UNIVERSITE 20 AOUT 1955-SKIKDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE
M2SI (2020/2021)

COURS : TECHNIQUES DE RECONNAISSANCE DES FORMES

S. HAZMOUNE

1 Présentation générale

La reconnaissance des formes (RdF) est un domaine très riche de l'informatique. RdF est utilisée, essentiellement, pour faciliter l'interaction homme/machine dans ses aspects de perception, de compréhension et de dialogue. L'idée est d'utiliser les techniques de l'apprentissage automatique pour construire des machines capables de reproduire le comportement de l'humain dans ses activités de percevoir et de reconnaitre les objets du monde réel, telles que la reconnaissance de visage et la reconnaissance de la parole. La RdF permet entre autres de :

- Faciliter l'interaction Homme/Machine, telles que la commande vocale.
- Soulager l'homme des tâches difficiles et ennuyeuses, telles que la restauration des documents anciens, le tri des chèques bancaires…
- Remplacer l'homme dans les cas où le traitement et la reconnaissance manuelle est impossible, telles que la recherche d'une anomalie dans une grande base de chiffres, la reconnaissance chromosomes et le comptage globules…

1.1 Vocabulaire

1.1.1 Forme

En RdF, une forme désigne tout objet du monde réel qui peut se présenter sous forme d'une image (caractère, empreinte, visage, etc.) ou d'un signal (son, ECG, radar, etc.).

1.1.2 Classe

Un groupe d'objets ayant des caractéristiques similaires mais différentes par rapport aux objets des autres classes. A chaque classe est associée une étiquette appelée aussi un label.

1.1.3 Exemple (échantillon)

Des objets différents peuvent avoir la même étiquette de classe. On appelle ces objets des exemples ou des échantillons de la classe. Par exemple, en reconnaissance de la parole, les différentes prononciations d'un mot sont des exemples de ce mot.

1.1.4 Vecteur de caractéristiques (Feature vector)

Un vecteur de caractéristiques est un vecteur à d dimensions de caractéristiques numériques qui représentent une forme comme par exemple : L'histogramme d'une image et les coefficients MFCC de la parole. De nombreux algorithmes d'apprentissage automatique nécessitent une représentation numérique des formes, car ces représentations facilitent le traitement et l'analyse statistique. Les caractéristiques représentant les formes doivent être choisies de telle manière à maximiser la similarité intra-classe et la dissimilarité inter-classes. Il est aussi possible d'utiliser des caractéristiques symboliques appelées primitives comme la couleur par exemple. Ce type de caractéristiques est utilisé en cas des approches structurelles.

L'espace de dimension d défini par le vecteur de caractéristiques est appelé espace de caractéristiques ou de représentation.

Exemples : Considérons une population d'individus appartenant à deux classes différentes : Adulte et Enfant. Chaque individu est représenté par deux caractéristiques poids et taille. L'espace des caractéristiques va ressembler au nuage de points montré dans la figure 1-1.

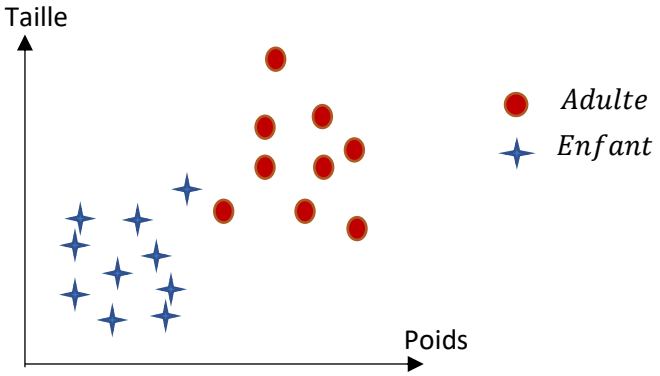


Figure 1.1 Exemple d'espace de caractéristiques

1.1.5 Similarité

Une mesure de similarité est la mesure de la similitude de deux objets de données. La similarité, dans le cas de modèles géométriques, peut être mesurée en calculant la distance entre les vecteurs des caractéristiques des objets. Alors que dans le cas des modèles probabilistes, elle est mesurée en termes de probabilité.

