

CHAPITRE 4

∞ METHODES PROPOSEES ET RESULTATS OBTENUES ∞

Introduction

Dans cette partie nous exposerons la méthode utilisée afin de réaliser la reconnaissance du locuteur. L'objectif de cette étude (connexionniste) est la réalisation d'un système de vérification automatique du locuteur (VAL) en utilisant des réseaux machine à vecteur de support (SVM). Nous allons présenter, dans un premier temps, la technique d'apprentissage et de test, puis dans un second temps nous parlerons de la phase de prétraitements effectuée sur les fichiers de parole. Enfin en donne La mise en œuvre d'algorithme SVM

1. Réseau SVM utilisé

Les SVM sont des classifieurs qui reposent sur deux idées clés, permettant de traiter des problèmes de discrimination non linéaire, et de reformuler le problème de classification comme un problème d'optimisation quadratique.

La première idée est la notion de marge maximale. La marge est la distance entre la frontière de séparation et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. Dans les SVM, la frontière de séparation est choisie comme celle qui maximise la marge. Ce choix est justifié par la théorie de Vapnik Chervonenkis (ou théorie statistique de l'apprentissage), qui montre que la frontière de séparation de marge maximale possède la plus petite capacité. Le problème est de trouver cette frontière séparatrice optimale, à partir d'un ensemble d'apprentissage. Ceci est fait en formulant le problème comme un problème d'optimisation quadratique, pour lequel il existe des algorithmes connus.

Afin de pouvoir traiter des cas où les données ne sont pas linéairement séparables, la deuxième idée clé des SVM est de transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, dans lequel il est probable qu'il existe une séparatrice linéaire. Ceci est réalisé grâce à une fonction noyau, qui doit respecter certaines conditions, et qui a l'avantage de ne pas nécessiter la connaissance explicite de la transformation à appliquer pour le changement d'espace. Les fonctions noyau permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux, en une simple évaluation ponctuelle d'une fonction. Cette technique est connue sous le nom de "kernel trick".

2. DESCRIPTION DE LA BASE DE DONNEES UTILISEE

Nous avons utilisé, pour l'apprentissage du réseau de neurones et les tests de notre système, un sous-ensemble de la base de données universelle (TIMIT) appelé (DR1). Cet sous-ensemble est composé de 37 locuteurs, 15 femmes et de 22 hommes. Chaque locuteur prononce 10 phrases dans un dialecte appelé "*New England* " ; huit phrases complètement différentes d'un locuteur à un autre et deux phrases communes entre tous les locuteurs. Les fichiers de parole sont enregistrés à travers un microphone de très haute sensibilité et sont échantillonnés à 16 kHz et codés sur 16 bits.

3. METHODE SPECTRALE

3.1 Analyse acoustique et paramétrisation du signal vocal

L'analyse de la parole consiste à extraire l'information pertinente et à réduire au maximum la redondance.

Ceci est réalisé par les étapes suivantes :

- Sélection d'une fenêtre de 3s du signal de parole obtenu après mixage. Cette fenêtre se décale avec un pas de 0.02 s.

- La fenêtre de 3 s est à son tour décomposée en fenêtres d'analyse de 32 ms, avec un recouvrement de 50%
- Extraction des caractéristiques pour chaque fenêtre d'analyse (paramétrisation). Dans ce travail nous avons choisi de représenter les signaux de parole par les coefficients MFSC (Mel Frequency Spectral Coefficients), qui sont fréquemment utilisés dans les applications de reconnaissance du locuteur. Ceux-ci sont obtenus après le calcul du spectrogramme sur une partie stationnaire du signal de parole. On multiplie alors le spectrogramme obtenu par un banc de filtres choisis de manière à imiter au mieux la perception auditive de l'être humain. Un tel banc de filtres est représenté dans la figure ci-dessous.

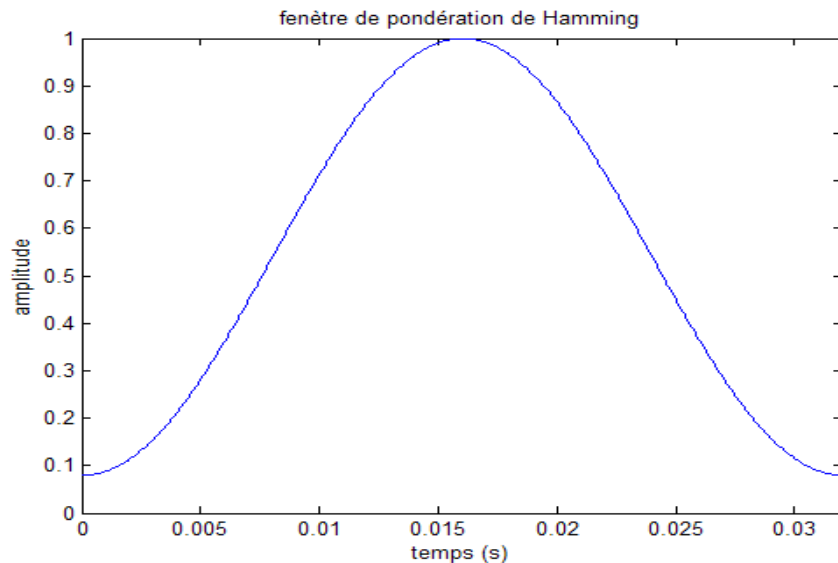


Figure 4.1 : Fenêtre de pondération de Hamming.

