

Processus d'extraction des règles floues pour la classification phonémique Etude du corpus de parole TIMIT

Dorra Ben Ayed Mezghani*,** — Nouredine Ellouze*

* UR : Signal, Image, Reconnaissance de Formes ; Groupe : Reconnaissance Vocale ; Dépt. GénieElectrique
Ecole Nationale d'Ingénieurs de Tunis
BP-37 Campus Universitaire 1002 Tunis - Tunisie
Dorra.mezghani@isi.rnu.tn, Dorrainsat@yahoo.fr
N.Ellouze@enit.rnu.tn

** Institut Supérieur d'Informatique
Dépt. Génie Logiciel et systèmes d'informations
Abou Raihane Bayrouni, 2080 l'Ariana - Tunisie

RÉSUMÉ Plusieurs techniques de classification et de décision ont été élaborées. La création des règles floues, pouvant servir à une bonne classification, constitue une difficulté majeure pour ces techniques. C'est dans cette perspective que cet article s'intéresse à proposer une nouvelle approche de processus d'extraction des règles floues pour la classification phonémique.

MOTS-CLÉS classifieurs flous, extraction de règles floues, fonction d'appartenance, hyperbox d'activation, hyperbox d'inhibition, reconnaissance phonémique.

ABSTRACT Many classification and decision techniques are elaborated and discussed. Creating fuzzy rules automatically is a one of difficulties for these techniques. In this work we developed a new method for extracting fuzzy rules directly from numerical data for phoneme classification.

KEYWORDS fuzzy classifiers, fuzzy rule extraction, membership function, activation hyperbox, inhibition hyperbox, phoneme recognition.

1. Introduction

Depuis l'apparition de la logique floue, de nombreux travaux s'intéressent à cette nouvelle logique (Zadeh et al., 1978). Le résultat de cet engouement est l'apparition de certaines approches floues. Quelques une ont hérité de celles développées en logique classique telque fuzzy c-means, k-plus proches voisins flous, d'autres, se sont basées que sur les principes de l'univers du flou telque fuzzy pattern matching, règles et relations floues.

La logique floue n'est pas considérée comme une science exacte. Par conséquent, elle est plus souple (Klir *et al.*, 1995), plus rapide (Abe *et al.*, 1999) et plus simple (Thawonmas *et al.*, 1997) comparée aux réseaux de neurones. Cependant, la construction des règles floues reste encore une tâche complexe vu que cette tâche est toujours consultée par nos experts humains. L'acquisition des connaissances automatiquement par des systèmes flous à partir des données numériques a été le centre de plusieurs recherches durant ces dernières années (Wang *et al.*, 1992) (Jang, 1993) (Chiu, 1994) (Abe *et al.*, 1995a) (Abe *et al.*, 1995b).

Les travaux en reconnaissance de la parole s'intéressent de nos jours de plus en plus à la logique floue qui exploite le raisonnement et l'expertise de l'être humain. C'est à ce titre que ce travail

s'intéresse à l'étude de problème de l'extraction et de la modélisation automatique des données phonémiques par apprentissage. Nous proposons un système nommé SERF permettant d'une part la génération de base de règles floues, et permettant d'autre part la classification d'un nouvel échantillon. Le processus d'extraction de règles floues, nommé ERF, est basé sur le principe de zone ou hyperbox d'activation et d'hyperbox d'inhibition. Les règles ainsi obtenues seront transmises par la suite au processus de classification, nommé CERF, pour identifier les nouveaux échantillons grâce aux calculs des degrés d'appartenance selon la base des règles. L'échantillon à classer est attribué à la classe du plus grand degré d'appartenance.

Ce papier est organisé comme suit : La première partie a pour but de présenter l'approche méthodologique d'extraction des règles floues. L'idée principale de cette approche est d'utiliser le chevauchement entre les classes afin de créer les règles de classification. Pour atteindre cet objectif, tout d'abord à l'aide d'un exemple simple, les notions d'hyperbox d'activation et d'hyperbox d'inhibition ont été illustrées. Ensuite, les différents types de chevauchement qui interviennent dans le calcul des zones d'intersection sont énumérés. Et enfin, l'aspect de construction des règles d'inférences et les formules de calculs des degrés d'appartenance sont démontrés. La deuxième partie a pour but de présenter la mise en œuvre du système SERF. Le système proposé est subdivisé en trois principales phases liées les unes aux autres. La première est la phase de préparation des échantillons (ou individus) d'apprentissage et de test du corpus de parole TIMIT. La deuxième phase de cette partie représente l'extraction des règles floues ERF. Un classifieur CERF est ensuite appelé pour identifier la classe appropriée selon les degrés d'appartenances. La dernière partie aborde l'étude expérimentale du système SERF sur les bases de test du corpus de parole TIMIT. Les résultats obtenus sont évalués tout en dégageant les ambiguïtés de la mauvaise classification apparues. Des suggestions sont proposées comme perspectives à cet effet afin de pallier les problèmes rencontrés et d'augmenter par conséquent les taux de bonnes classifications.

2. Processus d'extraction des règles floues

Le processus d'extraction des règles floues se base sur le principe de zone appelé hyperbox (Abe *et al.*, 1999). Nous définissons tout d'abord quelques paramètres essentiels à la construction des hyperbox d'activation et d'inhibition. Les paramètres sont : m est la dimension du vecteur d'entrée, n est le nombre de classes, $x = (x_1, x_2, \dots, x_m)^T$ est un vecteur d'entrée (un échantillon), X_i est l'ensemble des échantillons appartenant à la classe i avec $i=1, \dots, n$ et $X = \cup X_i$ représente l'ensemble des échantillons avec $i=1, \dots, n$. On parlera également de l'Hyperbox d'activation qui est la région maximale contenant les échantillons appartenant à une classe spécifique et de l'Hyperbox d'inhibition qui est la région maximale d'intersection entre les deux hyperbox d'activation de même niveau (voir figure 1).

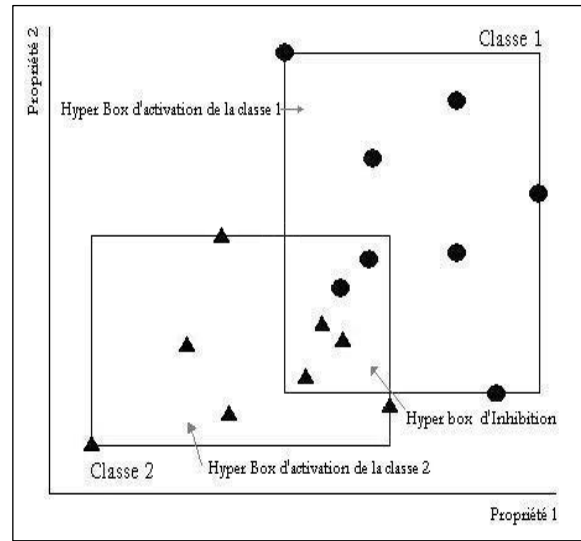


Figure 1. Hyperbox d'activation et hyperbox d'inhibition

2.1. Généralisation de l'hyperbox d'activation

Un hyperbox d'activation de niveau 1, noté $A_{ii}(1)$, est défini comme étant la région maximale contenant les échantillons de l'ensemble X_i (appartiennent à la classe i) :

$$A_{ii}(1) = \{x \in G / v_{ik}(1) \leq x_k \leq V_{ik}(1), k = 1, \dots, n\} \quad [1]$$

Où x_k : le $k^{\text{ième}}$ élément du vecteur d'entrée x .

$v_{ik}(1)$: la valeur minimale de x_k pour tout $x \in X_i$.

$V_{ik}(1)$: la valeur maximale de x_k pour tout $x \in X_i$.

Si les hyperboxs d'activation $A_{ii}(1)$ et $A_{jj}(1)$ ne se chevauchent pas, on obtient les règles floues de niveau 1 suivantes :

$$\text{Si } x \text{ est dans } A_{ii}(1) \text{ alors } x \text{ appartient à la classe } C_i \quad [2]$$

$$\text{Si } x \text{ est dans } A_{jj}(1) \text{ alors } x \text{ appartient à la classe } C_j \quad [3]$$

Par contre, s'il y a un chevauchement entre les hyperboxs $A_{ii}(1)$ et $A_{jj}(1)$, on résoud ce chevauchement d'une manière récursive par la définition d'une région d'inhibition de niveau 1, notée $I_{ij}(1)$. Dans ce cas, on obtient les règles floues de niveau 1 suivantes :

$$\text{Si } x \text{ est dans } A_{ii}(1) \text{ et } x \text{ est hors } I_{ij}(1) \text{ alors } x \text{ appartient à la classe } C_i \quad [4]$$

$$\text{Si } x \text{ est dans } A_{jj}(1) \text{ et } x \text{ est hors } I_{ij}(1) \text{ alors } x \text{ appartient à la classe } C_j \quad [5]$$

2.2. Généralisation de l'hyperbox d'inhibition

L'hyperbox d'inhibition est la région maximale d'intersection entre les deux hyperbox d'activation de même niveau (voir figure 1).

$$I_{ij}(l) = \{x \in G / w_{ijk}(l) \leq x_k \leq W_{ijk}(l), k = 1, \dots, n\} \quad [6]$$

Où x_k : le k ème élément du vecteur d'entrée x .

$w_{ijk}(l)$: la valeur minimale de la région d'intersection par rapport à la k ème propriété.

$W_{ijk}(l)$: la valeur maximale de la région d'intersection par rapport à la k ème propriété.

