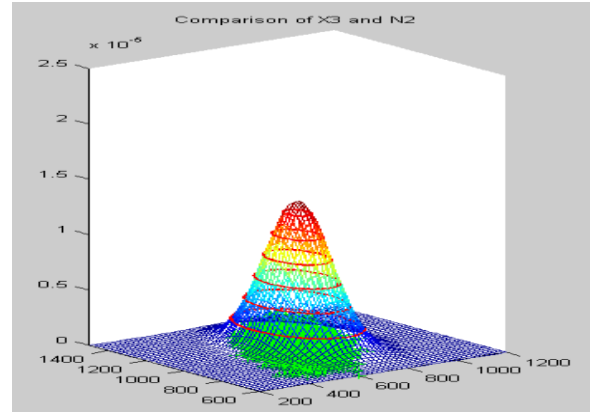


Figure (3.10) : X_3 et le Modèle N_2 dans R^3

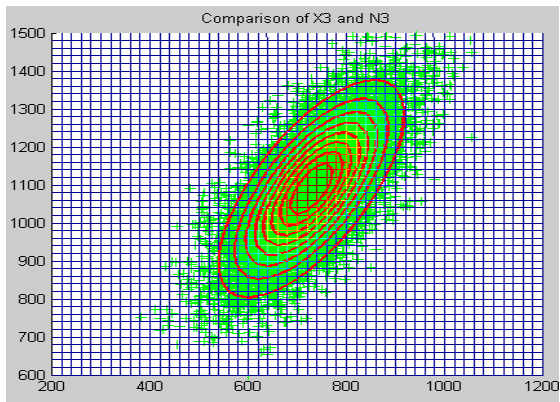


Figure (3.11) : X_3 et le Modèle N_3 dans R^2

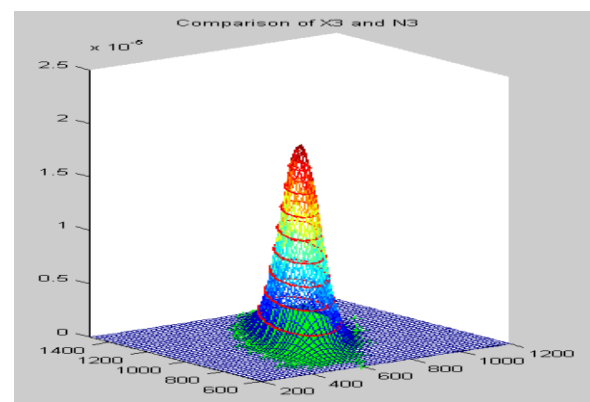


Figure (3.12) : X_3 et le Modèle N_3 dans R^3

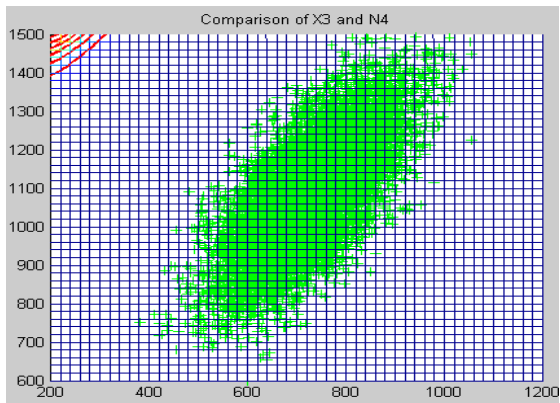


Figure (3.13) : X_3 et le Modèle N_4 dans R^2

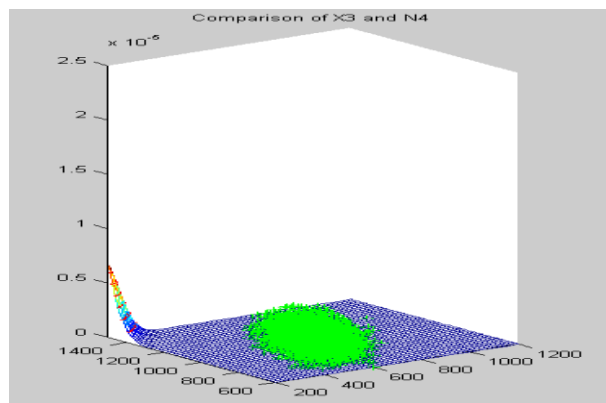


Figure (3.14) : X_3 et le Modèle N_4 dans R^3

Nous apercevons que :

- ✓ Le tableau 3.2 a permis de montrer que la log-vraisemblance conjointe est maximale pour le modèle Gaussien $\mathcal{N}_3(\Theta_3)$ et minimale pour le modèle Gaussien $\mathcal{N}_4(\Theta_4)$.
- ✓ Les figures de 3.12 à 3.14 déclarent que le modèle Gaussien $\mathcal{N}_3(\Theta_3)$ produit la vraisemblance conjointe la plus grande pour les données X_3 . Elles montrent aussi que le modèle Gaussien $\mathcal{N}_4(\Theta_4)$ exprime mal l'échantillon étudié.
- ✓ Les valeurs des log-vraisemblances conjointes pour les quatre modèles Gaussiens changent leurs valeurs après chaque nouvelle exécution de programme.

Commentaire :

Les résultats acquis assurent que le modèle Gaussien $\mathcal{N}_3(\Theta_3)$ produit la vraisemblance conjointe la plus grande pour les données X_3 . Ce qui nous permet de dire que les données X_3 ont plus de chance d'avoir été générées par ce modèle $\mathcal{N}_3(\Theta_3)$ que par d'autres modèles. Nous concluons que le modèle $\mathcal{N}_3(\Theta_3)$ a la plus grande capacité de représenter précisément les données, ainsi que le modèle Gaussien $\mathcal{N}_4(\Theta_4)$ est le plus mauvais représentant des données.

Le changement des valeurs des log-vraisemblances conjointes après chaque nouvelle exécution de programme est dû au fait que l'échantillon X_3 est généré aléatoirement.

En rapprochant les résultats de cette expérience et les paramètres (moyenne, matrice de covariance) de l'échantillon de données X_3 nous déduisons leurs compatibilités.

3.4.4 Expérience N° 4 : Probabilité de classes a priori

Notre base de données est nommée 'vowels.mat' est composée de paramètres de parole bidimensionnels simulés, sous forme de paires artificielles des fréquences des deux premiers formants : [F1; F2]. Ces valeurs artificielles extraits de plusieurs réalisations des voyelles /a/, /e/, /i/, /o/ et /y/. Les valeurs sont groupées dans des matrices de taille $N \times 2$, ou chacune des N lignes et un exemple d'entraînement qui contient les deux fréquences pour chaque voyelle.