On constate que les fenêtres sont très serrées dans les basses fréquences et s'élargissent pour les hautes fréquences. Ceci s'accorde avec le modèle de perception humaine. En effet l'oreille humaine distingue mieux les basses fréquences que les hautes fréquences.

- La sortie de chaque filtre est calculée en sommant toutes les composantes fréquentielles à l'intérieur de la bande passante du filtre, pondérées par les coefficients du filtre, comme ce qui suit :

$$Y_k(m) = \sum_i c_{ik} Y^i(m) \quad (4.1)$$

- où $Y^k(m)$ est la sortie du k^e canal du banc de filtres pour la m^e fenêtre. $Y^i(m)$ est l'énergie du signal (carré du module du spectre) dans le i^e coefficient de Fourier pour la m^e fenêtre. c_{ik} est le gain du k^e filtre dans le i^e échantillon fréquentiel et i est sommé sur la bande passante du filtre. Un écrêtage à seuil d'énergie est utilisé pour limiter le rang dynamique du signal de parole. A la sortie de l'écrêtage on a :

$$\begin{aligned} \tilde{s}_k &= \max [s_k, T_K^\circ] \\ \text{où } T_K^\circ &= \max_m [s_{k(m)}, T_K^\circ] - 30dB \end{aligned} \quad (4.2)$$

et où S_k représente la sortie Y_k dans l'échelle logarithmique (en dB).

Les canaux à faible énergie en sortie sont ainsi éliminés car ils correspondent souvent à l'effet du bruit. Une normalisation de ces coefficients par leur énergie moyenne est effectuée, parce que le niveau d'énergie de parole varie d'une prononciation à l'autre et ceci est indésirable dans le cas de la reconnaissance. Ceci se fait, d'abord, en calculant l'énergie moyenne $\bar{s}(m)$ de chaque trame sur les K canaux par :

$$\bar{s}(m) = \frac{1}{K} \sum_{k=1}^K \tilde{s}_k(m) \quad (4.3)$$

La normalisation d'énergie est donnée, alors, par

$$s_k^n(m) = \tilde{s}_k(m) - \bar{s}(m) \quad (4.4)$$

Finalement les coefficients $S^n(m)$ pour chaque trame " m " sont stockés dans des vecteurs de dimension K , appelés les MFSC ou *Mel Frequency Spectral Coefficients*. Ce sont les caractéristiques du locuteur utilisées dans cette étude.

Nous faisons remarquer que le signal de parole ne subit pas de préaccentuation.

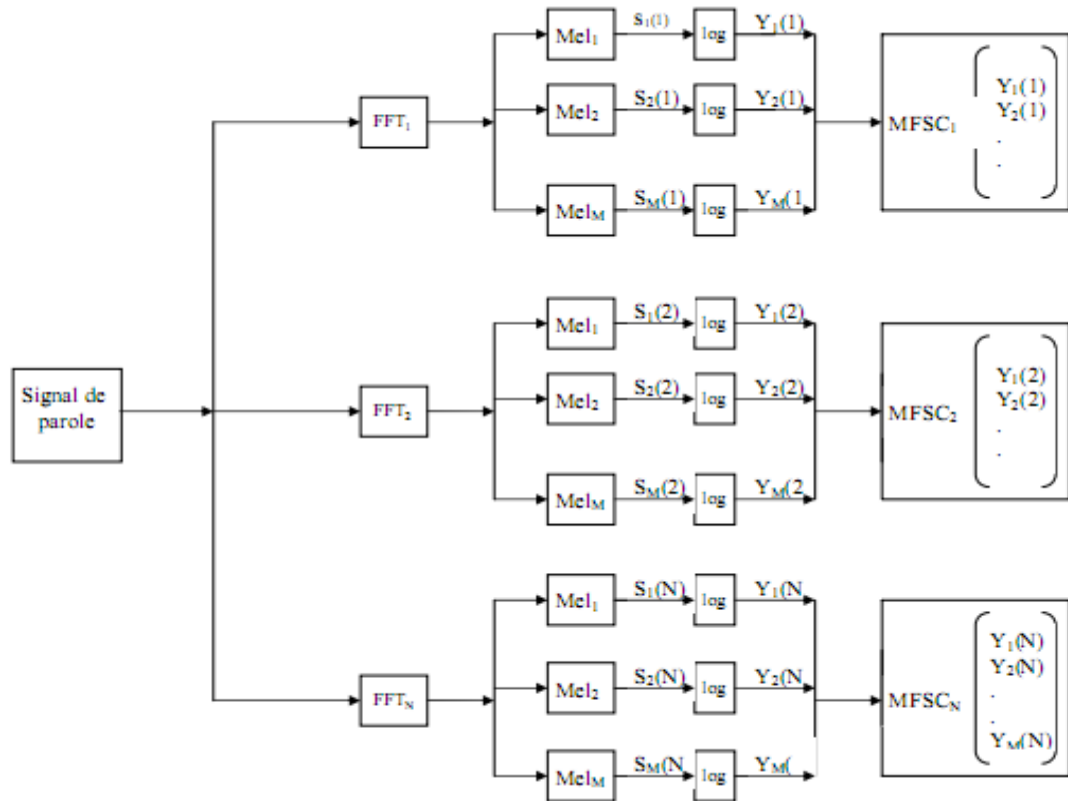


Figure 4.2 : Principe d'extraction des MFSC.