1.1.6 Reconnaissance

C'est l'utilisation des connaissances préalables sur l'environnement pour classifier les objets du monde réel selon certains critères de similarité. En reconnaissance des formes, ces connaissances peuvent être des caractéristiques extraites des données ou des modèles construits à partir de ces données dans une phase préalable appelée apprentissage.

1.2 Définition de la RdF

La reconnaissance des formes est l'ensemble des techniques informatiques et mathématiques qui nous permettent de distinguer les objets du monde réel après une série de traitements automatiques sur les données brutes appelée le processus de reconnaissance. La tâche principale de RdF est la classification, c-à-d., décider si un objet appartient à une classe ou à une autre, ou le clustering, c-à-d, le regroupement des objets en groupes les plus homogènes possible.

L'objectif de la RdF est de doter la machine des capacités de l'humain à analyser, comprendre et reconnaitre les formes. Pour ce faire, la RdF fait appel aux différentes disciplines :

- Psychologie, physiologie et biologie : Pour comprendre de quelle manière l'être humain effectue la reconnaissance.
- Statistiques, mathématiques et informatique (techniques de l'apprentissage automatique) : Pour pouvoir automatiser le processus de reconnaissance en reproduisant le comportement de perception (vision et écoute) et de compréhension (analyse, raisonnement et reconnaissance) de l'humain.

1.3 Champs d'application de la RdF

Il est possible d'imaginer plusieurs applications de la RdF qui sont généralement regroupés selon la nature de l'objet à reconnaitre en deux grandes familles : La reconnaissance des images et la reconnaissance des signaux (voir Tableau 1-1).

Tableau 1-1 Champs d'application de la RdF

	Domaine d'étude	Application
Reconnaissance d'images	Reconnaissance de l'écriture	Bureautique, saisie de textes, tri postal, chèques, matricules, etc.
	Reconnaissance des signatures numériques	Banques, commerce, etc.
	Reconnaissance des empreintes digitales, de visages, etc.	Banques, commerce, police, etc.
	Analyse de radiographies, échographies, reconnaissance chromosomes, comptage des globules, etc.	Médecine : Contrôle systématique de santé
	Détection de défauts : circuits intégrés, pièces métalliques, etc.	Contrôle de qualité industrielle
	Localisation d'objets	Guidage de robots industriels et guidage missiles, etc.
	Analyse d'images de satellites et de photos aériennes	Météorologie, agriculture, surveillance militaire

Reconnaissance des signaux	Reconnaissance de la parole	Bureautique, commande vocale, ordinateur sans clavier ni souris, traduction automatique en temps réel des langues étrangères, sous-titrage, etc.
	Reconnaissance du locuteur	Banques, commerce, police, etc.
	Reconnaissance d'émotions à partir de la voix	Commerce, psychologie, etc.
	Reconnaissance des signaux biomédicaux (ECG, PCG, etc.)	Médecine
	Analyse des signaux radar	Identification des cibles aériennes, reconnaissance Ami/Ennemi

1.4 Processus général de la RdF

Les informations issues du monde réel via le capteur et fournies au système de reconnaissance sont généralement trop volumineuses et peu pertinentes. Le processus de RdF sert à la réduction progressive de ces informations partant de l'espace d'observations (formes) et passant par celui de représentation et arrivant à celui d'interprétation. Ce processus comporte un ensemble de tâches permettant le passage d'un espace à un autre et chacune d'elles a des implications sur les autres. La figure 1-1 illustre le processus général de la RdF.