Nous supposons que la base de données couvre suffisamment un langage imaginaire composée uniquement des voyelles.

Nous voulons calculer la probabilité $P(C_{\text{voy}})$ de chaque classe de $C_{\text{voy}} \in \{/a/; /e/; /i/; /o/; /y/\}$ sachant que le nombre de chaque voyelle est de 5000,6000, 5000, 3000, 1000 successivement.

Nous exposons les résultats acquis dans le tableau 3.3 suivant :

Phonèmes	probabilité a priori
/a/	0.25
/e/	0.3
/i/	0.25
/o/	0.15
/y/	0.05

Tableau (3.3) : Probabilités a Priori des Voyelles.

D'après le tableau 3.3 nous voyons que :

- ✓ Les valeurs des probabilités a priori sont égales pour les deux phonèmes /a/ et /i/. Elles sont différentes pour les autres.

Commentaire :

Le phonème /e/ est le plus fréquent dans ce langage imaginaire mais le phonème /y/ est le moins utilisé.

3.4.5 Expérience N° 5: Modélisation gaussienne de classes

La modélisation Gaussienne est réalisée par l'estimation des deux paramètres de chaque classe des voyelles. Nous représentons graphiquement dans le plan bidimensionnel le nuage de points des voyelles et leurs vecteurs moyens.

Phonèmes	Vecteur Moyen	Matrice de covariance
/a/	[729.47 1087.7]	$\begin{bmatrix} 1.6321e + 03 & 5.2020e + 03 \\ 5.2020e + 03 & 5.2276e + 04 \end{bmatrix}$
/e/	[531.57 1844.9]	$\begin{bmatrix} 1.4494e + 04 & 7.2817e + 03 \\ 7.2817e + 03 & 3.6447e + 04 \end{bmatrix}$
/i/	[270.84 2292.2]	$\begin{bmatrix} 2.6027e + 03 & 1.1376e + 03 \\ 1.1376e + 03 & 3.5310e + 04 \end{bmatrix}$
/o/	[569.38 842.69]	$\begin{bmatrix} 1.9774e + 03 & 3.5698e + 03 \\ 3.5698e + 03 & 2.0690e + 04 \end{bmatrix}$
/y/	[441.58 1022.8]	$\begin{bmatrix} 7.8942e + 03 & 8.7702e + 03 \\ 8.7702e + 03 & 1.9333e + 04 \end{bmatrix}$

Tableau (3.4) : Paramètres Gaussiens des Phonèmes

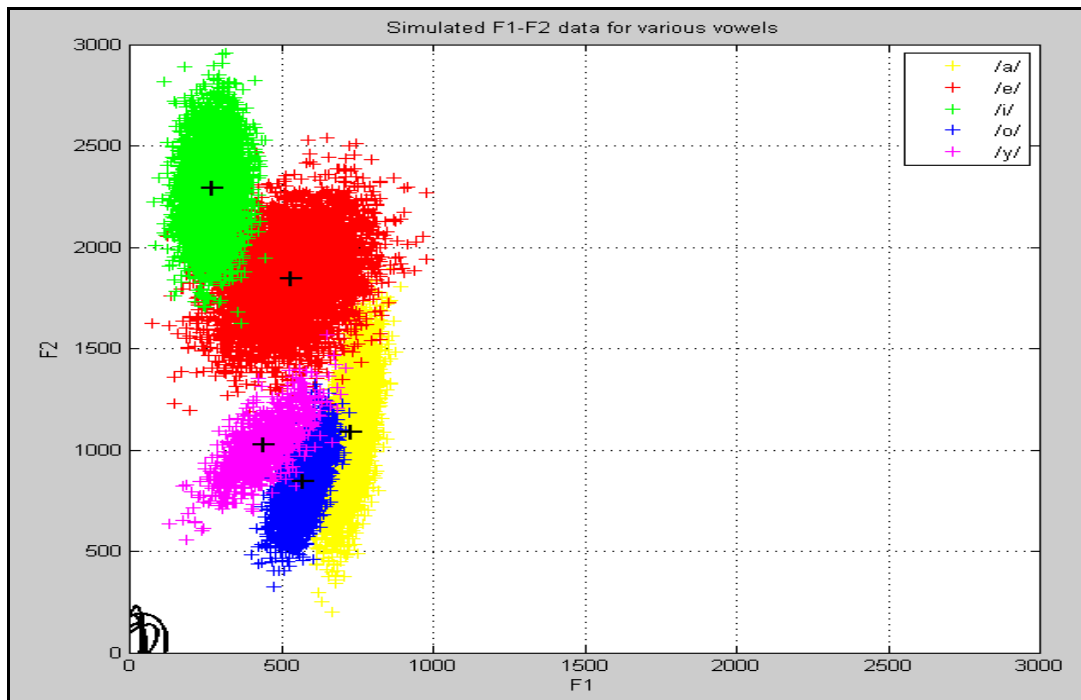


Figure (3.15) : Nuage des Voyelles et leurs Vecteurs Moyens.

D'après le tableau 3.4 et la figure 3.15 nous constatons que :

les valeurs des deux coordonnées des vecteurs moyens calculés des voyelles sont les mêmes avec celles représentés graphiquement dans la figure 3.5.

Commentaire :

Le vecteur moyen est un paramètre de position du centre de gravité d'un ensemble de points.

3.4.6 Expérience N° 6: Classification bayésienne

Maintenant, nous avons modélisé chaque classe de voyelle avec une fonction de densité de probabilité Gaussienne (en calculant les moyennes et les matrice de covariances), nous connaissons les probabilités a priori $P(C_{\text{voy}})$ des classes dans le langage imaginaire, et nous supposons que les paramètres de parole (par opposition aux classes) sont équiprobables.

Nous désirons calculer les probabilités a posteriori pour designer la classe la plus probable pour chaque couple des formants.