- Réduction de l'information

Dans cette méthode, nous utilisons les caractéristiques spectrales du signal de parole issu des locuteurs (24 caractéristiques MFSC sont utilisées). A partir de ces caractéristiques MFSC, on extrait le vecteur moyen MFSCmoy qui sera le seul élément représentatif du locuteur dans ce cas : dans cette perspective nous supposons que la variabilité de la matrice de covariance est négligeable. Nous préservons alors seulement le vecteur moyenne : MFSCmoy (de dimension 24x1) et chaque locuteur sera représenté par ce vecteur.

3.2. Caractéristique Relative du Locuteur (RSC)

Dans le but d'optimiser le nombre d'entrées et la convergence des SVM, nous avons développé une nouvelle Caractéristique Relative du Locuteur, que nous avons appelée RSC (Relative Speaker Characteristic).

3.3 Notion de DRSC (Diagonal of RSC)

Pour modéliser le locuteur, la plupart des systèmes de reconnaissance du locuteur existants utilisent les composantes statistiques comme le vecteur moyenne ou la matrice de covariance, qui sont extraits à partir des caractéristiques acoustiques comme : les coefficients MFSC, les LFCC (Linear Frequency Cepstral Coefficients), les MFCC (Mel Frequency Cepstral Coefficients), Cette caractéristique est appelée : Caractéristique Relative du Locuteur (RSC). De plus, en utilisant cette dernière, nous arrivons à réduire la taille du vecteur d'entrée des deux classifieurs et accélérer la phase d'apprentissage.

Soient à comparer deux segments de parole, appartenant aux locuteurs x et y . Leurs matrices de covariance sont respectivement X et Y et leurs vecteurs moyenne sont respectivement x_m et y_m .

$$\varphi(X, Y) = \frac{1}{p} \left[-\log \left(\frac{\det(Y)}{\det(X)} \right) + \text{tr}(YX^{-1}) \right] - 1 \quad (4.5)$$

Mais, du fait que

$$\frac{\det(Y)}{\det(X)} = \det(Y/X) \quad (4.6)$$

Alors

$$\varphi(X, Y) = \frac{1}{p} \left[-\log(\det(Y/X) + \text{tr}(Y/X)) \right] - 1 \quad (4.7)$$

Si nous introduisons le rapport de relativité \Re par :

$$\Re(X, Y) = \frac{X}{Y} = X Y^{-1} \quad (4.8)$$

$$\text{Alors} \quad \varphi(x, y) = \frac{1}{p} \left[-\log(\det(\Re(y, x)) + \text{tr}(\Re(y, x))) \right] - 1 \quad (4.9)$$

Donc $\varphi(x, y)$ est une fonction du rapport de relativité $\Re(y, x)$

Nous avons appelé le rapport \Re : caractéristique relative du locuteur (Relative Speaker Characteristic RSC). L'acronyme DRSC représente la diagonale de la matrice RSC.

$$DRSC(x, y) = \text{diag}(\Re(y, x)) \quad (4.10)$$

4 TECHNIQUE D'APPRENTISSAGE ET DE TEST

Les techniques d'apprentissage et de test, utilisées dans nos expériences, sont résumées sur le synoptique de la figure 4.3 suivante :