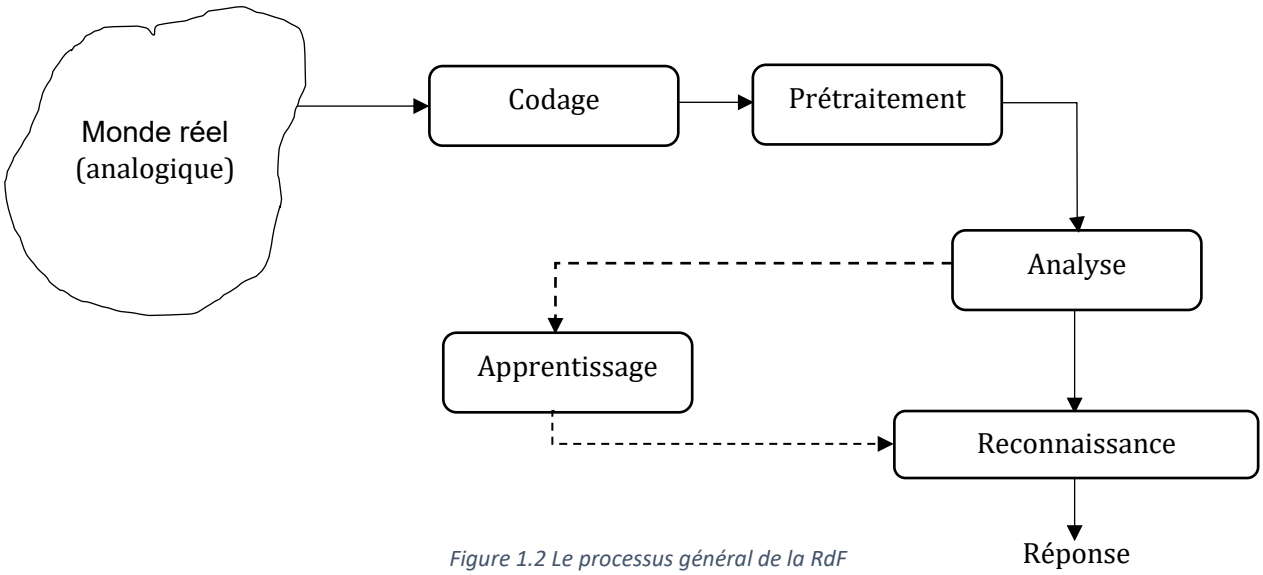


Figure 1.2 Le processus général de la RdF

Dans ce qui suit, nous allons présenter en détail chaque étape du processus.

1.4.1 Le codage

Le codage consiste à transformer un ensemble de données analogiques en un ensemble de données numériques de telle sorte à pouvoir les traiter par l'ordinateur. Cette transformation doit se faire de la façon la plus fidèle possible, c-à-d., sans perte d'information et en maintenant les propriétés essentielles de la forme analogique. Le codage, comme le montre la figure suivante, se fait en deux étapes : La détection et la numérisation.

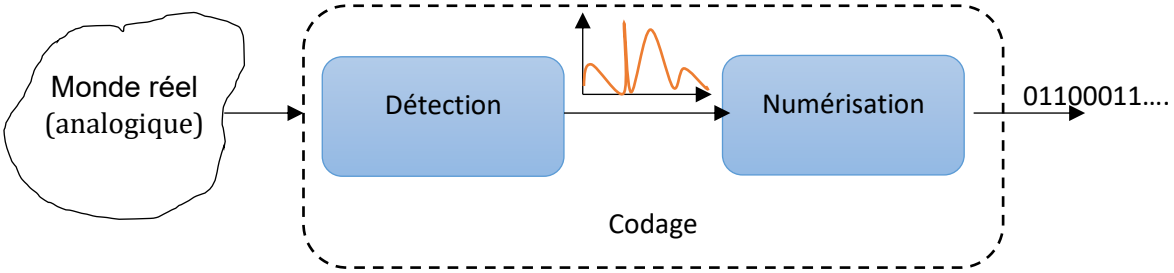


Figure 1.3 étape de codage

1.4.1.1 La détection

Un capteur qui se substitue à l'œil ou à l'oreille d'un observateur convertit des grandeurs physiques mesurées sur un objet en signaux électriques. Par exemple, en reconnaissance de la parole, un microphone convertit une vibration de pression de l'air en tension électrique.