I	$X_i=[F1, F2]^T$	$\log(pC_a/X_i)$	$\log(pC_e/X_i)$	$\log(pC_i/X_i)$	$\log(pC_o/X_i)$	$\log(pC_y/X_i)$	Classe la plus
1	$[400,1800]^T$	-88.9257	-13.6356	-19.9132	-75.5643	-49.3573	/e/
2	$[400,1000]^T$	-58.3930	-22.8786	-41.7136	-27.0938	-14.0427	/y/
3	$[530,1000]^T$	-28.5610	-23.8978	-53.6589	-14.5352	-15.1644	/o/
4	$[600,1300]^T$	-22.7700	-18.3107	-51.7360	-18.2221	-16.0276	/y/
5	$[670,1300]^T$	-15.5094	-19.4504	-62.5418	-17.5389	-17.2378	/a/
6	$[420,2500]^T$	-122.0943	-21.1723	-16.9512	-148.2887	-131.4180	/i/

Tableau (3.5) : Probabilités a Posteriori Réelles des Voyelles.

Nous découvrons que la probabilité à posteriori est maximale pour la voyelle

- ✓ /e/ avec le premier couple des formants [400,1800].
- ✓ /y/ avec le deuxième et le quatrième couple des formants [400,1000], [600,1300].
- ✓ /o/ avec le troisième couple des formants [530,1000].
- ✓ /a/ avec le cinquième couple des formants [670,1300].
- ✓ /i/ avec le sixième couple des formants [420,2500].

Interprétation :

La sélection des voyelles correspondantes aux différents couples des formants est basée sur la classification Bayésienne des logarithmes des probabilités a posteriori. La voyelle correspondante à la valeur maximale de la probabilité a posteriori acquise assure que le couple des formants correspondant appartient a cette voyelle.

3.5 Entraînement non Supervisé

3.5.1 Expérience sur l'Algorithme du K_means N° 7

Cet algorithme admet en entrée une base de données (des voyelles) ainsi que le nombre de classes d'appartenance et retourne en sortie les partitions des cinq classes et leurs centres correspondants.

Voici les figures obtenus.

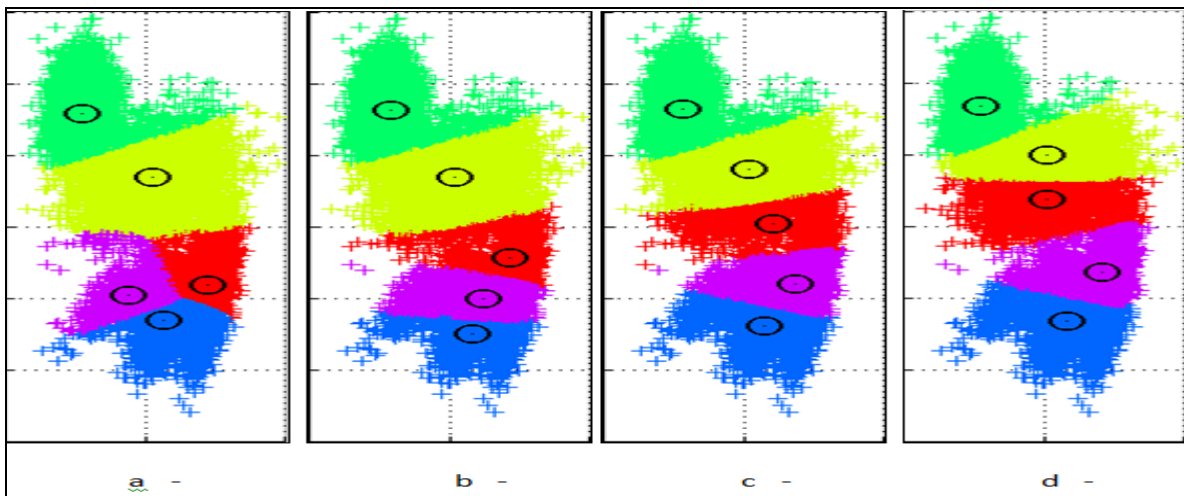


Figure (3.16) : Quatre Phases d'Exécution de l'Algorithme K_means. Dans (a) représente le commencement (0 itérations). Dans (b) après 25 itérations. Dans (c) après 50 itérations. Dans (d) la convergence (après 77 itérations).

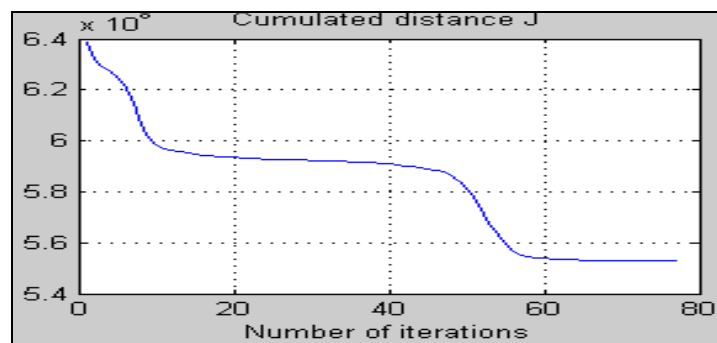


Figure (3.17) : Critère Global J en Fonction des Nombres des Itérations (K_means).

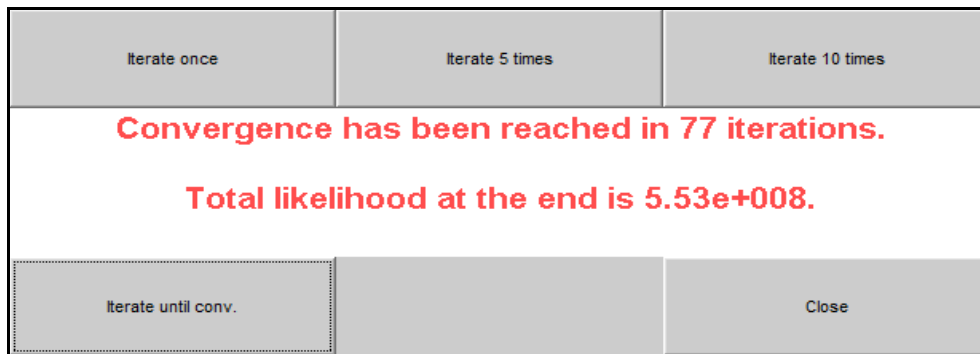


Figure (3.18) : Résultats de K_means à la Convergence.

D'après les figures de 3.16 à 3.18 nous observons que :

- ✓ L'affectation des données aux k classes change et la distance totale carrée (J) diminue après chaque itération ce qui induit le changement des positions des k nouveaux centres.
- ✓ L'arrêt de l'algorithme est obtenu à la convergence après 77 itérations, lorsqu'aucune donnée ne change de classe et le critère de distance totale J est minimisé. Sa valeur à la convergence locale est égale à $5.53 \cdot 10^8$.
- ✓ Le nombre d'itérations nécessaire pour réaliser la convergence change après chaque nouvelle exécution de l'algorithme.
- ✓ Les classes ne se chevauchent pas et elles sont bien séparées.