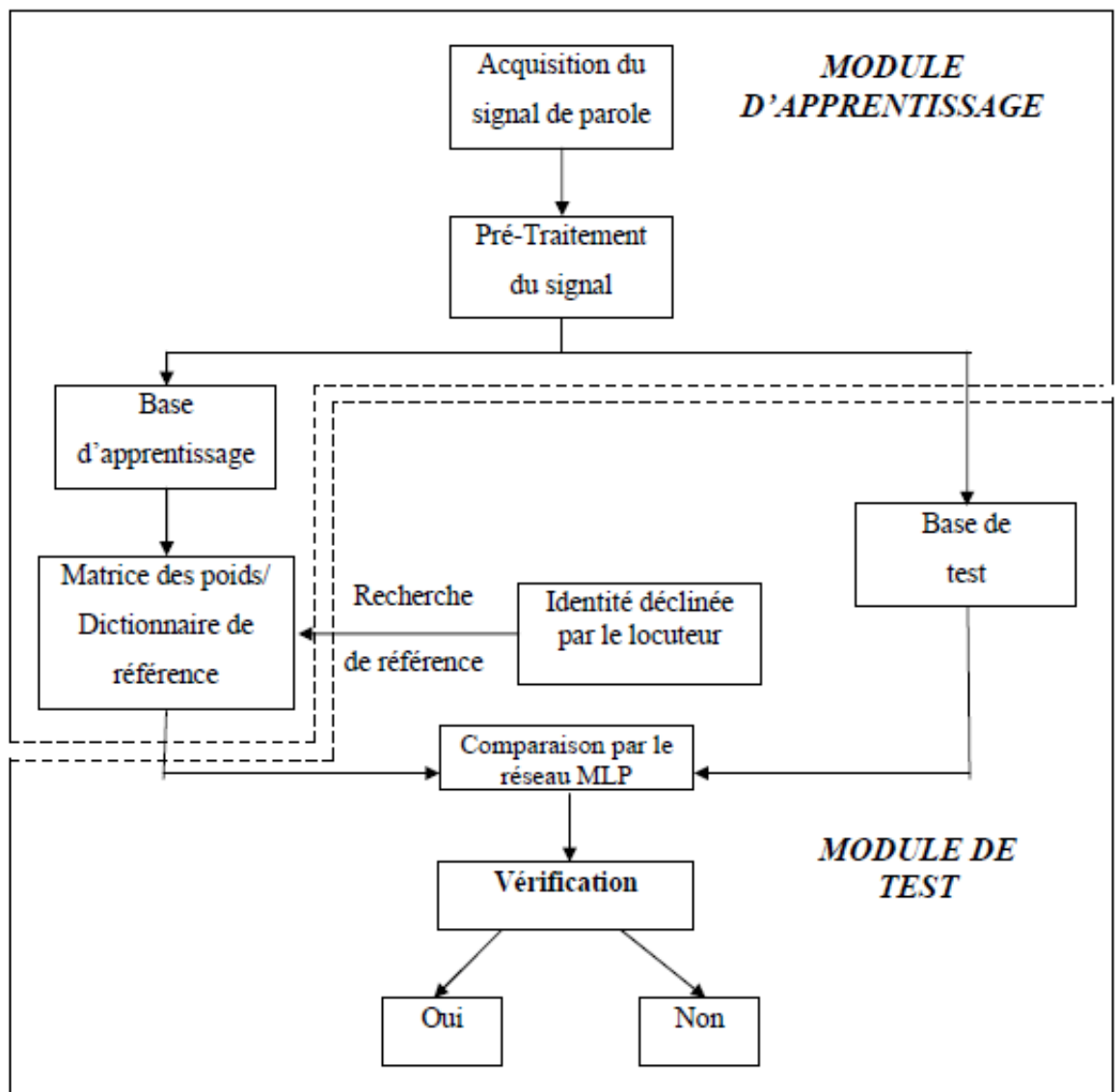


Figure 4.3: Synoptique général de la technique d'apprentissage et de test.

Le module d'apprentissage (pour un locuteur donné) fait présenter au SVM une série d'exemples différents appartenant au locuteur désiré (que l'on veut référencer) et à d'autres locuteurs différents. A chaque fois, on vérifie la sortie du SVM. A la fin de l'apprentissage on sauvegarde les poids obtenus : ceux-ci représentent, alors, la référence SVM du locuteur désiré.

Le module de test, par contre, fait affecter au vecteur support les poids (déjà sauvegardés) correspondant au locuteur proclamé. Il calcule ensuite la sortie du réseau qui décidera de l'acceptation ou du refus de l'identité proclamée.

5. MISE EN ŒUVRE D'ALGORITHME SVM

La mise en œuvre d'algorithme SVM requiert la programmation d'outils spécifiques. Nous allons présenter dans cette section, les différents aspects de la mise en œuvre software de l'algorithme d'apprentissage et de généralisation de la méthode SVM. Nous allons reprendre une partie de la théorie présentée dans ce chapitre concernant cette méthode pour expliquer les spécifications des outils d'apprentissage et de généralisation.

5.1 Dimension des Entrées de Classifieur SVM

Le nombre de machine à vecteur de support d'entrée est un vecteur de taille fixe qui devrait représenter la taille des coefficients MFSC. Or pour chaque fenêtre du signal, un vecteur de taille M ($M=12$) est produit. Ainsi, si nous avons 1000 fenêtres. De ce fait, si nous avons 37 caractéristiques acoustiques (dans notre cas: 37 MFSC (Mel Frequency Spectral Coefficients)) pour chaque locuteur.

Pour cette raison on a procédé à une normalisation (en éliminant le facteur temps) des entrées en utilisant :

- Le vecteur moyen.
- la diagonale de la covariance extraite des MFSC,
- les deux premiers vecteurs propres de cette covariance.

Ainsi, la dimension de la couche d'entrée ne dépendra que des caractéristiques choisies à l'entrée du réseau : par exemple, si l'on choisit les 2 premiers vecteurs propres comme entrée du réseau alors le nombre de neurones d'entrée sera 2×12 soit 24 neurones. Rappelons que ces trois paramètres sont calculés statistiquement de la manière suivante : Soit $x : \{ x_j \}$ une séquence de N vecteurs de dimension M résultant de l'analyse acoustique d'un signal de parole prononcé par le locuteur x (coefficients MFSC, par exemple). La moyenne \bar{x} et la matrice de covariance Cov sont alors définies par:

$$\bar{x} = \frac{1}{N} \sum_{j=1}^N x_j \quad , \quad cov_x = \frac{1}{N} \sum_{j=1}^N (x_j - \bar{x})(x_j - \bar{x})^T \quad (4.11)$$

Le vecteur diagonal est directement déduit de la matrice de covariance.