1.4.1.2 La numérisation

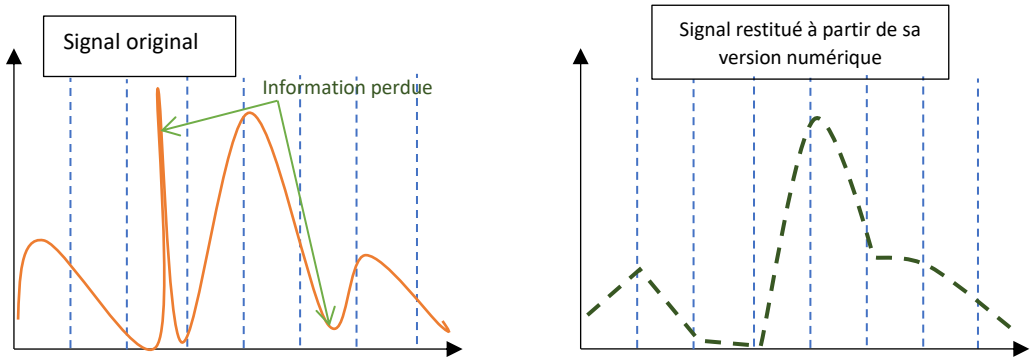
Cette opération est effectuée par un convertisseur analogique-numérique permettant le passage d'une représentation continue à une représentation discrète selon les deux étapes suivantes :

a. L'échantillonnage : Son principe consiste à découper temporellement et de façon régulière le signal analogique à des tranches appelées échantillons. L'intervalle de temps entre deux prises d'échantillons T_e est appelé période d'échantillonnage ($T_e = T_n - T_{n-1}$), et son inverse s'appelle fréquence d'échantillonnage F_e qui représente le nombre d'échantillons prélevés par seconde, son unité de mesure est le HZ (1/s).

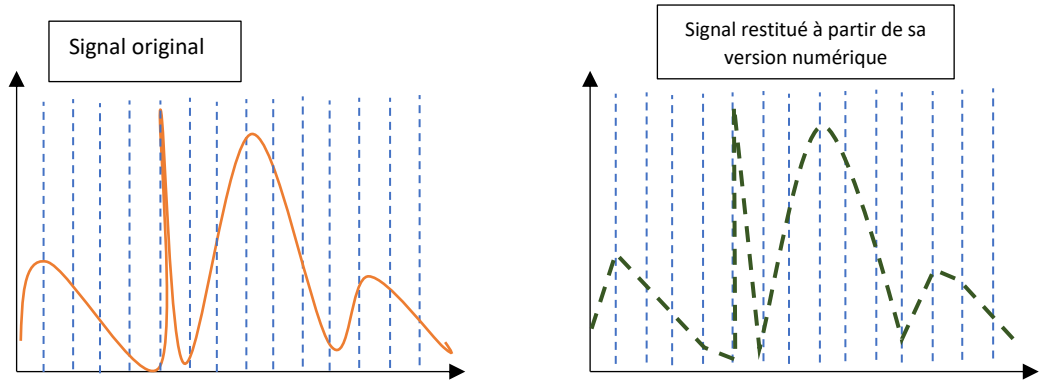
F_e doit être suffisamment élevée si l'on ne veut pas perdre trop d'informations sur le signal. Cependant plus F_e est élevée, plus le temps disponible pour effectuer les traitements numériques sera court et plus le nombre d'échantillons à traiter sera important. La question qui se pose est donc : Comment choisir F_e?

Règle de Shanon : La F_e doit être au moins deux fois la fréquence maximale du signal F. Pour un signal de parole avec $F = 4\ KHZ$ par exemple, ça nécessite une F_e d'au moins 8 KHZ. Lorsqu' on utilise un F_e trop faible, on se trouve face au phénomène repliement du spectre d'où la déformation du signal restitué (reconstruit à partir de sa version numérique) puisque les variations rapides du signal analogique ne sont pas enregistrées.

a. La quantification : Elle consiste à donner des valeurs numériques aux amplitudes des échantillons résultants de l'étape précédente. C'est une discrétisation du signal analogique en valeurs. Pour coder ces valeurs, on utilise un nombre fixe de bits généralement 4, 8 ou 16. Si on suppose que n est le nombre de bits utilisé pour le codage, alors le nombre de niveaux différents de mesures sera 2^n. Notons que plus le nombre de bits est grand, plus la qualité de numérisation sera meilleure et plus l'espace mémoire occupé sera important. Il faut donc chercher un bon compromis selon les besoins de l'application.



Fréquence d'échantillonnage trop faible ($F_e < 2F_{max}$)



Fréquence d'échantillonnage convenable ($F_e \geq 2F_{max}$)

Figure 1.4 L'effet du choix de la fréquence d'échantillonnage sur la qualité de la numérisation

Remarque : En cas d'images, l'échantillonnage est un découpage spatial d'un signal continu (intensité lumineuse) à deux variables, ce qui donne une grille d'éléments appelés pixels. Alors que la quantification consiste à donner des valeurs entières aux pixels et de les coder ensuite sous forme binaire.