On peut facilement constater que $I_{ij}(l) = I_{ji}(l)$.

2.3. Calcul des limites de l'hyperbox d'inhibition

Selon la répartition des échantillons appartenant aux deux classes en question, on distingue différentes limites de l'hyperbox d'inhibition. De ce fait quatre types de chevauchement sont à étudier : chevauchement à gauche, chevauchement à droite, chevauchement interne et chevauchement externe.

Le type de chevauchement s'applique par classe et selon une seule propriété. La construction de l'hyperbox d'inhibition dépend du type de chevauchement. D'où, l'importance de cette distinction. Dans ce qui suit nous aurons besoin de quelques paramètres :

- l : le $l^{\text{ème}}$ niveau de chevauchement,
- $i' = j$ pour $l=1$ et $i' = i$ pour $l \geq 2$,
- $j' = i$ pour $l=1$ et $j' = j$ pour $l \geq 2$.

2.3.1 Chevauchement à gauche

Un chevauchement est dit de type *chevauchement à gauche* (voir figure 2) si l'inéquation suivante est vérifiée :

$$v_{jjk}(l) \leq v_{iik}(l) \leq V_{jjk}(l) \leq V_{iik}(l) \quad [7]$$

Les limites de niveau l , $w_{ijk}(l)$ et $W_{ijk}(l)$, de l'hyperbox d'inhibition selon une propriété sont définies par :

$$\begin{cases} w_{ijk}(l) = v_{ij'k}(l) \\ W_{ijk}(l) = V_{j'i'k}(l) \end{cases} \quad [8]$$

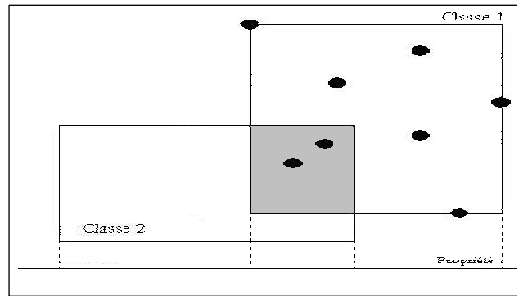


Figure 2. *Chevauchement à gauche*

2.3.2 Chevauchement à droite

Un chevauchement est dit de type *chevauchement à droite* si l'inéquation suivante est vérifiée :

$$v_{iik}(1) \leq v_{jjk}(1) \leq V_{iik}(1) \leq V_{jjk}(1) \quad [9]$$

Les limites de niveau 1 (voir figure 3), $w_{ijk}(l)$ et $W_{ijk}(l)$, de l'hyperbox d'inhibition selon une propriété sont définies par :

$$\begin{cases} w_{ijk}(l) = v_{j'k}(l) \\ W_{ijk}(l) = V_{ij'k}(l) \end{cases} \quad [10]$$

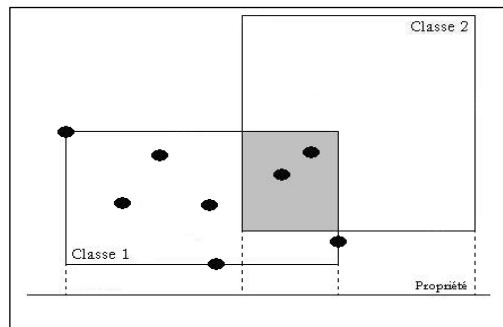


Figure 3. *Chevauchement à droite*

2.3.3 Chevauchement interne

Un chevauchement est dit de type *chevauchement interne* si l'inéquation suivante est vérifiée:

$$v_{jjk}(1) \leq v_{iik}(1) \leq V_{iik}(1) \leq V_{jjk}(1) \quad [11]$$

Les limites de niveau 1 (voir figure 4), $w_{ijk}(l)$ et $W_{ijk}(l)$, de l'hyperbox d'inhibition selon une propriété sont définies par :

$$\begin{cases} w_{ijk}(l) = v_{ij'k}(l) \\ W_{ijk}(l) = V_{ij'k}(l) \end{cases} \quad [12]$$

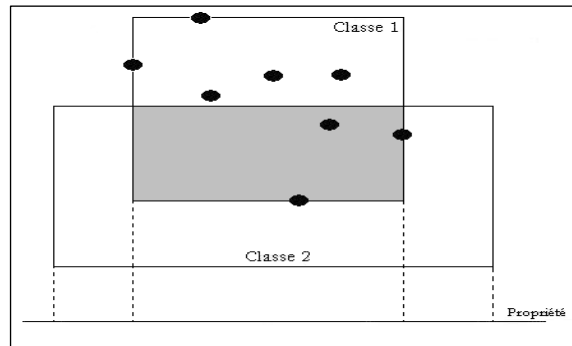


Figure 4. *Chevauchement interne*

2.3.4 Chevauchement externe

Un chevauchement est dit de type *chevauchement externe* si l'inéquation suivante est vérifiée :

$$v_{iik}(1) \leq v_{jjk}(1) \leq V_{jjk}(1) \leq V_{iik}(1) \quad [13]$$

Les limites de niveau 1 (voir figure 5), $w_{ijk}(l)$ et $W_{ijk}(l)$, de l'hyperbox d'inhibition selon une propriété sont définies par :

$$\begin{cases} w_{ijk}(l) = v_{ji'k}(l) \\ W_{ijk}(l) = V_{ji'k}(l) \end{cases} \quad [14]$$

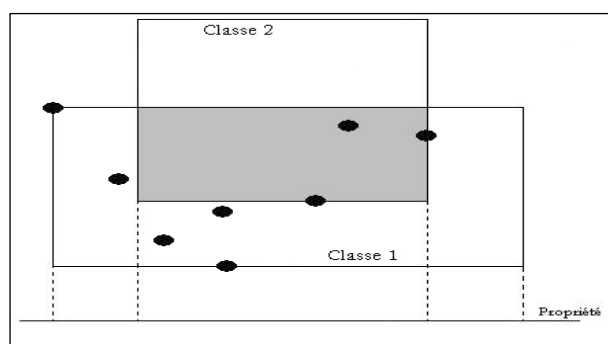


Figure 5. *Chevauchement externe*

2.4. Expansion de l'hyperbox d'inhibition

L'hyperbox d'inhibition, précédemment défini présente une défaillance majeure. Tout échantillon se trouvant à sa frontière peut ne pas appartenir à l'une des deux classes. Afin de surmonter ce problème, on élargie l'hyperbox d'inhibition original $I_{ij}(I)$ associé à $A_{ii}(I)$ et $A_{jj}(I)$. Ce nouveau hyperbox est noté hyperbox d'inhibition étendu par $J_{ij}(I)$, défini par :

$$J_{ij}(I) = \{x \in G / u_{ijk}(I) \leq x_k \leq U_{ijk}(I), k = 1, \dots, n\} \quad [15]$$

2.4.1. Hyperbox d'inhibition étendu

Les hyperboxes d'inhibition étendus (voir figure 6) pour $A_{ij}(I)$ et $A_{ji}(I)$ sont $J_{ij}(I)$ et $J_{ji}(I)$. Ces deux régions sont distinctes ($J_{ij}(I) \neq J_{ji}(I)$).

Cette nouvelle idée met en cause le degré d'appartenance des échantillons se trouvant à sa surface. Une erreur de mesure ou un manque d'appréciation peut influencer sur les coordonnées de cet échantillon. Pour calculer les limites de l'hyperbox d'inhibition étendu, nous pesons les paramètres d'initialisation suivants :

- l : le 1^{er} niveau de chevauchement.
- $i' = j$ pour $l=1$ et $i' = i$ pour $l \geq 2$.
- $j' = i$ pour $l=1$ et $j' = j$ pour $l \geq 2$.

Comme lors du calcul de la zone d'inhibition, on distingue les mêmes types de chevauchement. L'utilisation de ces types est indispensable lors de la création des hyperboxes d'inhibition étendus.

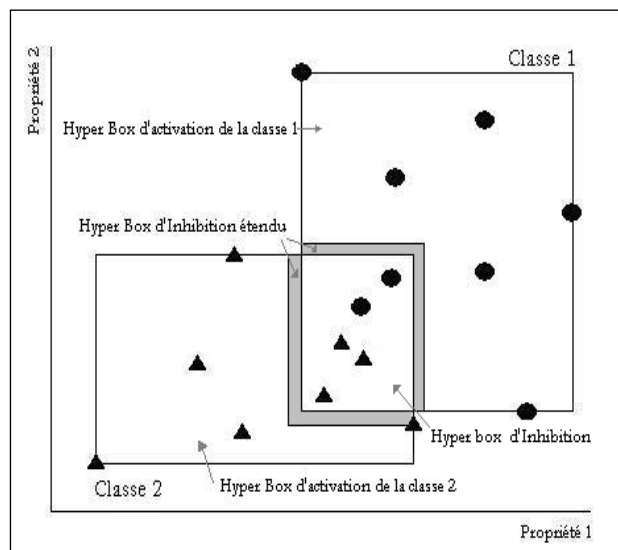


Figure 6. Hyperbox d'inhibition étendu

2.4.2. Chevauchement étendu à gauche

Un chevauchement est de type *chevauchement étendu à gauche* (voir figure 7) si l'inéquation suivante est vérifiée :

$$v_{ji^*k}(l) \leq v_{ij^*k}(l) \leq V_{ji^*k}(l) \leq V_{ij^*k}(l) \quad [16]$$

Les limites, $u_{ijk}(l)$ et $U_{ijk}(l)$, de l'hyperbox d'inhibition étendu selon une propriété sont définies par :

$$\begin{cases} u_{ijk}(l) = v_{ij^*k}(l) \\ U_{ijk}(l) = V_{ji^*k}(l) + \alpha(V_{ij^*k}(l) - V_{ji^*k}(l)) \end{cases} \quad [17]$$

Où α est un paramètre d'expansion tel que $0 < \alpha < 1$.