Pour le calcul des deux premiers vecteurs propres, on calcule tout d'abord les valeurs propres λ_i de la covariance, puis on calcule leurs vecteurs propres correspondants V_i à partir de l'équation suivante :

$$(\lambda_i I - cov)V_i = 0 \quad (4.12)$$

5.2 Apprentissage

Nous avons posé précédemment le problème SVM comme un problème de minimisation quadratique. Il est valable pour le cas d'un problème séparable. Dans le cas d'un problème de classification non séparable, la machine attribuera une sortie fautive à un vecteur x_i si le ξ_i correspondant est supérieur à 1. La somme de tous les ξ_i représente donc une borne du nombre d'erreurs. Dans ce cas, au lieu de chercher le vecteur de poids w_0 qui minimise la norme (w, w) , on cherche maintenant à minimiser, sous les contraintes exprimées ci-dessus :

$$Q(w, \xi) = \frac{1}{2} (w, w) + C \sum_{i=1}^m \xi_i \quad (4.13)$$

La solution a_i s'obtient une fois encore en maximisant le Lagrangien dual admettant la même expression que dans l'équation, mais sous une contrainte un peu différente, $0 \leq a_i \leq C$, pour $i = \{1 \dots m\}$, où C est un paramètre qui peut être choisi par l'utilisateur (plus ce paramètre est grand, plus cela revient à attribuer une forte pénalité aux erreurs). On obtient l'équation d'un hyperplan qui est optimal. Les vecteurs de support sont toujours les vecteurs-exemples les plus proches de l'hyperplan, mais, cette fois-ci, il existe des vecteurs exemple qui sont situés dans le mauvais demi-espace et ils ne seront donc pas considérés pour construire l'hyperplan.

La réalisation d'un programme d'apprentissage par SVM se ramène essentiellement à résoudre un problème d'optimisation impliquant un système de résolution de programmation quadratique dans un espace de dimension conséquente avec les contraintes. On peut exprimer ce problème d'optimisation dans sa formulation duale sous forme matricielle, on obtient :

$$\max_{\alpha_i} \quad \frac{1}{2} \alpha^t H \alpha + c^t \alpha \quad (4.14)$$

$$\text{telque:} \quad H = ZZ^t, \quad c^t = (-1, \dots, -1) \quad (4.15)$$

$$\text{sous contraintes :} \quad \alpha^t Y = 0, \quad \alpha_i \geq 0, \quad i = 1, \dots, l \quad (4.16)$$

$$\text{telque :} \quad Z = \begin{pmatrix} y_1 x_1 \\ \vdots \\ y_l x_l \end{pmatrix} \quad Y = \begin{pmatrix} y_1 \\ \vdots \\ y_l \end{pmatrix} \quad (4.17)$$

a : sont les multiplicateurs de Lagrange

$$H : \text{la matrice Hessienne, telque} \quad H = y_i y_j K(x_i, x_j) \quad (4.18)$$

Ce problème d'optimisation des SVM peut être résolu de manière analytique uniquement quand le nombre d'exemples est réduit ou quand, dans le cas séparable, l'on sait quels exemples sont les vecteurs de support. Pour des problèmes réduits, on peut utiliser les outils d'optimisation qui résolvent des programmes quadratiques convexes restreints.

Pour des problèmes plus importants, il existe des techniques pour pouvoir calculer une solution. Le mécanisme de base est le suivant :

- Noter les conditions d'optimalité (KKT) que la solution doit satisfaire ;
- Définir une stratégie pour arriver à l'optimalité en augmentant uniformément la fonction objective duale sous contrainte (méthode d'optimisation).

Les paramètres d'entrées du programme d'apprentissage sont les suivantes :

- Base de données, vecteurs de données et la classe correspondante ;
- La taille des vecteurs d'entrées, noter n ;
- Le nombre d'exemples de la base de données, noter m ;
- Kernel ou fonction noyau : linéaire, polynomial, RBF,... et leurs paramètres ;
- L'algorithme d'optimisation quadratique à utiliser ;

5.3 Noyaux et constante C

▪ **Noyaux** : sont des fonctions mathématiques réalisent un produit interne entre les vecteurs d'entrée. Satisfaisant les conditions de Mercer, nous avons utilisé les trois noyaux les plus couramment utilisés suivants :

- Fonction linéaire

$$K(x_i, x_j) = x_i \cdot x_j \quad (4.19)$$

- Fonction polynomiale de degré 2

$$K(x_i, x_j) = (x_i \cdot x_j)^2 \quad (4.20)$$

- RBF (Radial Basis Function)

$$K(x_i, x_j) = \exp \left\{ -\frac{|x_i - x_j|^2}{2\sigma^2} \right\} \quad (4.21)$$

▪ **Constante C** (facteur de pénalisation d'erreurs dans la classification) : le réglage de la constante C relève du problème de la sélection de modèle de classification, il a eu une relation directe avec les performances de l'algorithme d'apprentissage.