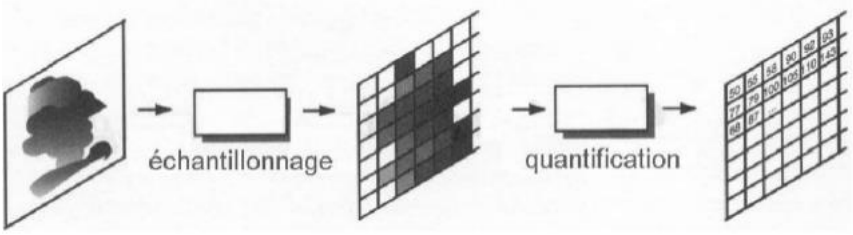


Figure 1.5 Numérisation d'images

1.4.2 Le prétraitement

Le but de prétraitement est de préparer les données issues du capteur pour l'étape suivante d'analyse. Cette dernière n'est pas possible et surtout fiable que si les données reçues du

capteur sont dénuées de bruits, corrigées de leurs erreurs éventuelles, normalisées et réduites à l'essentiel.

1.4.2.1 Suppression de bruit

L'objectif de cette opération de prétraitement est de débarrasser les données du bruit de l'acquisition et de ne garder que l'information significative de la forme. Le bruit provient de plusieurs sources telles que le capteur (qualité, réglage, …) et les conditions de prise de mesures (mauvais éclairage, positionnement, …).

1.4.2.2 Correction des erreurs

Des erreurs peuvent être commises lors de prise de mesures. Elles sont de différents types :

- Erreurs dus au matériel de saisie : Les parasites qui troublent la réception des signaux suite à un mauvais réglage du capteur comme par exemple, image floue.
- Erreurs dus à l'environnement : Le bruit ambiant, mauvais éclairage (la forme saisie peut être incomplète où l'information peut manquer à des endroits spécifiques), le positionnement des objets à saisir, …
- Erreurs dus à l'objet lui-même : Qui peut être considéré comme un cas pathologique parce qu'il ne présente pas les mêmes caractéristiques que l'ensemble des autres objets de la même classe.

1.4.2.3 Normalisation des données

L'objectif de la normalisation est de pouvoir ranger des objets identiques mais de taille différente dans une même classe.

Exemple. En reconnaissance de la parole, la même élocution peut être prononcée lentement ou rapidement. On normalise le signal en durée afin de s'affranchir les variations de rythme et en amplitude afin de s'affranchir les variations de volume. En reconnaissance de l'écriture manuscrite, on peut normaliser les dimensions des images de telle sorte que les deux exemples c et C soient rangées dans une même classe (la lettre C).

1.4.2.4 Réduction de données

On peut réduire les données soit en éliminant les informations redondantes et superflues ou en rassemblant dans un même code un ensemble de données qui vérifient toutes une même propriété.

1.4.3 L'analyse

Cette étape est aussi appelée paramétrisation ou extraction des caractéristiques. Elle consiste à tirer les informations jugées pertinentes de l'objet physique et ne garder que celles discriminantes. L'objectif est de réduire l'espace de représentation et de faciliter leur

interprétation (classification). Ces informations sont exprimées sous une forme numérique et dites caractéristiques (features) (en cas des approches de classification statistiques) ou sous une forme symbolique et dites primitives (en cas des approches de classification structurelles). Dans ce cours, on va s'intéresser aux approches de classification statistiques.

Les caractéristiques/ primitives représentant les objets doivent être :

- Pertinentes : Suffisantes
- Discriminantes : Elles doivent permettre la séparation les objets différents facilement, c-à-d., assurer une similarité intra-classe maximale et inter-classes minimale.
- Robustes : Elles ne doivent pas trop sensibles à l'environnement (bruit, perturbations, …).