Discussion :

Le k-means est une technique de classification non supervisée. Les centres des groupes sont positionnés par une procédure itérative qui les amène progressivement dans leurs positions finales localement stables en minimisant le critère de distance totale carrée qu'atteint un seuil minimal à la convergence ce qui conduit à l'arrêt de l'algorithme. L'appartenance des données à leurs classes est fixée localement.

Le nombre d'itérations nécessaire pour assurer la convergence change après une nouvelle exécution de l'algorithme à cause du choix aléatoire des centres au début de l'algorithme. Vu que, le choix aléatoire des centres des k classes (les premiers centres des k

classes ne sont pas les mêmes avec celles correspondants aux exécutions précédentes) au début, Le nombre d'itérations nécessaire pour la convergence change après chaque nouvelle exécution de cet algorithme. Donc le résultat final dépend de l'initialisation de l'algorithme.

- Le résultat final de l'algorithme des K-moyennes est incapable de s'ajuster sur les classes gaussiennes de phonèmes parce qu'il dépend de l'initialisation de l'algorithme. La distance euclidienne carrée totale décroît de façon monotone jusqu'à un seul minium correspondant à la convergence locale. Le système d'attribution des points de données basée sur une distance minimale est équivalent à une surface discriminante linéaire.

3.5.2 Expérience sur l'Algorithme EM N° 8

Cet algorithme exige en entrée une base de données (des voyelles) ainsi que le nombre de classes d'appartenance et retourne en sortie les partitions des cinq classes et leurs centres correspondants ainsi que leurs matrices de covariances.

Nous présentons quelques figures acquises lors de l'exécution de cet algorithme

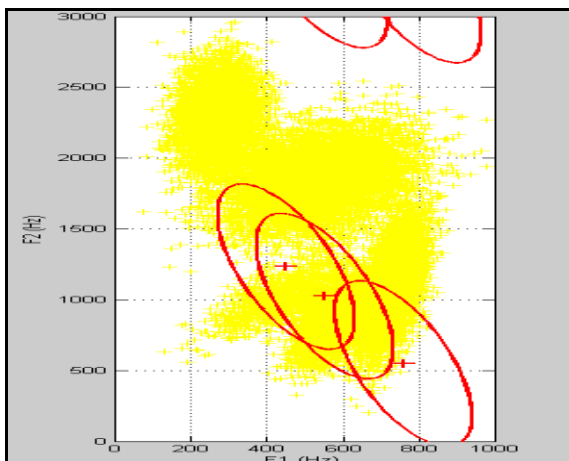


Figure (3.19) : EM à l'Initialisation.

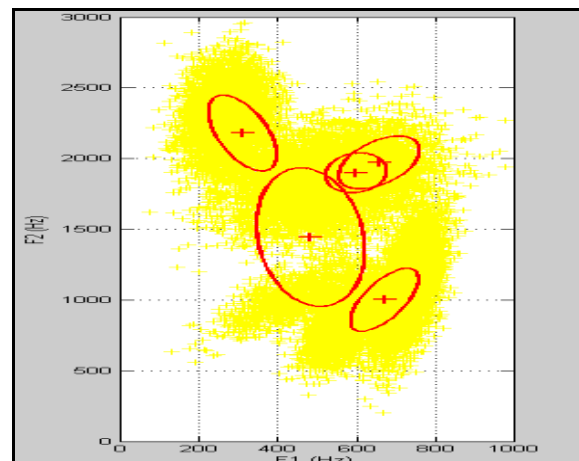


Figure (3.20) : EM après 10 Itérations.

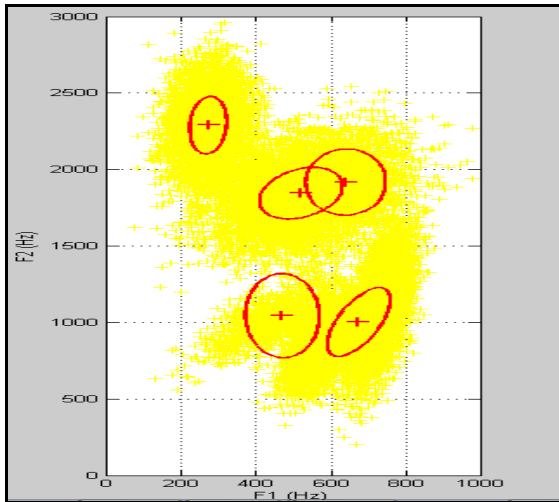


Figure (3.21) : EM après 30 Itérations.

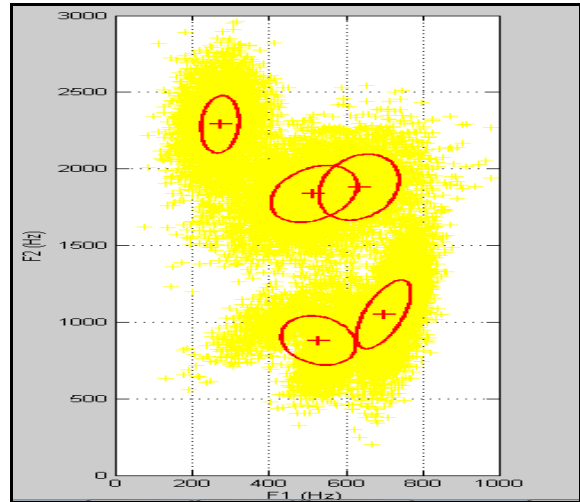


Figure (3.22) : EM après 50 Itérations.

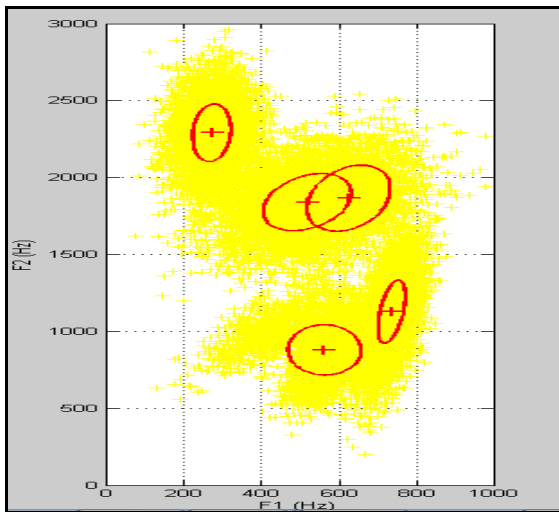


Figure (3.23) : EM après 70 Itérations.