La structure générale du programme d'apprentissage de l'algorithme SVM suit l'organigramme suivant :

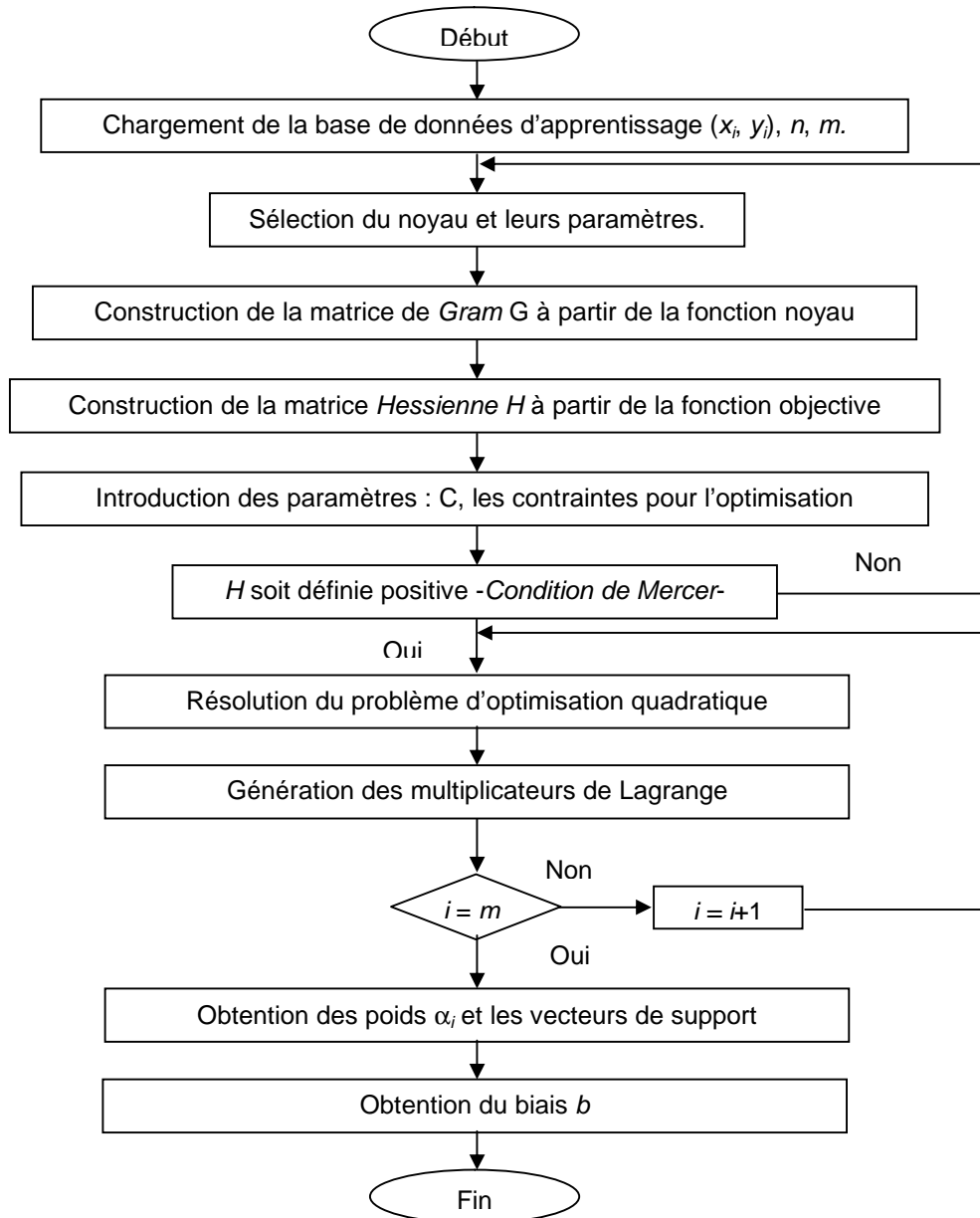


Figure 4.4 : Structure générale du programme d'apprentissage SVM.

Les sorties du programme sont :

- Les vecteurs de support (VS) ;
- Les poids du réseau ou valeurs α_i ;
- Valeur de seuil (le biais) du réseau ou valeur b ;
- Les vecteurs d'entrées retenus comme vecteurs de support ;

Le temps de calcul, ainsi que l'information générale comme par exemple le nombre de vecteurs de support.

5.4 Généralisation

L'implémentation de l'algorithme de généralisation s'appuie sur la programmation des différentes fonctions de décision du réseau choisi à partir de la phase d'apprentissage. Nous reprenons la fonction de décision :