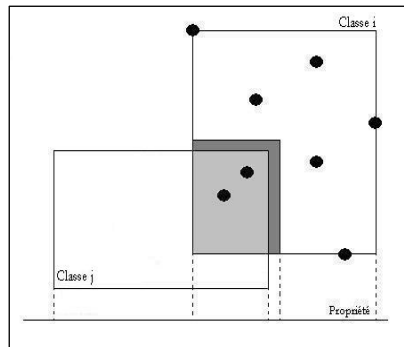


Figure 7. Chevauchement étendu à gauche

2.4.3. Chevauchement étendu à droite

Un chevauchement est dit de type *chevauchement étendu à droite* (voir figure 8) si l'inéquation suivante est vérifiée :

$$v_{ij^*k}(l) \leq v_{ji^*k}(l) \leq V_{ij^*k}(l) \leq V_{ji^*k}(l) \quad [18]$$

Les limites, $u_{ijk}(l)$ et $U_{ijk}(l)$, de l'hyperbox d'inhibition étendu selon une propriété sont définies par :

$$\begin{cases} u_{ijk}(l) = v_{ji^*k}(l) - \alpha(v_{ji^*k}(l) - v_{ij^*k}(l)) \\ U_{ijk}(l) = V_{ij^*k}(l) \end{cases} \quad [19]$$

Où α est un paramètre d'expansion tel que $0 < \alpha < 1$.

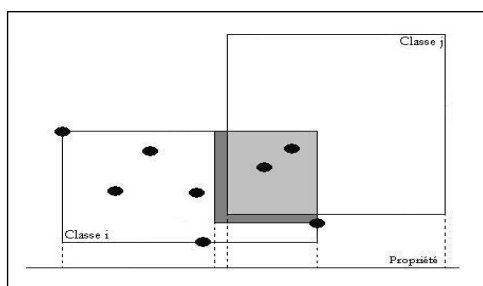


Figure 8. Chevauchement étendu à droite

2.4.4. Chevauchement étendu interne

Un chevauchement est dit de type *chevauchement étendu interne* (voir figure 9) si l'inéquation suivante est vérifiée:

$$v_{ji^*k}(I) \leq v_{ij^*k}(I) \leq V_{ij^*k}(I) \leq V_{ji^*k}(I) \quad [20]$$

Les limites, $u_{ijk}(l)$ et $U_{ijk}(l)$, de l'hyperbox d'inhibition étendu selon une propriété sont définies par :

$$\begin{cases} u_{ijk}(l) = v_{ij^*k}(l) \\ U_{ijk}(l) = V_{ij^*k}(l) \end{cases} \quad [21]$$

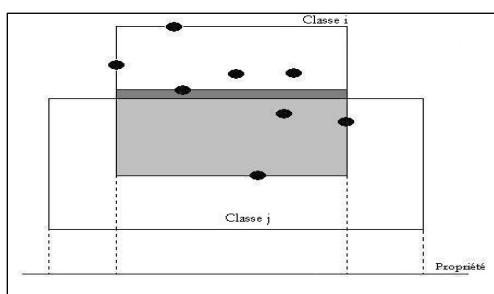


Figure 9. Chevauchement étendu interne

2.4.5. Chevauchement étendu externe

Un chevauchement est de type *chevauchement étendu externe* (voir figure 10) si l'inéquation suivante est vérifiée :

$$v_{ij^*k}(I) \leq v_{ji^*k}(I) \leq V_{ji^*k}(I) \leq V_{ij^*k}(I) \quad [22]$$

Les limites, $u_{ijk}(l)$ et $U_{ijk}(l)$, de l'hyperbox d'inhibition étendu selon une propriété sont définies par :

$$\begin{cases} u_{ijk}(I) = v_{ji'k}(I) - \alpha(v_{ji'k}(I) - v_{ij'k}(I)) \\ U_{ijk}(I) = V_{ji'k}(I) + \alpha(V_{ij'k}(I) - V_{ji'k}(I)) \end{cases} \quad [23]$$

Où α est un paramètre d'expansion tel que $0 < \alpha < 1$.

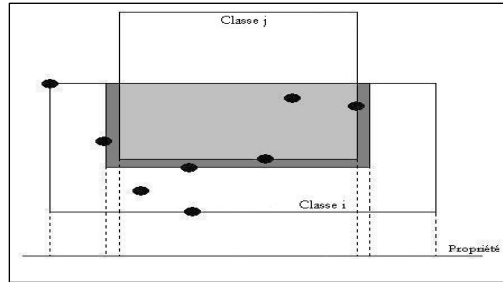


Figure 10. Chevauchement étendu externe

3. Construction des règles d'inférences floues

Le principe de construction des règles d'inférences floues se base sur le principe du processus d'extraction précédemment présenté.

Dans le cas où les deux hyperbois d'activation, ainsi définis, se chevauchent, on détermine une règle floue d'appartenance à la classe C_j de la manière suivante :

$$\text{Si } x \text{ est dans } A_{11}(1) \text{ et } x \text{ est hors } I_{12}(1) \text{ alors } x \text{ appartient à la classe } C_1 \quad [24]$$

De la même manière, on détermine la règle floue d'appartenance à la classe C_2 de la manière suivante :

$$\text{Si } x \text{ est dans } A_{22}(1) \text{ et } x \text{ est hors } I_{12}(1) \text{ alors } x \text{ appartient à la classe } C_2 \quad [25]$$

Cependant, dans le cas où les deux hyperbois d'activation ne se chevauchent pas, un échantillon ne peut appartenir à la classe C_j sauf s'il vérifie la nouvelle règle floue :

$$\text{Si } x \text{ est dans } A_{11}(1) \text{ alors } x \text{ appartient à la classe } C_1 \quad [26]$$

De même, pour qu'un échantillon appartienne à la classe C_2 il doit vérifier la règle suivante :

$$\text{Si } x \text{ est dans } A_{22}(1) \text{ alors } x \text{ appartient à la classe } C_2 \quad [27]$$

Supposons que $A_{ii}(I)$ et $A_{jj}(I)$ se chevauchent et soit $J_{ij}(I)$ la région d'inhibition étendue associée à ces deux régions. On définit la règle floue de niveau 1 associée à la classe C_i comme suit :

Si x est dans $A_{ii}(1)$ et x est hors $J_{ij}(1)$ alors x appartient à la classe C_i [28]

La règle [28] se traduit mathématiquement par la règle [29] comme suit :

Soient :

$$\text{Pour } x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad A_{ii}(1) = \begin{pmatrix} [v_{i1}, V_{i1}] \\ [v_{i2}, V_{i2}] \\ \vdots \\ [v_{im}, V_{im}] \end{pmatrix} \text{ et } J_{ij}(1) = \begin{pmatrix} [u_{j1}, U_{j1}] \\ [u_{j2}, U_{j2}] \\ \vdots \\ [u_{jm}, U_{jm}] \end{pmatrix}, \text{ on a:}$$

$$\text{Si } \begin{cases} x_1 \in [v_{i1}, V_{i1}] \\ x_2 \in [v_{i2}, V_{i2}] \\ \vdots \\ x_m \in [v_{im}, V_{im}] \end{cases} \text{ et } \begin{cases} x_1 \notin [u_{j1}, U_{j1}] \\ x_2 \notin [u_{j2}, U_{j2}] \\ \vdots \\ x_m \notin [u_{jm}, U_{jm}] \end{cases} \text{ alors } x \text{ appartient à la classe } C_i \quad [29]$$

De la même manière, on définit la règle de classification de niveau 1 à la classe C_j .

Si x est dans $A_{jj}(1)$ et x est hors $J_{ji}(1)$ alors x appartient à la classe C_j [30]

De la même manière, la règle [30] se traduit mathématiquement par la règle [31] comme suit :

Soient :

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad A_{jj}(1) = \begin{pmatrix} [v_{j1}, V_{j1}] \\ [v_{j2}, V_{j2}] \\ \vdots \\ [v_{jm}, V_{jm}] \end{pmatrix} \text{ et } J_{ji}(1) = \begin{pmatrix} [u_{i1}, U_{i1}] \\ [u_{i2}, U_{i2}] \\ \vdots \\ [u_{im}, U_{im}] \end{pmatrix}, \text{ on a:}$$

$$\text{Si } \begin{cases} x_1 \in [v_{j1}, V_{j1}] \\ x_2 \in [v_{j2}, V_{j2}] \\ \vdots \\ x_m \in [v_{jm}, V_{jm}] \end{cases} \text{ et } \begin{cases} x_1 \notin [u_{i1}, U_{i1}] \\ x_2 \notin [u_{i2}, U_{i2}] \\ \vdots \\ x_m \notin [u_{im}, U_{im}] \end{cases} \text{ alors } x \text{ appartient à la classe } C_j \quad [31]$$

Il est à noter que certains échantillons de la classe i appartiennent à la fois à $A_{ii}(1)$ et $J_{ij}(1)$, on crée l'hyperbox d'activation de niveau 2, noté $A_{ij}(2)$ qui est inclus dans la région d'inhibition étendue du niveau précédent, $J_{ij}(1)$. De même, on crée l'hyperbox d'activation de niveau 2 $A_{ji}(2)$.

$$A_{ij}(2) = \{x \in G / v_{ijk}(2) \leq x_k \leq V_{ijk}(1), k = 1, \dots, n\} \quad [32]$$

Où x_k : le $k^{\text{ème}}$ élément du vecteur d'entrée x ,

$v_{ijk}(2)$: la valeur minimale de x_k pour tout $x \in X_i$ et $x \in J_{ij}(1)$,

$V_{ijk}(2)$: la valeur maximale de x_k pour tout $x \in X_i$ et $x \in J_{ij}(1)$,

et $u_{ijk}(1) \leq v_{ijk}(2) \leq V_{ijk}(2) \leq U_{ijk}(1)$.