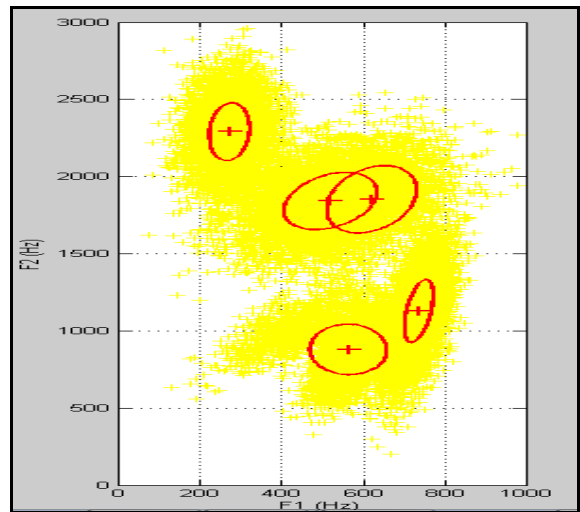


Figure (3.24) : EM après 90 Itérations.

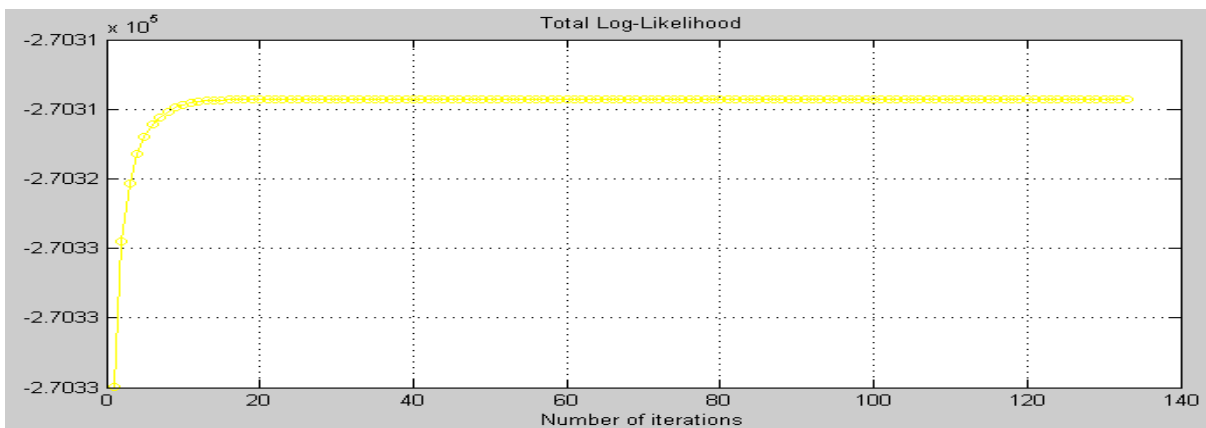


Figure (3.25) : Total de Logarithme de Vraisemblance (EM)

D'après les figures de 3.19 à 3.25 nous signalons que :

- ✓ Les vecteurs de position (moyens) s'adaptent au fur à mesure avec leurs classes, ainsi que, le logarithme de vraisemblance totale augmente de façon monotone itérativement, jusqu' à la convergence locale.
- ✓ Après chaque nouvelle exécution de l'algorithme les vecteurs de position indiqués dans les figures ne sont pas les mêmes dans toutes phase d'exécution, est compris à l'initialisation.

Interprétation :

Le résultat final dépend de l'initialisation de l'algorithme. Ce résultat correspond à un maximum local de vraisemblance totale dans le sens de la classification Bayésienne avec des modèles gaussiens.

- L'algorithme EM est capable de s'ajuster parfaitement sur les deux paramètres Gaussiens de phonèmes, mais seulement si le nombre et le placement des paramètres initiaux sont estimés correctement.

En pratique, les paramètres initiaux sont généralement initialisés par l'algorithme des K-moyennes, mais comme nous l'avons vu dans l'expérience N° 7 ceci n'est pas forcément une Estimation optimale. Le meilleur compromis est d'initialiser les paramètres dans des positions bien choisies.

3.5.3 Expérience sur l'Algorithme Viterbi_EM N° 9

Cet algorithme requête en entrée une base de données (des voyelles) ainsi que le nombre de classes d'appartenance et renvoie en sortie les divisions des cinq classes et leurs centres correspondants ainsi que leurs matrices de covariances.

Nous étalons quelques figures acquises lors de l'exécution de cet algorithme.

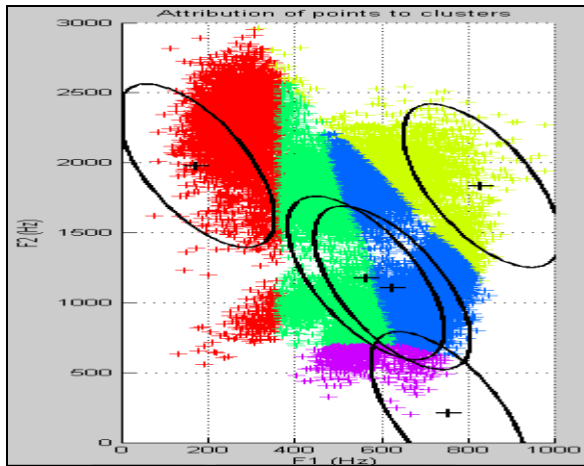


Figure (3.26) : Viterbi_EM à l'Initialisation.

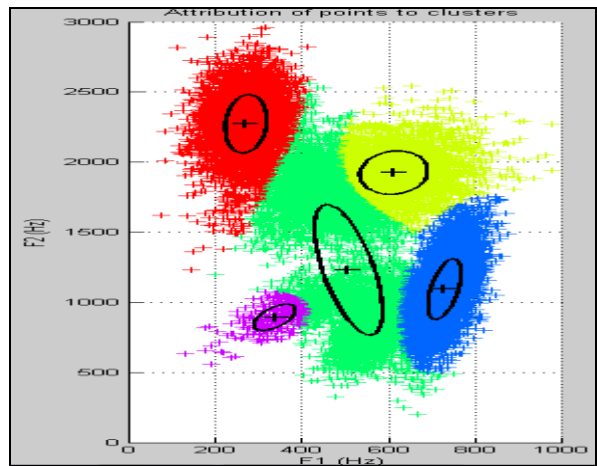


Figure (3.27) : Viterbi_EM après 10 Itérations.