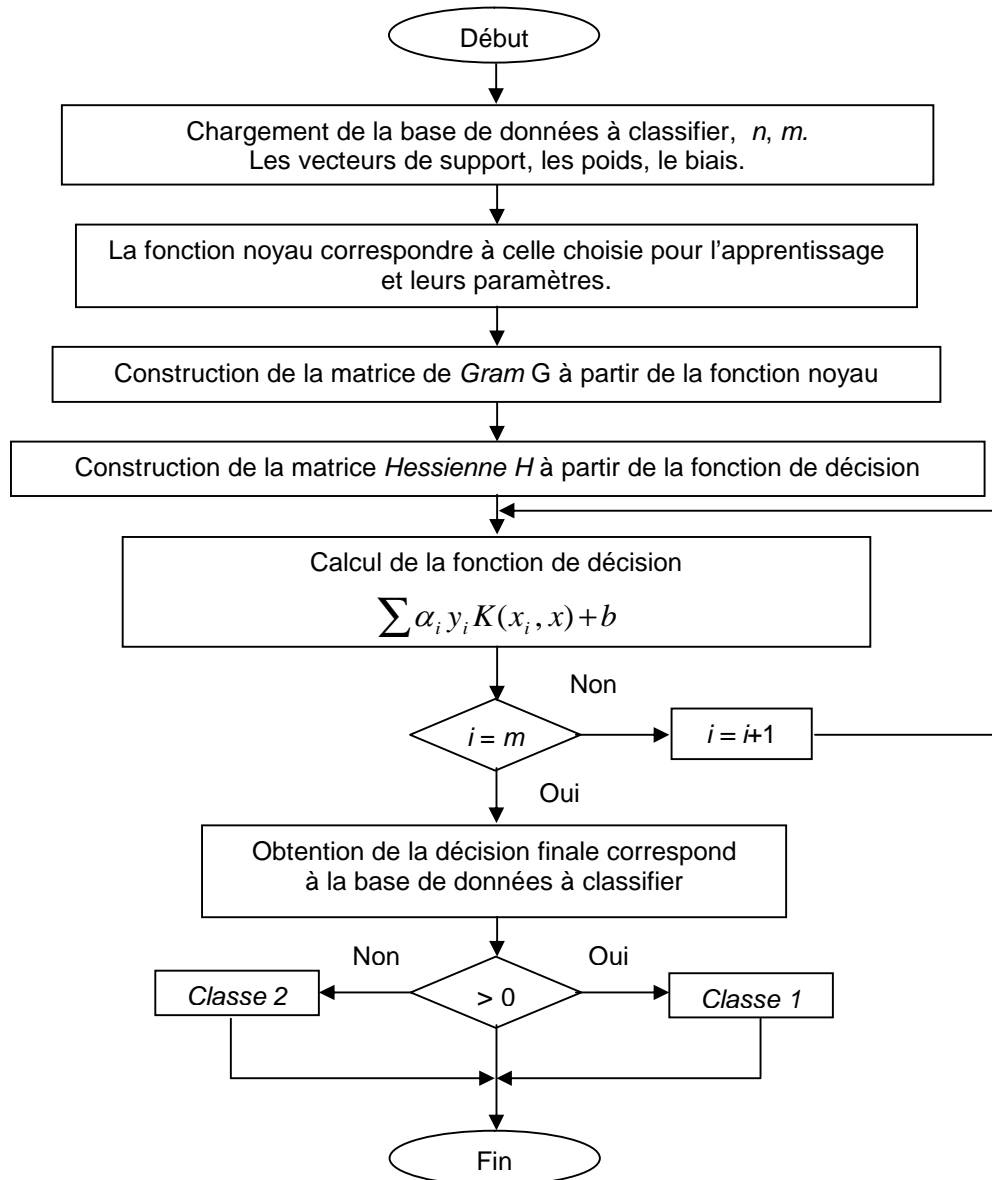
$$f(x, \alpha) = \text{sign} \left[\sum_{\text{Vecteurs de Support}} y_i \alpha_i K(x_i, x) + b \right] \quad (4.22)$$

En remplaçant $K(x_i, x)$ par les fonction noyau définies dans le réseau choisi : linéaire, polynomial, RBF,.....

Nous avons donc, pour le programme de généralisation, les paramètres d'entrées sont les suivants :

- La base d'exemples à classifier ;
- La taille de vecteurs d'entrée n ;
- Le nombre d'exemples de la base de données à classifier m ;
- Fonction noyau, qui doit correspondre à celle choisie pour l'apprentissage et leurs paramètres ;
- Les trois fichiers créent lors de l'apprentissage : vecteurs de support, poids synaptiques et le biais.

La structure générale du programme de généralisation de l'algorithme SVM suit l'organigramme suivant :



Les sorties de la généralisation représentant les classes des exemples évalués à partir de la fonction de décision. Comme dans le cas de l'apprentissage, le temps de calcul approximatif est obtenu lors de l'exécution de programme.

L'utilisation de ces programmes revient surtout de résoudre un problème d'optimisation lors de la phase d'apprentissage.

6 EXPERIENCES ET RESULTATS OBTENUS

Dans ce qui suit, nous procédons à des expériences de VAL (vérification automatique du locuteur) par le biais du SVM. Mais, tout d'abord, nous devons définir taux d'erreur souvent employés en VAL.

6.1 Résultats obtenus pour même locuteur

Tableau.4.1 : Les résultats obtenus pour même locuteur.