Il est à signaler aussi que s'il n'y a qu'un seul hyperbox d'activation ou si les deux hyperboxs d'activation ne se chevauchent pas, on définit les règles floues [33] et [34] de niveau 2 comme suit :

Si x est dans $A_{ij}(2)$ alors x appartient à la classe C_i [33]

Si x est dans $A_{ji}(2)$ alors x appartient à la classe C_j [34]

.

Par contre, s'il y a un chevauchement, on construit les deux hyperboxs d'inhibition de niveau 2. Un hyperbox pour chaque classe tel que :

$J_{ij}(2) = \{x \in G / w_{ijk}(2) \leq x_k \leq W_{ijk}(1), k = 1, \dots, n\}$ [35]

Où $v_{ijk}(2) \leq w_{ijk}(2) \leq W_{ijk}(2) \leq V_{ijk}(2)$

$J_{ji}(2) = \{x \in G / w_{jik}(2) \leq x_k \leq W_{jik}(1), k = 1, \dots, n\}$ [36]

Où $v_{ijk}(2) \leq w_{ijk}(2) \leq W_{ijk}(2) \leq V_{ijk}(2)$

Ainsi, les règles floues de niveau 2 sont de la forme suivante :

Si x est dans $A_{ij}(2)$ et x est hors $J_{ij}(2)$ alors x appartient à la classe C_i [37]

Si x est dans $A_{ji}(2)$ et x est hors $J_{ji}(2)$ alors x appartient à la classe C_j [38]

Les règles floues de niveau supérieur à 2 peuvent être définies d'une manière similaire. La forme générale d'une règle floue, sans chevauchement, est alors :

Si x est dans $A_{ij^l}(1)$ alors x appartient à la classe C_i [39]

(Avec $j^l = i$ pour $l=1$ et $j^l = j$ pour $l \geq 2$)

Dans le cas d'un chevauchement, une règle floue prend la forme suivante :

Si x est dans $A_{ij^l}(1)$ et x est hors $J_{ij^l}(1)$ alors x appartient à la classe C_i [40]

(Avec $j' = i$ pour $l=1$ et $j' = j$ pour $l \geq 2$)

Le processus récursif d'extraction des règles floues s'arrête si l'une des conditions suivantes est atteinte:

- $A_{j'k}(l) \cap A_{j''k}(l) = \emptyset$, pas de chevauchement entre les hyperboxes d'activation ;
- $A_{j'k}(l) = A_{j''k}(l) = I_{ij}(l-1)$, les deux hyperboxes d'activation sont confondus ;
- On atteint un seuil de chevauchement maximal fixé a priori.

4. Mise en oeuvre du système de classification phonémique

Nous proposons dans cette partie le système de classification phonémique à base d'extraction des règles floues, nommée SERF (voir figure 11). Il est décomposé en trois principales phases :

- Une phase de préparation des bases d'apprentissage et de test ;
- Une phase d'extraction des règles floues : le processus ERF;
- Une phase de classification : le processus CERF.

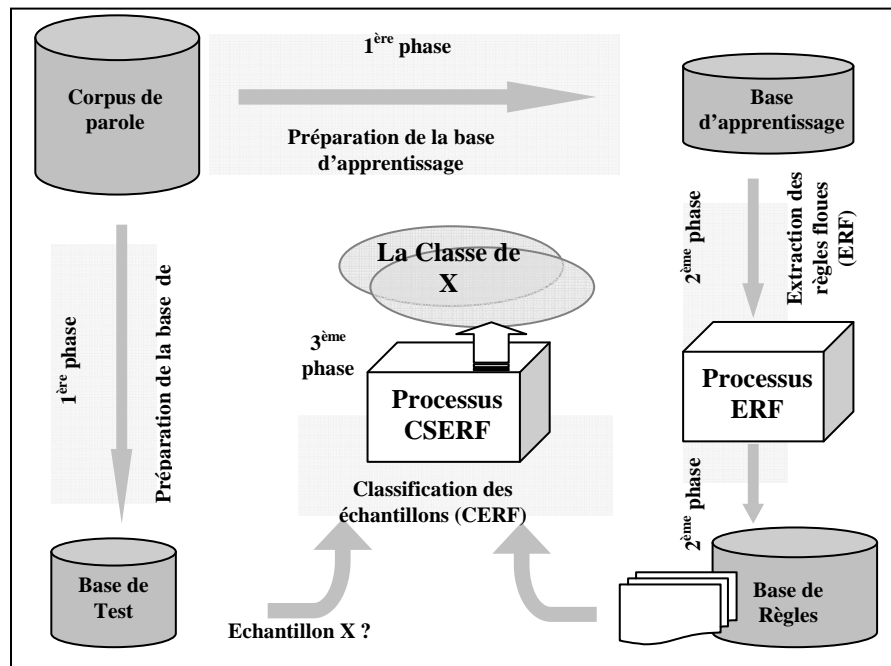


Figure 11. Structure globale du système SERF

4.1. Préparation des bases d'apprentissage et de test

4.1.1 Structure

L'extraction de connaissance est un but et une nécessité afin d'effectuer l'apprentissage (Fisher *et al.*, 1986) (Prade *et al.*, 1987). Il faut donc tout d'abord organiser l'information représentant l'échantillon à expérimenter. Le processus d'extraction manipule une base numérique pour la

construction des règles floues. Cette base est dite *base d'apprentissage*. La base d'apprentissage est structurée de la manière suivante : m colonnes représentant m propriétés des échantillons dont une colonne représente l'identificateur numérique de la classe à laquelle un échantillon appartient. En fait, la base d'apprentissage est un tableau de n lignes (les échantillons) et $m+1$ colonnes (les propriétés et identificateur de la classe).

Le tableau 1 indique la structure de la base d'apprentissage et la base de test que le processus d'extraction des règles floues et le processus de classification peuvent manipuler. Chaque échantillon (Ligne) doit être évalué par rapport à toutes les propriétés. Une propriété vide n'est pas acceptée.

Propriété 1	Propriété 2	Propriété m	Classe
$E1_{P1}$	$E1_{P2}$		$E1_{Pm}$	1
⋮	⋮		⋮	⋮
⋮	⋮		⋮	⋮
En_{P1}	En_{P2}		En_{Pm}	t

Tableau 1. Structure des bases d'apprentissage et de test

4.1.2 Corpus de sons numérique

Nous avons utilisé le corpus de sons TIMIT dans le but d'étudier et d'évaluer le système de classification proposé. Cette base a été constituée pour illustrer au mieux la variabilité acoustique de l'anglais américain. Elle contient une segmentation phonémique de référence (étiquetage fin) qui simplifie l'apprentissage initial des modèles phonémiques.

Une représentation du signal numérisé, moins redondante que le signal acoustique lui-même, tout en préservant les informations discriminantes est une phase importante pour l'identification du signal. Le choix d'une technique de paramétrisation revêt une importance fondamentale car elle conditionne l'efficacité des systèmes de reconnaissance. Nous avons utilisé une technique de paramétrisation cepstrale par banc de filtres selon une échelle fréquentielle *mel* en vue de générer une base de données numériques et étiquetées (Ben Ayed *et al.*, 2001).

Nous avons organisé le corpus de sons TIMIT sous une forme d'une base numérique hiérarchique de sorte qu'une première partie est réservée comme base d'apprentissage et une deuxième partie comme une base de test. Chacune des bases, d'apprentissage et de test, est divisée en sept groupes phonétiquement homogènes constituant ainsi des sous bases de classes phonémiques.

Chaque sous-base de ce groupe phonétique est organisée en tables. Chaque table caractérise un ensemble de phonèmes de même type et possède un ensemble de descripteurs qui contiennent :

- des informations relatives aux locuteurs,
- des informations relatives à son identification,
- des indications sur sa position dans la phrase,

- le signal audio numérisé,
- des valeurs paramétriques caractérisant le phonème.

4.1.3. Structure de la base à paramètre MFCC

Le signal vocal est analysé en fenêtres dont la longueur satisfait le principe de stationnarité. La longueur de la fenêtre est de 256 points, ce qui donne une durée de 16 ms (Reynolds, 1994). Un recouvrement de moitié, soit de 128 points, est choisi et chaque tranche est multipliée par une fenêtre de Hamming de 256 points pour réduire l'effet de Gibbs. Le spectre de Fourier est calculé sur une fenêtre d'analyse répartie sur un banc de 24 filtres selon une échelle de fréquence *mel*. Les *mel-frequency cepstral coefficients* (MFCC) (Davis et al., 1980), (Haton et al., 1991), s'obtiennent par une transformée en cosinus discrète inverse du spectre d'amplitude modifié.

Dans chaque fenêtre du phonème de la base, les coefficients *mel* cepstre sont calculés. La base de données obtenue est représentée sous la forme d'une matrice à valeurs réelles. Chaque ligne représente les 12 coefficients *mel* cepstre d'une fenêtre d'un phonème donné. À chaque ligne de la matrice est associée une étiquette comprenant le libellé du phonème, le libellé de la macro classe, le code de la phrase et le code du locuteur.