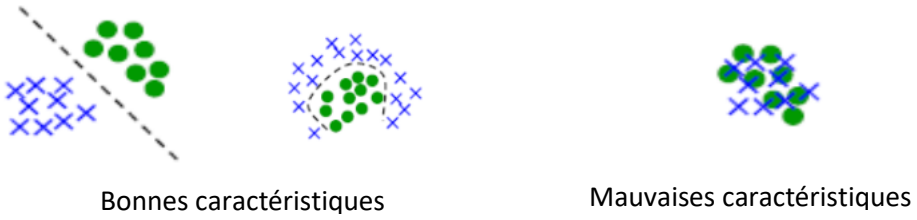


Figure 1.6 Bonnes & mauvaises caractéristiques

Exemple : On veut classifier un ensemble de formes géométriques en trois classes : Triangle, carré et rectangle. Proposer un ensemble de caractéristiques permettant de distinguer les objets des 3 classes.

Prenons dans un premier temps, comme caractéristiques, le nombre d'angles $NombreAngles$, la somme d'angles $SommeAngles$ et la variable logique $AnglesEgaux$ qui vaut 1 si tous les angles de la forme sont égaux et 0 autrement. Les caractéristiques d'une forme X sont regroupées dans un vecteur de la forme :

$$X=\begin{bmatrix} NombreAngles \\ SommeAngles \\ AnglesEgaux \end{bmatrix}$$

$$\text{Triangle} = \begin{bmatrix} 3 \\ 180° \\ 0\ ou\ 1 \end{bmatrix}, \text{Carré} = \begin{bmatrix} 4 \\ 360° \\ 1 \end{bmatrix}, \text{Rectangle} = \begin{bmatrix} 4 \\ 360° \\ 1 \end{bmatrix}$$

On remarque que les 3 caractéristiques sont pertinentes mais non discriminantes, car elles ne permettent pas de distinguer la classe Carré de la classe Rectangle.

Pour remédier à cette situation, on peut ajouter une 4ième caractéristique discriminante soit par exemple la variable logique CôtésEgaux valant 1 si tous les côtés de la forme sont égaux et 0 si non. Le vecteur des caractéristiques d'une forme X aura donc la forme :

$$X=\begin{bmatrix} NombreAngles \\ SommeAngles \\ AnglesEgaux \\ CôtésEgaux \end{bmatrix}$$

Ainsi les 3 classes sont représentées comme suit :

$$\text{Triangle} = \begin{bmatrix} 3 \\ 180° \\ 0\ ou\ 1 \\ 0\ ou\ 1 \end{bmatrix}, \text{Carré} = \begin{bmatrix} 4 \\ 360° \\ 1 \\ 1 \end{bmatrix}, \text{Rectangle} = \begin{bmatrix} 4 \\ 360° \\ 1 \\ 0 \end{bmatrix}$$

On remarque cette fois-ci que les caractéristiques sont pertinentes mais aussi discriminantes. Cependant, on peut encore optimiser les vecteurs de caractéristiques en éliminant la 3ième caractéristique ($AnglesEgaux$) pour réduire l'espace de représentation. On remarque également que la 1ière et la 2ième caractéristiques sont corrélées, on peut donc éliminer l'une des deux, soit la 2ième par exemple. Ainsi, on obtient un ensemble réduit de caractéristiques pertinentes, discriminantes et robustes :

$$X=\begin{bmatrix} NombreAngles \\ CôtésEgaux \end{bmatrix}$$

Ainsi les objets des 3 classes sont représentés comme suit :

$$\text{Triangle} = \begin{bmatrix} 3 \\ 0\ ou\ 1 \end{bmatrix}, \text{Carré} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \text{Rectangle} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

Pour résumer, il faut toujours chercher un compromis entre la qualité des caractéristiques et leur nombre.

1.4.4 L'apprentissage

L'apprentissage consiste à fournir au système un ensemble de formes appelé base d'apprentissage (training dataset) et utiliser des algorithmes de l'apprentissage automatique pour générer des modèles de référence à partir de ces données. Ces modèles sont utilisés par la suite pour reconnaitre de nouvelles formes inconnues. En cas de classification, une forme est exprimée sous forme de couple (X, Y), tel que X est le vecteur de caractéristiques extrait de la forme lors de l'étape d'analyse et Y son étiquette (label) de classe.

L'apprentissage est la phase la plus importante car la performance du système dépend des modèles établis.