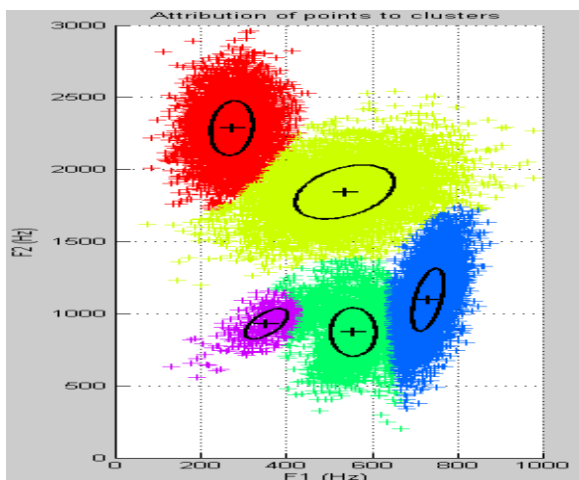
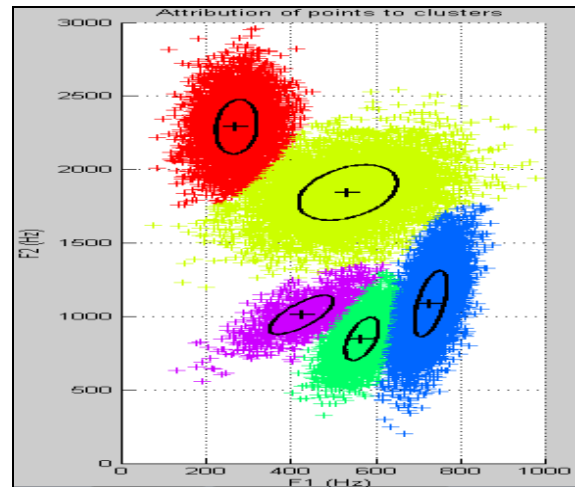


Figure (3.28) : Viterbi_EM après 20 Itérations.



Figure(3.29) : Viterbi_EM après 30 Itérations.

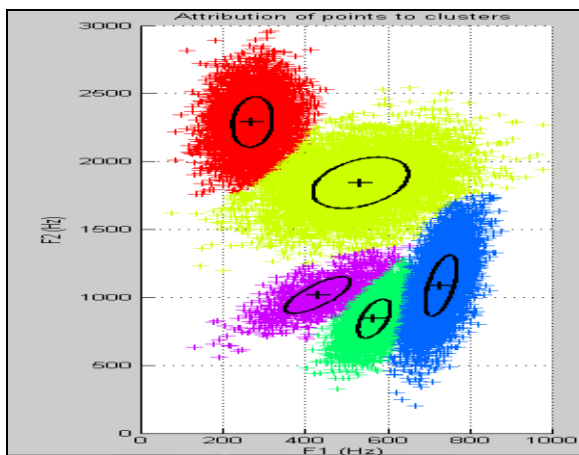


Figure (3.30) : Viterbi_EM après la Convergence.

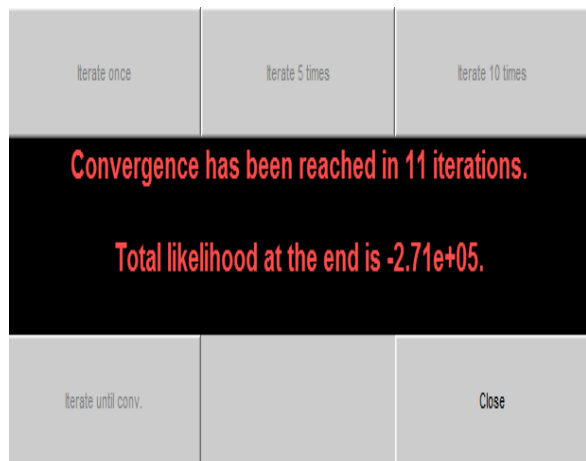


Figure (3.31) : Résultats Finals de Viterbi_EM.

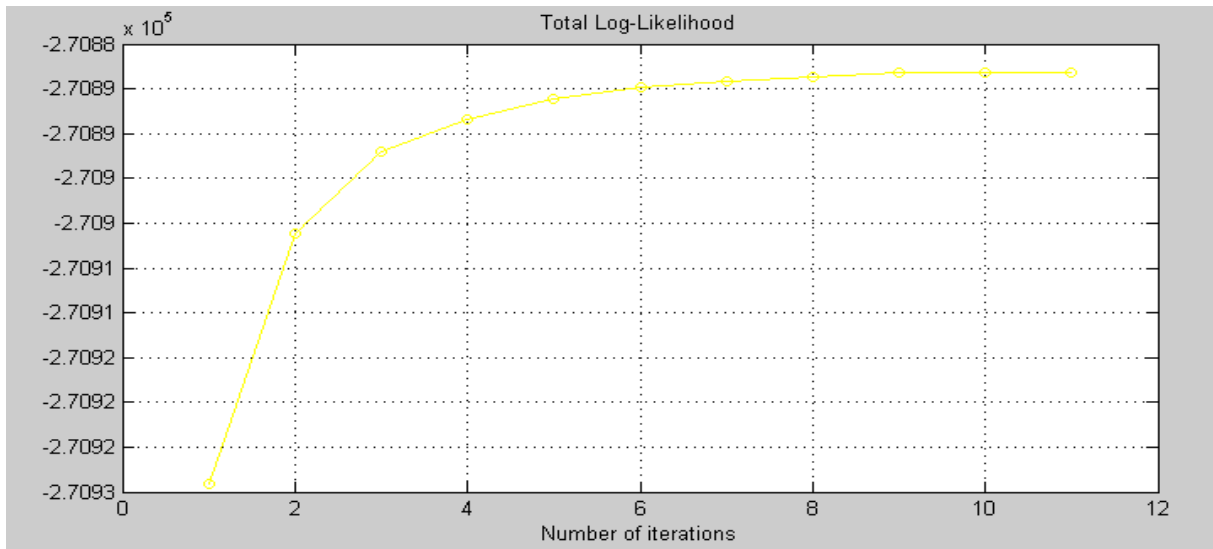


Figure (3.32) : Totale du Logarithme de Vraisemblance à la Convergence (Viterbi_EM).