La structuration des données est fondée sur une hiérarchie, ce qui signifie que les descripteurs d'un niveau supérieur restent valables pour les niveaux inférieurs (voir figure 12). D'après cette organisation et selon la méthode de paramétrisation appliquée, nous avons élaboré :

Une base de données phonémique des paramètres MFCC qui répond à la structure du tableau 1 constituée de :

- sept sous bases de locuteurs masculins des différents groupes phonémiques,
- sept sous bases de locuteurs féminins des différents groupes phonémiques.

Il est à noter que les sept sous bases sont nommées de la sorte : base des voyelles, base des semi-voyelles, base des fricatives, base des affricates, base des plosives, base des nasales et la base des autres.

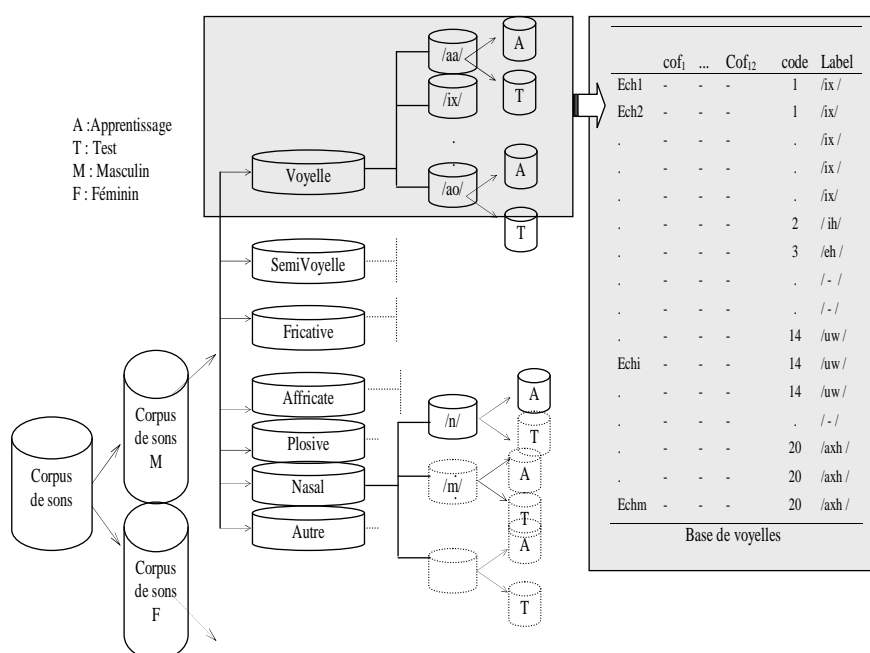


Figure 12. Structure de la Base MFCC

Le tableau 2 présente la codification des classes phonémiques constituant les bases de test et d'apprentissage. Ces codifications seront utilisées aussi bien par les bases d'apprentissage que les bases de test des locuteurs féminins et masculins.

Identifiant	Phonème	Identifiant	Phonème
1	Voyelle	4	Plosif
2	Semi-voyelle	5	Affriqué
3	Fricatif	6	Nasale

Tableau 2. Codification des classes phonémiques

4.2. Extraction des règles floues :ERF

4.2.1 Processus ERF

Partons d'une base d'apprentissage à paramètres réels, nous appliquons le processus d'extraction des règles floues mis en œuvre selon le principe proposé dans le paragraphe 2 tout en adaptant d'une part, la technique d'expansion de l'hyperbox d'inhibition par l'ajustement du paramètre α . Et d'autre part, en adoptant le seuil maximal de chevauchement fixé a priori comme étant une condition d'arrêt du processus.

A la sortie de ce processus, on obtient une base de règles floues. La figure 13 décrit la phase d'extraction de règles floues ERF.

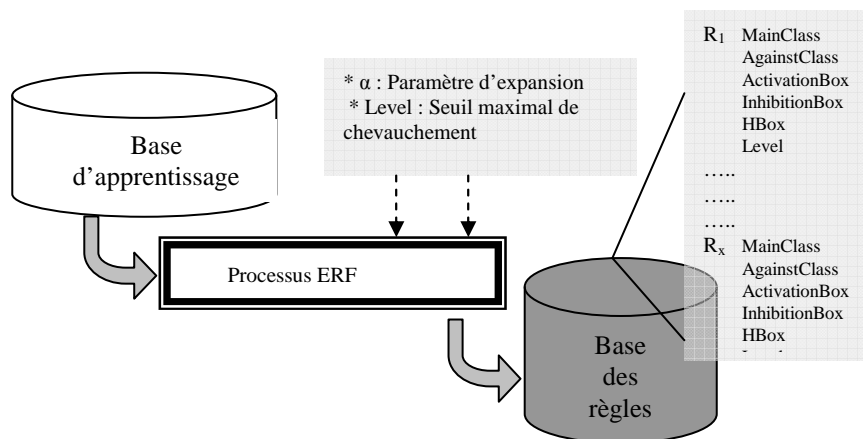


Figure 13. Vue générale du processus d'extraction des règles floues (ERF)

Chaque règle est une structure composée de : (voir Figure 14)

- MainClass : contient l'identifiant de la classe principale.
- AgainstClass : contient l'identifiant de la classe qui se chevauche avec la classe principale.

- **ActivationBox** : détermine le domaine de l'hyperbox d'activation. Il caractérise l'ensemble A_{ij} défini en [1].

- **Min** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur minimale $v_{ik}(l)$ des échantillons de A_{ij} par rapport à une propriété $k \in [1, n]$.

- **Max** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur maximale $v_{ik}(l)$ des échantillons de A_{ij} par rapport à une propriété $k \in [1, n]$.

- **InhibitionBox** : détermine le domaine de l'hyperbox d'inhibition. Il caractérise l'ensemble J_{ij} défini en [15].

- **Min** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur minimale $u_{ik}(l)$ des échantillons de J_{ij} par rapport à une propriété $k \in [1, n]$.

- **Max** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur maximale $U_{ik}(l)$ des échantillons de J_{ij} par rapport à une propriété $k \in [1, n]$.

- **HBox** : détermine le domaine de l'hyperbox d'inhibition. Il caractérise l'ensemble H_{ij} .

- **Min** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur minimale des échantillons de H_{ij} par rapport à une propriété $k \in [1, n]$.

- **Max** : est un vecteur de dimension $[1, n]$. Chaque colonne contient la valeur maximale des échantillons de H_{ij} par rapport à une propriété $k \in [1, n]$.

- **Level** : contient le niveau de chevauchement.

RuleMatrix (1)		
MainClass		1
AgainstClass		2
ActivationBox		
	Min	[0.17 0.33 0.35 0.36 0.00 0.00 0.17]
	Max	[0.18 0.98 1.00 0.92 0.79 0.71 0.85]
InhibitionBox		
	Min	[0.17 0.33 0.35 0.36 0.01 0.01 0.17]
	Max	[0.86 0.98 0.99 0.92 0.78 0.71 0.85]
HBox		
	Min	[0.17 0.00 0.00 0.00 0.01 0.01 0.00]
	Max	[0.86 0.98 0.99 0.00 0.78 0.71 0.85]
Level		2

Figure 14. Structure et exemple d'une règle floue créée par ERF

L'exemple de la figure 14 présente l'organisation d'une règle floue construite par le processus ERF. Le nombre de propriétés est égal à 7. Cette règle décrit un chevauchement au niveau 2 (*Level*) entre la classe 1 (*MainClass*) et la classe 2 (*AgainstClass*). Dans ce cas, tous les hyperboxs (*ActivationBox*, *InhibitionBox* et *HBox*) sont remplis. Par contre, dans le cas où il n'y a plus de chevauchement entre les classes *MainClass* et *AgainstClass*, seul l'*ActivationBox* serait rempli.

4.2.2 Algorithme du processus ERF

L'algorithme 1 décrit le processus d'extraction des règles floues.

```
Début
data                                     % charger la base
NbClass                                  % nombre de classes
ruleMatrix = ''                          % matrice des règles
Pour i=1 : NbClasse
dataFirstClass = data(i)                 % la base des échantillons de la classe i
Pour j=(i+1) : NbClasse
dataSecondClass = data(j)                % la base des échantillons de la classe j
arrêtBoucle = 0                          % indicateur de boucle
level = 0                                 % Niveau de chevauchement
    Tant que (arrêtBoucle = 0)
        Si (chevauchement)
            % Création des hyperbox d'inhibition
            InhibFirstClass = createInhib (dataFirstClass ,  $\alpha$ )
            InhibSecondClass = createInhib (dataSecondClass,  $\alpha$ )
            % Création des H-box
            HFirstClass = createHbox (dataFirstClass)
            HSecondClass = createHbox (dataSecondClass)
            % Elimination des échantillons n'appartenant pas à l'inhibition box
            dataFirstClass=EliminateSample(dataFirstClass, InhibFirstClass)
            dataSecondClass=EliminateSample(dataSecondClass, InhibSecondClass)
            Si (size(dataFirstClass) = 0 ou size(dataSecondClass) = 0 )
                ArrêtBoucle = 1           % Arrêt de la boucle
            Fin Si
        % Construction de la règle
        NewFirstRule = createRule (i,j, InhibFirstClass, HFirstClass, level)
        NewSecondRule = createRule (j, i, InhibSecondClass, HSecondClass, level)
        Si
            ArrêtBoucle = 1               % Arrêt de la boucle
            % Construction de la règle
            NewFirstRule = createRule (i,j, '', '', level)
            NewSecondRule = createRule (j, i, '', '', level)
        Fin Si
        % Ajouter les règles à la matrice des règles
        ruleMatrix = AddRule (ruleMatrix, NewFirstRule)
        ruleMatrix = AddRule (ruleMatrix, NewSecondRule)
    Fin Tant que
Fin Pour j
Fin Pour i
Save ruleMatrix                          % Sauvegarder la matrice des règles
Fin
```

Algorithme 1. Algorithme d'extraction de règles floues (ERF)

4.3 Classification d'un échantillon : CERF

4.3.1 Le processus CERF

La figure 15 illustre la structure générale du processus de classification (CERF). Etant donné, un échantillon X, le système calcule le degré d'appartenance de l'échantillon d'entrée par rapport toutes les règles données par le processus ERF. Le classifieur prend en charge par la suite de calculer le degré d'appartenance à chacune des classes mises en jeu pour enfin choisir la classe à laquelle l'échantillon pourrait appartenir.