Fichier audio	Durée	Décision
L1.1 L1.2	4.584	0
L2.1 L2.2	4.512	0
L3.1 L3.2	4.5625	0
L4.1 L4.2	4.569	0
L5.1 L5.2	4.5754	0
L6.1 L6.2	4.5258	0
L7.1 L7.2	4.582	0
L8.1 L8.2	4.594	0
L9.1 L9.2	4.606	0
L10.1 L10.2	4.5518	0
L11.1 L11.2	4.5755	0
L12.1 L12.2	4.5645	0

Fichier audio	Durée	Décision
L13.1 L13.2	4,57	0
L14.1 L14.2	4.556	0
L15.1 L15.2	4.587	0
L16.1 L16.2	4.5744	0
L17.1 L17.2	4.3897	0
L18.1 L18.2	4.5821	0
L19.1 L19.2	4.5721	0
L20.1 L20.2	4.608	0
L21.1 L21.2	4.5602	0
L22.1 L22.2	4.5505	0
L23.1 L23.2	4.5699	0
L24.1 L24.2	4.5580	0
L25.1 L25.2	4.561	0
L26.1 L26.2	4.5712	0
L27.1 L27.2	4.5812	0
L28.1 L28.2	4.5647	0
L29.1 L29.2	4.5111	0
L30.1 L30.2	4.5704	0

On remarque que le nombre qui vérifié de locuteurs différent reconnais est 0/30 locuteurs donc le taux total de vérification est 0 %

6.2 Résultats obtenus pour locuteurs différents

Tableau. 4.2 Les résultats obtenus pour locuteurs différents.

Fichier audio	Durée	Décision
L1.1 L2.2	4.5548	0
L2.1 L3.2	4.5472	0
L3.1 L4.2	4.5559	0
L4.1 L5.2	4.549	0
L5.1 L6.2	4.5435	0
L6.1 L7.2	4.5624	0
L7.1 L8.2	4.5723	0
L8.1 L9.2	4.5989	0
L9.1 L10.2	4.5891	0
L10.1 L11.2	4.6018	0
L11.1 L12.2	4.532	0
L12.1 L13.2	4.5724	0
L13.1 L14.2	4.5506	0
L14.1 L15.2	4.5611	0
L15.1 L16.2	4.562	0
L16.1 L17.2	4.5883	0
L17.1 L18.2	4.5577	0

Fichier audio	Durée	Décision
L18.1 L19.2	4.6047	0
L19.1 L20.2	4.5565	0
L20.1 L21.2	4.523	0
L21.1 L22.2	4.576	0
L22.1 L23.2	4.5547	0
L23.1 L24.2	4.5086	0
L24.1 L25.2	4.5673	0
L25.1 L26.2	4.5642	0
L26.1 L27.2	4.5862	0
L27.1 L28.2	4.5589	0
L28.1 L29.2	4.5689	0
L29.1 L30.2	4,5981	0
L30.1 L31.2	4,5623	0
L31.1 L32.2	4.5473	0
L32.1 L33.2	4.5547	0
L33.1 L34.2	4.5577	0
L34.1 L35.2	4.5882	0
L35.1 L36.2	4.5612	0
L36.1 L37.2	4.5506	0

Fichier audio	Durée	Décision
L37.1 L37.2	4.5724	0
L1.1 L3.2	4.5518	0
L2.1 L4.2	4.5424	0
L3.1 L5.2	4.5419	0

On remarque que le nombre qui vérifié de locuteurs différent reconnais est 40/40 locuteurs donc le taux total de vérification est 100 %

$$\text{Taux global d'identification} = \frac{\text{le nombre de locuteurs reconnus}}{\text{le nombre de locuteurs totale}} \times 100$$

On remarque que le nombre qui vérifié de même locuteurs différent 30 / le nombre qui vérifié de locuteurs différent 40. donc le Taux global de vérification est 75 %.

On définit :

- **Le Taux de Bonne Acceptation (TBA %) par**

$$\text{Taux de bonne Acceptation} = \frac{\text{le nombre de locuteurs accepter}}{\text{le nombre du même locuteurs}} \times 100$$

- **Le Taux de Faux Rejet (TFR %) par**

$$\text{Taux de Faux Rejet} = \frac{\text{le nombre de locuteurs rejeter}}{\text{le nombre de locuteurs différents}} \times 100$$

Durant nos expériences de VAL, nous avons utilisé plusieurs types de paramètres d'entrée (du réseau) : nous avons utilisé le même locuteur, le locuteur différent.

Les tableaux ci dessous résument les taux global d'acceptation et rejet :

Tableau 4.3 : Taux global de vérification.

Réponse correct	40
Réponse faux	30
Taux global	75%

Tableau 4.4 : Taux de vérification obtenus en utilisant le même locuteur.

	locuteurs
Locuteur accepté	0/30
Locuteur rejeté	30/30

Tableau 4.5 : Taux de vérification obtenus en utilisant le différent de locuteur.

	locuteurs
Locuteur accepté	40/40
Locuteur rejeté	0/40

Conclusion

Dans cette étude, nous avons essayé d'améliorer les performances du réseau de machine à vecteur de support en jouant sur les paramètres de sortie de ce dernier

Par conséquent, le choix des paramètres et des seuils initiaux du réseau possède un rôle très important en vérification automatique du locuteur. Il s'en suit qu'un choix judicieux doit être fait avant toute tentative de VAL.

Finalement, nous avons pu concevoir durant un système de VAL à base de SVM et assez sécurisé, présentant un taux de vérification pouvant atteindre 75%.