Après la convergence de l'algorithme Viterbi_EM les valeurs de vecteurs moyens, des matrices de covariances et des probabilités a priori des cinq classes sont affichés dans le tableau 3.6 suivant :

Classes	Vecteur moyen	Matrice de covariance	Probabilité à priori
/a/	[0270.5 2291.2]	02571 01360 01360 34961	0.2520
/e/	[0533.1 1844.0]	13908 08089 08089 35849	0.2974
/i/	[0566.9 0884.43]	01724 03264 03264 20953	0.1508
/o/	[0729.1 1088.0]	01606 05129 05129 51919	0.2540
/y/	[0429.5 1018.0]	06559 08294 08294 18699	0.0459

Tableau (3.6) : Résultats Obtenus par Viterbi_EM.

Remarques et Interpretation :

En joignant les résultats du tableau 3.6 et celles des deux tableaux 3.3 et 3.4 nous constatons que :

Le nombre de phonèmes /a/ détectés par l'algorithme Vierbi_EM est le même nombre réel correspondant. (La probabilité à priori de la voyelle réelle (tableau 3.4) est égale a celle calculée par l'algorithme (tableau 3.6)). Les différences existant entre les vecteurs moyens et les matrices de covariance réelles et calculées prouvent que certain détectés comme une voyelle alors qu'ils ne sont pas.

Le nombre de phonèmes /e/ identifiés par cet algorithme est presque égal au nombre réel correspondant. (Les deux probabilités à priori (identifié et réelle) sont quasiment égales). Le rapprochement des valeurs des deux vecteurs moyens et des deux matrices de covariances des trois tableaux 3.3, 3.4 et 3.6 déclarent que l'algorithme à réussi à peu près avec ce phonème.

L'algorithme a mal reconnu les voyelles /i/ et /o/. Ceci est apparent dans La dissemblance des valeurs de probabilité, vecteur moyen et matrice de covariance des trois tableaux 3.3, 3.4 et 3.6.

Vu les appréciables valeurs du phonème /y/ réelles et calculées (des probabilités à priori, vecteurs moyens et matrices de covariances), nous déduisons que la majorité des voyelles nommés sont reconnues.

- Dans ce chapitre nous avons décrit toutes les expériences réalisées, ainsi que les résultats obtenus et leurs interprétations. Nous avons démontré que la convergence de chaque algorithme utilisé dans notre recherche dépend de l'initialisation des paramètres.

CONCLUSION GENERALE

Il reste incontestablement que le domaine de reconnaissance de la parole est du point de vue recherche le plus riche, Le système reconnaissance de la parole (RAP) a été développé pour être utilisé dans des application diverses, comme la réservation touristique ou dans l'hôtelière.

Il peut être utilisé pour la correction, dans une certaine mesure, les erreurs survenues lors d'une transmission à travers un canal bruité

Notre projet de fin d'étude nous a permis en premier lieu d'aborder le thème de la reconnaissance de forme, en particulier la reconnaissance de parole et plus exactement la classification des phonèmes. Notre but et aussi d'Expérimenter et d'approfondir nos connaissances théoriques sur trois algorithmes de classification (1- kmeans, 2- EM (Expectation Maximization) 3- Viterbi_Em) par un aspect expérimental sur une base de données des voyelles.

Et pour atteindre notre but nous avons opté pour l'utilisation de la classification bayésienne

Programme

```
% base de données [F1i ;F2i]
%% Expérience N°1

clc
clear
close all

N = 10000 ;
mu = [730 1090] ;
sigma_1 = [8000 0 ; 0 8000] ;
X1 = randn(N,2) * sqrtm(sigma_1) + repmat(mu,N,1) ;
gausview(X1,mu,sigma_1, 'Sample X1') ;

sigma_2 = [8000 0 ; 0 18500] ;
X2 = randn(N,2) * sqrtm(sigma_2) + repmat(mu,N,1) ;
gausview(X2,mu,sigma_2, 'Sample X2') ;

sigma_3 = [8000 8400 ; 8400 18500] ;
X3 = randn(N,2) * sqrtm(sigma_3) + repmat(mu,N,1) ;
gausview(X3,mu,sigma_3, 'Sample X3') ;

%% Expérience N°2

close all
clear all
clc
N = 6000 ;
mu=[730 1090];
sigma_3 = [8000 8400 ; 8400 18500] ;
X3 = randn(N,2) * sqrtm(sigma_3) + repmat(mu,N,1) ;
gausview(X3,mu,sigma_3, 'Sample X3') ;
%
X_1 = X3(1:6000, :) ;
N1 = size(X_1,1);
mu_1000 = sum(X_1)/N1;
sigma_1000 = cov(X_1);

% Comparaison des valeurs :
e_mu_1000 = sqrt( (mu_1000 - mu) * (mu_1000 - mu)' )
% la distance euclidienne entre mu_1000 et mu)
e_sigma_1000 = norm( sigma_1000 - sigma_3 )
% la norme matricielle de la diff_érence entre sigma_1000 sigma_3)

%% Expérience N°3

N = 10000 ;
mu = [730 1090] ; sigma = [8000 8400 ; 8400 18500] ;
```

```

X3 = randn(N,2) * sqrtm(sigma) + repmat(mu,N,1) ;
gausview(X3,mu,sigma,'Sample X3')
N = size(X3,1)
mu_1 = [270 1690] ; sigma_1 = [8000 8400 ; 8400 18500] ;
logLikel = 0 ;
for i = 1 :N ;
logLikel = logLikel + (X3(i, :) - mu_1)*inv(sigma_1)*(X3(i, :) - mu_1)' ;
end ;
logLikel = - 0.5 * ( logLikel + N*log(det(sigma_1)) + 2*N*log(2*pi) )
gausview(X3,mu_1,sigma_1,'Comparaison de X3 et N1') ;

```

%% Expérience N°4

```

% phonèmes /a/,/e/,/i/,/o/,/y/
clear all ;
load vowels.mat ;

Na = size(a,1) ;
Ne = size(e,1) ;
Ni = size(i,1) ;
No = size(o,1) ;
Ny = size(y,1) ;
N = Na + Ne + Ni + No + Ny ;
Pa = Na/N
Pi = Ni/N
Pe = Ne/N
Po = No/N
Py = Ny/N
%
mu_a = mean(a) ;
sigma_a = cov(a) ;
plotgaus(mu_a,sigma_a,[1 0 0]) ;
%
mu_e = mean(e) ;
sigma_e = cov(e) ;
plotgaus(mu_e,sigma_e,[0 1 0]) ;
%
mu_y = mean(y) ;
sigma_y = cov(y) ;
plotgaus(mu_y,sigma_y,[0 0 1]) ;
%
mu_i = mean(i) ;
sigma_i = cov(i) ;
plotgaus(mu_i,sigma_i,[1 1 0]) ;
%
mu_o = mean(o) ;
sigma_o = cov(o) ;
plotgaus(mu_o,sigma_o,[0 1 1]) ;