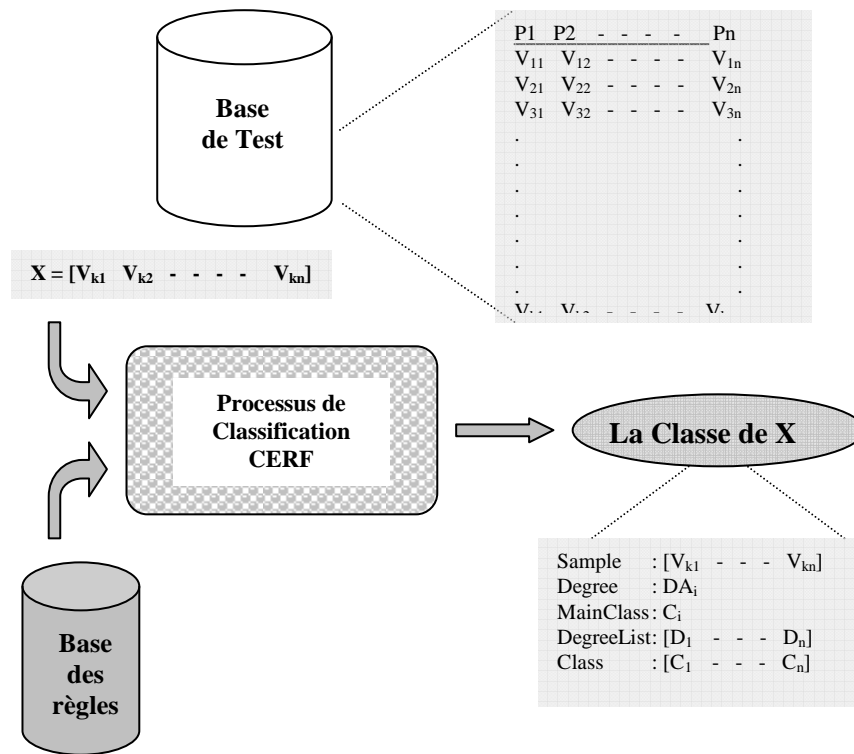


Figure 15. Vue générale du processus de classification (CERF)

4.3.2 Algorithme du processus CERF

L'algorithme 2 décrit le processus de classification adopté.

A la sortie de cet algorithme, nous obtenons une structure de données organisée de la manière suivante :

- **Sample** : contient un vecteur de dimension $[1, m]$, avec $m \geq n$, où n est le nombre de paramètres des données d'apprentissage. Ce vecteur représente l'échantillon à classer.
- **Degree** : est le degré d'appartenance à la classe *MainClass*. C'est le degré maximal par rapport à tous les degrés des autres classes.
- **MainClass** : est la classe à laquelle l'échantillon *Sample* est classé selon le CSERF.
- **DegreeList** : est un vecteur de dimension $[1, l]$, où l est le nombre de classes mises en jeu. Il contient les degrés par rapport à toutes les classes.
- **Class** : est un vecteur de dimension $[1, l]$, où l est le nombre de classes mises en jeu. Il contient les identifiants de toutes les classes organisés dans le même ordre que le *DegreeList*.

```

Début
ruleData = ruleMatrix           % récupération de la base des règles
ClassVector                     % la liste des classes
MeilleurDegré = 0               % Meilleur degré d'appartenance
Pour i = 1 : size(ClassVector, 2)
    % Degré d'appartenance à une classe
    DegréAllClass = 0
    ListRuleForClass = selectRule(ClassVector(i), ruleData)

    Pour j = 1 : size(ListRuleForClass, 2)
        % Degré d'appartenance à une classe par rapport à une classe
        DegréUneClass = 0
        ListRuleAgainstClass = SelectRule(ClassVector(j), ListRuleForClass)
        Pour k = 1 : size(ListRuleAgainstClass, 2)
            % Degré d'appartenance à une classe par niveau
            DegréParNiveau = CalculDegré(Sample, ListRuleAgainstClass(k))
            Si (DegréParNiveau < DegréUneClass)
                DegréUneClass = DegréParNiveau
            Fin Si
        Fin Pour k

        Si (DegréAllClass < DegréUneClass)
            DegréAllClass = DegréUneClass
        Fin Si
    Fin Pour j
    Si (MeilleurDegré < DegréAllClass)
        MeilleurDegré = DegréAllClass
    Fin Si
Fin Pour i
Sauver MeilleurDegré           % sauvegarde du résultat de classification
Fin

```

Algorithme 2. Algorithme de classification (CERF)

4.3.3 Degré d'appartenance

La figure 16 illustre un exemple de la structure fournie à la fin du processus de classification. On retrouve dans cette structure l'échantillon X (*Sample*), son degré le plus élevé (*Degree*) lui correspond la classe (*MainClass*). On peut aussi avoir tous les degrés d'appartenance de cet échantillon par rapport à toutes les autres classes.

Degree_De_X								
Sample	[0.99	0.10	0.34	0.20	0.00	0.90	0.17]	
Degree	0.98							
MainClass	3							
DegreeList	[0.20	0.56	0.98	0.75	0.00	0.00	0.44]	
Class	[1	2	3	4	5	6	7]	

Figure 16. Exemple de résultat d'une classification

5. Etude expérimentale

5.1 Résultats du processus ERF

5.1.1. Variation du nombre de règles par rapport au paramètre d'expansion

Nous avons appliqué le processus d'extraction des règles floues sur les bases d'apprentissage phonémiques des locuteurs masculins et des locuteurs féminins. Le nombre des règles formulées est important et varie selon le paramètre d'expansion α . La figure 17 présente la variation du nombre de règles par rapport à α . Le seuil maximal de chevauchement est fixé à 300. On remarque que le nombre de règles augmente avec l'élargissement de la zone d'inhibition (l'augmentation de la valeur de α).

Ce résultat est tout à fait attendu, puisque les échantillons frontaliers se retrouvent à l'intérieur de la zone d'inhibition ce qui favorise la construction de nouvelles règles. On remarque aussi que le nombre de règles construites à partir de la base d'apprentissage des locuteurs masculins est toujours plus élevé que celui à partir de la base d'apprentissage des locuteurs féminins.

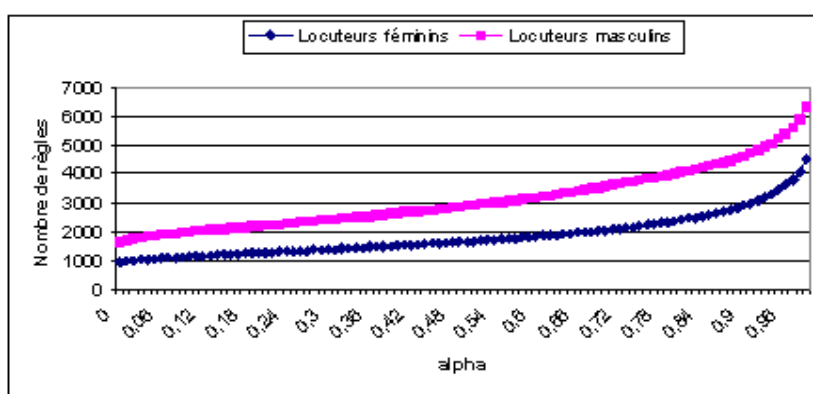


Figure 17. Variation du nombre de règles floues extraites par rapport à α

Les figures 18 et 19 montrent la variation du nombre de règles par classe suivant la valeur du degré d'élargissement de l'hyperbox d'inhibition selon les locuteurs masculins et les locuteurs féminins. On remarque que le nombre de règle par classe augmente d'une manière exponentielle.

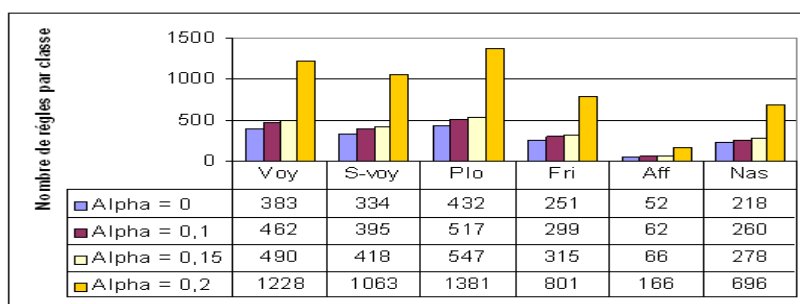


Figure 18. Variation du nombre de règles par classe phonémique pour différentes valeurs de α pour les locuteurs masculins

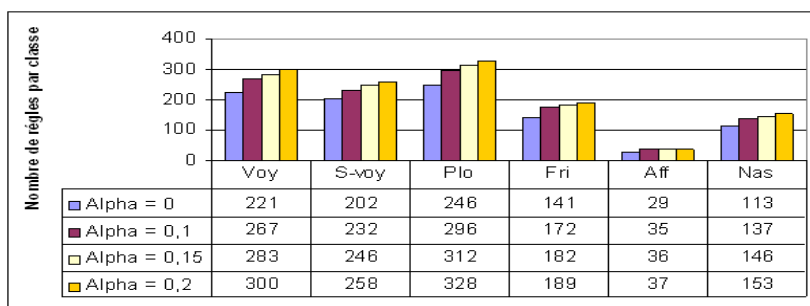


Figure 19. Variation du nombre de règles par classe pour différentes valeurs de α pour les locuteurs féminins

5.1.2 Variation du nombre de règles par rapport à la variation du niveau de chevauchement

Nous avons étudié également la variation du nombre de règles par rapport à la variation du niveau de chevauchement. En effet, nous avons varié le seuil maximal ainsi que la paramètre d'expansion α de 0.1 à 0.2 avec un pas de 0.01 et nous avons choisi de représenter les résultats obtenus pour $\alpha = 0$, $\alpha = 0.1$, $\alpha = 0.15$ et enfin pour $\alpha = 0.2$ (figure 20 et figure 21). On remarque que cette variation dépend de la valeur de α . En effet, le nombre de règles augmente avec l'augmentation du paramètre d'expansion α . Et vu que la base d'apprentissage des locuteurs masculins est plus importante que celle des locuteurs féminins, le nombre de règles pour le même α , pour le même nombre de seuil maximal de chevauchement est certes plus important.

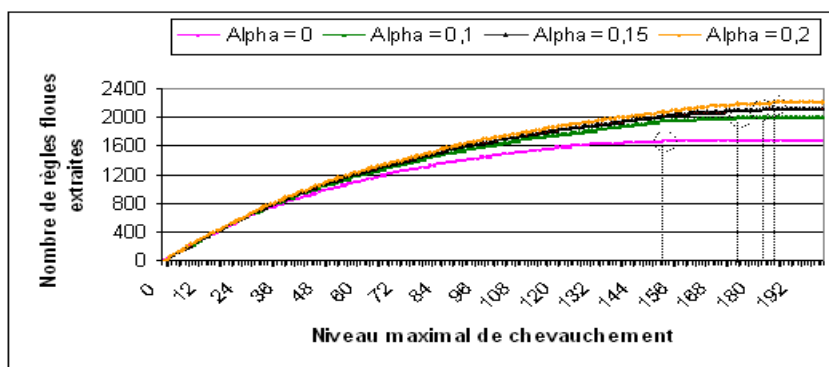


Figure 20. Variation du nombre de règles des masculins selon le seuil maximal pour différentes valeurs de α .

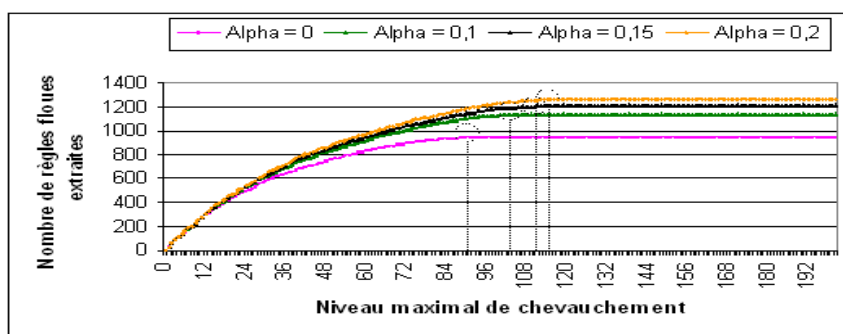


Figure 21. Variation du nombre de règle des féminins selon le seuil maximal pour différentes valeurs de α .

5.2 Résultats du processus de classification CERF

Après avoir construit la base des règles, nous entamons la phase de classification permettant la décision. Cette phase est importante, permettant d'avoir une idée sur l'efficacité de la méthode adoptée.

En introduisant tous les échantillons de la base de test des locuteurs féminins et masculins, nous répartissons les résultats de classifications obtenus dans une matrice de confusion. Autre que la matrice de confusion, nous notons également :

- le nombre d'échantillons correctement classifiés par classe phonémique (NBrBC).
- le nombre d'échantillons mal classifiés par classe phonémique (NBrErr).
- le taux de mal classification par classe phonémique (TErr).
- le total des échantillons à chacune des classes phonémiques.

Les graphes de la figure 22, et de la figure 23 illustrent la variation des taux de bonne classification par classe phonémique par rapport à différentes valeurs de α allant de 0 à 0,15 avec un pas de 0,01.

On note d'après ces graphes que la classe des phonèmes de type voyelle possède le taux de bonne classification le plus élevé suivie par la classe des phonèmes de type plosive. Par contre, la classe des phonèmes de type affriquée possède le taux le moins élevé. Ceci peut s'expliquer par le fait que les échantillons les plus représentés sont des *voyelles* et des *plosives*.

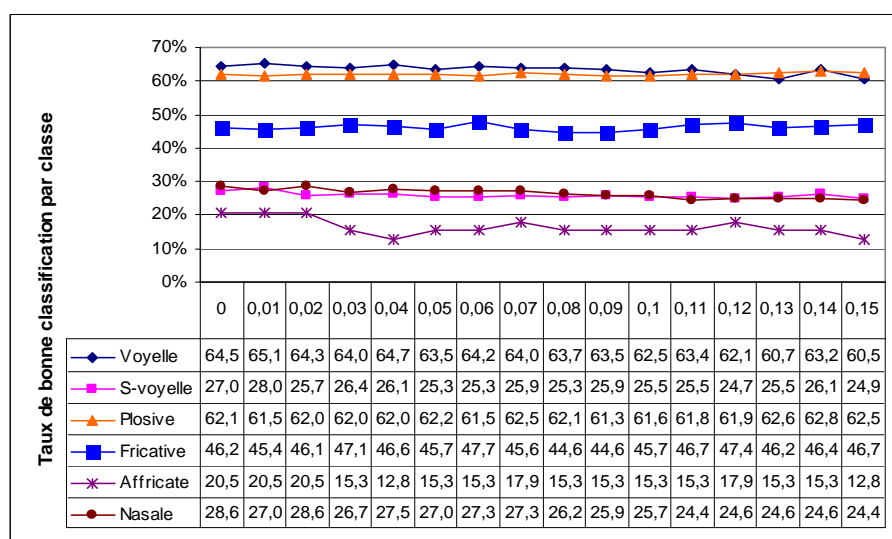


Figure 22. Variation des taux de bonnes classifications par classe phonémique des locuteurs féminins par rapport à α

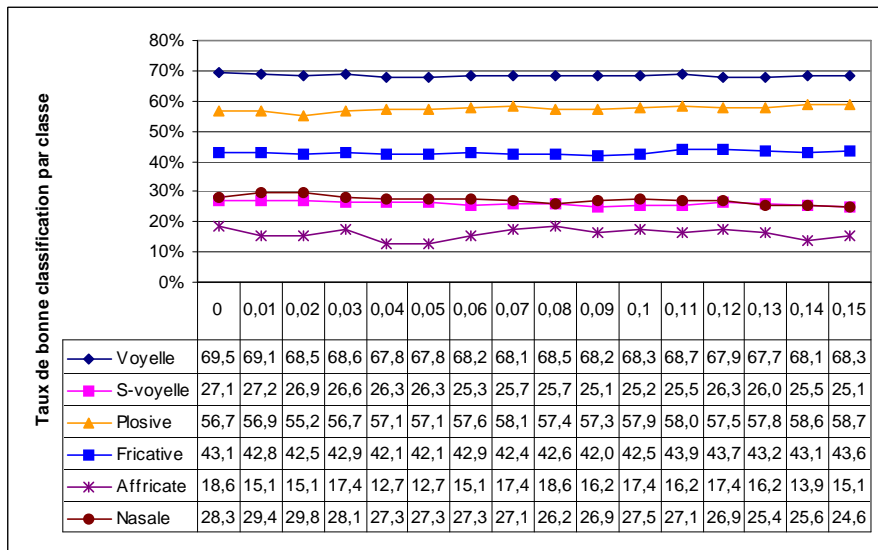


Figure 23. Variation du taux de bonnes classifications par classe phonémique des locuteurs masculins par rapport à α

5.3 Evaluation et synthèse des résultats

Nous notons que :

- les taux de mauvaises classifications des échantillons phonémiques de la base des locuteurs masculins et féminins sont quasiment égaux.
- le taux de bonnes classifications par classe dépend étroitement des pourcentages des classes dans les bases d'apprentissage. En effet, la classe des voyelles, étant la plus représentée, possède le meilleur taux de bonne classification qui est de l'ordre de 70%.
- la classe des plosives possède un taux de bonne classification de l'ordre de 60%.
- la classe des affriquées, la moins représentée, possède le taux de reconnaissance le plus faible.
- l'analyse des cas de mauvaises classifications a permis d'extraire les situations ambiguës. En effet, l'hyperbox est construit à partir des valeurs minimales et maximales des vecteurs (caractéristiques) d'une classe. Si ces valeurs sont proches (c'est-à-dire les vecteurs sont très proches les uns des autres), les échantillons de test seront difficilement affectés à la bonne classe. Quatre cas peuvent expliquer ces difficultés :
 - *1^{er} cas* : Dans la figure 24, l'échantillon b peut être classifié comme étant la classe A parce qu'il se trouve entièrement à l'intérieur de l'hyperbox d'activation de A . Pourtant, b est plus proche du centre de B que de A .