```

%% Expérience N°5

```
% base de données
clc
clear
close all
%phonemes /a;/e;/i;/o;/y/
load vowels.mat;
na=size(a,1);ne=size(e,1);ni=size(i,1);
no=size(o,1);ny=size(y,1);
n=na+ne+ni+no+ny;
%% probabilites a priori des classes %%
pa=na/n;pe=ne/n;pi=ni/n;
po=no/n;py=ny/n;
mua=mean(a);sigmaa=cov(a);plotgaus(mua,sigmaa,[0 1 1]);
mue=mean(e);sigmae=cov(e);plotgaus(mue,sigmae,[1 1 1]);
mui=mean(i);sigmai=cov(i);plotgaus(mui,sigmai,[0 1 0]);
muo=mean(o);sigmao=cov(o);plotgaus(muo,sigmao,[1 0 0]);
muy=mean(y);sigmay=cov(y);plotgaus(muy,sigmay,[0 0 1]);
```

%% Expérience N°6

```
% base de données
clc
clear
close all
%phonemes /a;/e;/i;/o;/y/
load vowels.mat;
na=size(a,1);ne=size(e,1);ni=size(i,1);
no=size(o,1);ny=size(y,1);
n=na+ne+ni+no+ny;
%% probabilites a priori des classes %%
pa=na/n;pe=ne/n;pi=ni/n;
po=no/n;py=ny/n;
mua=mean(a);sigmaa=cov(a);plotgaus(mua,sigmaa,[0 1 1]);
mue=mean(e);sigmae=cov(e);plotgaus(mue,sigmae,[1 1 1]);
mui=mean(i);sigmai=cov(i);plotgaus(mui,sigmai,[0 1 0]);
muo=mean(o);sigmao=cov(o);plotgaus(muo,sigmao,[1 0 0]);
muy=mean(y);sigmay=cov(y);plotgaus(muy,sigmay,[0 0 1]);
gausview(a,mua,sigmaa,'sample x1');
gausview(a,mue,sigmae,'sample e');
xx{1}=[400,1800];xx{2}=[400,1000];xx{3}=[530,1000];
xx{4}=[600,1300];xx{5}=[670,1300];xx{6}=[420,2500];
for i=1:6

yy{i}=[gloglike(xx{i},mua,sigmaa)+log(pe);gloglike(xx{i},mue,sigmae)+log(pe)
];

gloglike(xx{i},mui,sigmai)+log(pi);gloglike(xx{i},muo,sigmao)+log(po);
gloglike(xx{i},muy,sigmay)+log(py)];
xx{i}=max(yy{i});
end
```

%% Expérience N°7

```
clc
clear
load vowels.mat;
%kmeans(allvow,5);
mu_a=mean(a);mu_e=mean(e);mu_i=mean(i);
mu_o=mean(o);mu_y=mean(y);
means={mu_a,mu_e,mu_i,mu_o,mu_y};
kmeans(allvow,means);
%agrandit la fenêtre, puis pousse les boutons, zoom ect. après la
convergence, utilisez:
for k=1:5;
    disp(kmeans_result_means{k});
end
```

%% Expérience N°8

```
clear all ;
load vowels.mat ;
Na = size(a,1) ;
Ne = size(e,1) ;
Ni = size(i,1) ;
No = size(o,1) ;
Ny = size(y,1) ;
N = Na + Ne + Ni + No + Ny ;
Pa = Na/N
Pi = Ni/N
%
mu_a = mean(a) ;
sigma_a = cov(a) ;
plotgaus(mu_a,sigma_a,[1 0 0]) ;
%
mu_e = mean(e) ;
sigma_e = cov(e) ;
plotgaus(mu_e,sigma_e,[0 1 0]) ;
%
mu_y = mean(y) ;
sigma_y = cov(y) ;
plotgaus(mu_y,sigma_y,[0 0 1]) ;
%
mu_i = mean(i) ;
sigma_i = cov(i) ;
plotgaus(mu_i,sigma_i,[1 0 0]) ;
%
mu_o = mean(o) ;
sigma_o = cov(o) ;
plotgaus(mu_o,sigma_o,[1 0 0]) ;
means = { mu_a, mu_e, mu_i, mu_o, mu_y } ;
vars = { sigma_a, sigma_e, sigma_i, sigma_o, sigma_y } ;
emalgo(allvow,means,vars) ;
```



```

%Aggrandissez la fen^etre, puis essayez les boutons, agrandissez, etc.
Apr_es la convergence, utilisez :
for k=1 :5,disp(emalgo_result_means{k}) ; end
for k=1 :5,disp(emalgo_result_vars{k}) ; end
for k=1 :5,disp(emalgo_result_priors(k)) ; end

```

%% Expérience N°9

```

clear all ;
load vowels.mat ;

mu_a = mean(a) ;
sigma_a = cov(a) ;
plotgaus(mu_a,sigma_a,[0 0 1]) ;
%
mu_e = mean(e) ;
sigma_e = cov(e) ;
plotgaus(mu_e,sigma_e,[1 0 0]) ;
%
mu_y = mean(y) ;
sigma_y = cov(y) ;
plotgaus(mu_y,sigma_y,[1 0 0]) ;
%
mu_i = mean(i) ;
sigma_i = cov(i) ;
plotgaus(mu_i,sigma_i,[1 0 0]) ;
%
mu_o = mean(o) ;
sigma_o = cov(o) ;
plotgaus(mu_o,sigma_o,[1 0 0]) ;
means = { mu_a, mu_e, mu_i, mu_o, mu_y } ;
vars = { sigma_a, sigma_e, sigma_i, sigma_o, sigma_y } ;
viterb(allvow,means,vars) ;

%Aggrandissez la fen^etre, puis essayez les boutons, agrandissez, etc.
Apr_es la convergence, utilisez :

for k=1 :5, disp(viterb_result_means{k}) ; end
for k=1 :5, disp(viterb_result_vars{k}) ; end
for k=1 :5, disp(viterb_result_priors(k)) ; end

```