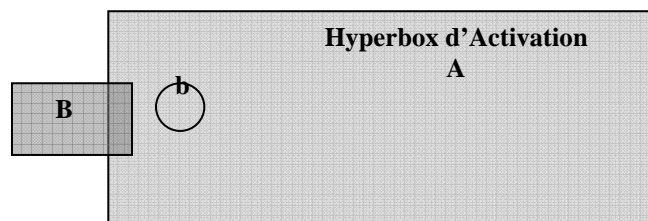


Figure 24. 1^{er} cas de situation ambiguë

- 2^{ème} cas : Dans la figure 25, l'échantillon b est classé comme étant de classe A puisque sa distance à l'hyperbox d'activation de A est inférieur à celle de B . Pourtant, il est clair que b est plus proche du centre de B et par la suite, il doit être attribué à la classe B .

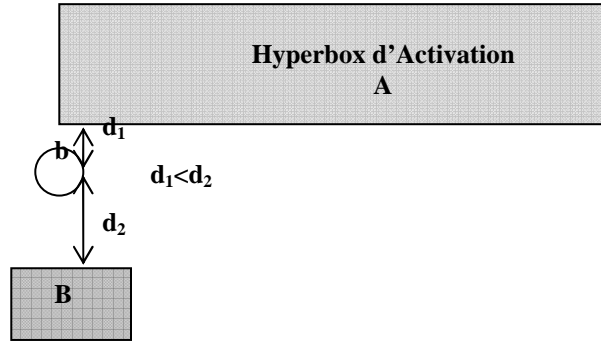


Figure 25. 2^{ème} cas de situation ambiguë

- 3^{ème} cas : La figure 26 montre un autre cas de mal classification. L'échantillon b se trouve à une même distance des hyperboxes d'activation de A et B mais il est toujours plus proche du centre de B que celui de A .

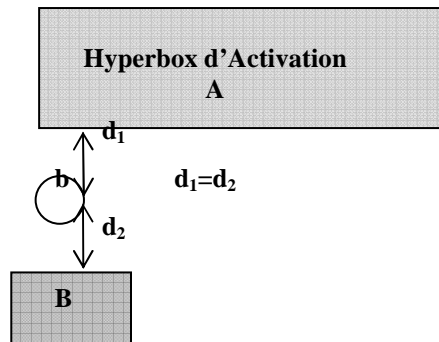


Figure 26. 3^{ème} cas de situation ambiguë

- 4^{ème} cas : Dans ce dernier exemple, l'échantillon b est de la classe A , parce qu'il existe entièrement dans l'hyperbox d'activation de A . Le cas, présenté à la figure 27, est plus complexe que les précédents. En effet, b est plus proche du centre de A que du centre de B .

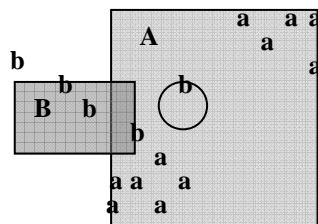


Figure 27. 4^{ème} cas de situation ambiguë

6. Conclusion et perspectives

Nous nous sommes intéressés, dans ce présent travail, à la classification des phonèmes par la génération automatique de règles floues. Cette classification s'effectue suite à la mise en œuvre d'un système d'extraction des règles floues (SERF) constitué de trois processus, à savoir, un processus de préparation des bases d'apprentissage et de test, un processus d'extraction des règles (ERF) et un processus de classification (CERF). Ainsi, nous avons présenté:

- l'étude de la méthodologie d'extraction des règles floues à partir des données numériques. Cette méthode s'appuie sur la délimitation itérative des zones de validation (activation) et des zones de chevauchement (inhibition). A chaque niveau d'itération (chevauchement) on construit une règle floue associée aux familles imbriquées,
- la mise en œuvre et l'implémentation des trois différents processus constituant le système d'extraction de règles floues,
- l'expérimentation du système d'extraction des règles floues appliquée à la base des données phonémiques à coefficients Mel Cepstre (MFCC).

Les résultats obtenus par le système mis en œuvre sont encourageants en terme de bonne classification et en terme du temps de calcul. Néanmoins, certaines questions se sont posées et méritent d'être citées comme perspectives :

- introduction d'un seuil d'échantillons : La méthode élaborée dans ce document, crée des règles floues à partir des individus présents dans une zone de chevauchement. Parfois, le nombre de ces individus est très petit comparé au nombre initial (et surtout lorsque le niveau de chevauchement est élevé), ce qui en résulte la construction de règles floues sans intérêt (void rule). L'introduction d'un seuil d'échantillon peut résoudre considérablement le taux de bonne classification,
- *introduction de la notion de pondération (poids) par propriété* : Les propriétés d'un phonème n'ont pas forcément la même importance. En effet, il peut y avoir des propriétés dominantes et d'autres faibles. Une distinction appliquée sur le poids des propriétés peut améliorer la reconnaissance,
- *utilisation des régions ellipsoïdales et des régions polyédriques* : L'extraction des règles floues peut être réalisée à partir, non seulement par des hyperbox, mais aussi par des régions ellipsoïdales et polyédriques. Chacune de ces techniques possède des limitations au moment de la construction d'une règle floue. Mais en les combinant, on peut réaliser une solution plus puissante et robuste qui permet d'augmenter sensiblement les taux de bonne classification. La méthode, ainsi développée, peut remédier certains cas de mauvaise classification (cas 1, 2 et 3),
- *utilisation de c-moyenne flou (Fuzzy c-means)* : Cette technique réalise une partition de l'espace des attributs en c classes. Chaque vecteur d'attributs ne peut appartenir qu'à une et une seule classe. L'utilisation de cette technique peut résoudre le quatrième cas de mauvaise classification. On utilise cette technique sur chaque classe afin de générer, si nécessaire, des sous-classes appartenant à la classe initiale. On aura une augmentation du nombre de classes, certes, mais on

obtient des classes plus compactes et homogènes. Ensuite, on applique le système d'extraction sur les nouvelles classes.

7. Références

- Abe S., Lan M. S., « A method for fuzzy rules extraction directly from numerical data and its application to pattern classifications », *IEEE Fuzzy System*, vol. 3, pp. 18-28, 1995.
- Abe S., Lan M. S., « Fuzzy rules extraction directly from numerical data for function approximation », *IEEE Trans. Syst. Man. Cybern.*, vol. 25, pp. 119-129, 1995.
- Abe S., Thawonmas R., « Kayama M., A fuzzy classifier with ellipsoidal regions for diagnosis problems », *IEEE Trans. Syst., Man. Cybern. – Part C: Applications and reviews*, vol 29, NO.1, pp. 140-149, February 1999
- Ben Ayed D., Arous N., Ellouze N., « Pramétrisation cepstrale adaptée au corpus de parole de Otago », *Journées Scientifiques 2001, JS, Borj El Amri - Tunis, mai 2001, Tunisie.*
- Bertrand P., Bel Mufti G., « Loevinger's measures of rule quality for assessing cluster stability », *Computational Statistics and Data Analysis*, vol. 50/4, pp 992-1015, 2006
- Billard L., Diday E. *Symbolic Data Analysis: conceptual statistics and data Mining*. 321 pages. Wiley series in computational statistics. Wiley. ISBN 0-470-09016-2, 2006.
- Boite R., Boulard H., Dutoit T., Hancq J., Leich H., « Traitement de la parole », Presses Polytechniques et universitaires romandes, Collection électricité, 2000.
- Chiu S. L., « Fuzzy model identification based on cluster estimation », *Fuzzy syst.*, vol 2, pp. 267-278, 1994.
- Fisher W., Zue V., Bernstein J., Pallet D., « An acoustic-phonetic database », *JASA suppl., A.*, Vol. 81(S92), 1986.
- Haton J.P., Pierrel J.M., Perennou G., Caelen J., Gauvain J.L., *Reconnaissance automatique de la parole*, Dundo informatique, Bordas (Ed.), 1991.
- Jang R., « ANFIS: Adaptive-network-based fuzzy inference system », *IEEE Trans. Syst., Man.Cybern.*, vol 23, pp. 665-685, May-June 1993.
- Klir G. J., Bo Y., *Fuzzy sets and fuzzy logic: theory and application*, Prentice Hall, 1995.
- Prade H., Testemale C., « Fuzzy relational database: Representational issues and reduction using similarity measures », *Journal of the American society for information science*, 38(2) pp.118-126, 1987.
- Reynolds D. A., « Experimental evaluation of features for robust speaker identification », *IEEE transaction Speech Audio Processing*, vol. 2, 1994, pages 639-643.
- Thawonmas R., Abe S., « A novel approach to feature selection based on analysis of class regions », *IEEE Trans. Syst., Man.Cybern. – Part B*, vol 27, pp. 196-207, April 1997.
- Wang L. X, Mendel J. M., « Generating fuzzy rules by learning from examples », *IEEE Trans. Syst., Man. Cybern.*, vol 22, pp. 1414-1427, November-December 1992
- Zadeh L. A., «Fuzzy Sets as a basis foe theory of possibility », *Fuzzy sets and Systems journal*, 1: 3-28, 1978.