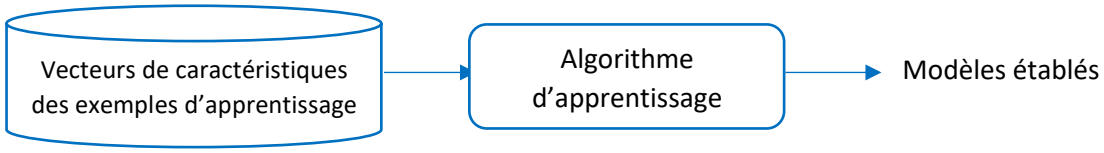


Figure 1.7 Phase d'apprentissage

En général, ces modèles permettent de structurer l'espace des caractéristiques en régions(classes) avec des frontières plus ou moins complexes et de faire en sorte que ce partitionnement permette de prendre les meilleures décisions (affectation d'un vecteur de caractéristiques à la bonne classe).

Exemple : Soient deux classes W_1 et W_2. Pour chaque classe, on dispose de 5 exemples définis par des vecteurs dans $\mathbb{R}^2$: $\begin{pmatrix}\alpha\\ \beta\end{pmatrix}$.

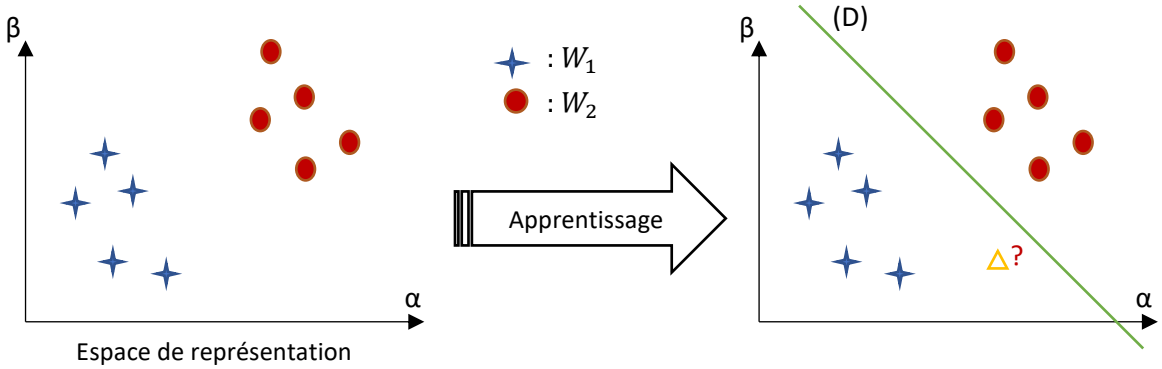


Figure 1.8 Apprentissage (frontière de décision simple)

L'apprentissage, dans ce cas (2 classes), consiste à trouver l'équation de la courbe (D) : la frontière de décision. La reconnaissance d'un nouvel exemple consiste à déterminer la position de l'exemple par rapport à la courbe (D). On peut distinguer deux cas :

- (D) est une droite : On parle de problème linéairement séparable.
- (D) n'est pas une droite : On parle de problème non linéairement séparable, d'où la difficulté de trouver l'équation de la courbe optimale.

Pour réaliser l'apprentissage, on fait appel à des techniques plus ou moins sophistiquées de l'apprentissage automatique : Les SVM, les RN, les HMM, les arbres de décision, le classifieur bayésien…

Remarque : La construction d'un modèle nécessite la recherche de ses paramètres optimaux qui peuvent être des :

- Règles dans les règles de décision (systèmes à base de connaissances)

- Conditions et branchements dans les arbres de décision
- Coefficients dans les classificateurs linéaires
- Distributions de probabilité dans les classificateurs probabilistes (HMM, Bayes, …)
- Poids dans les réseaux de neurones
- Paramètres des hyperplans séparateurs pour les SVM

Il existe plusieurs types d'apprentissage à partir de données : Supervisé (classification ou régression), non supervisé (clustering) et semi-supervisé. Pour plus de détail, référer au cours Techniques d'apprentissage en IA (M1 SI).

1.4.5 La reconnaissance (décision)

La décision est la phase ultime de tout système de RdF. En se basant sur les modèles (classifieur) établis préalablement lors de l'apprentissage, elle permet d'affecter une classe à un nouvel exemple représenté par son vecteur de caractéristiques. La stratégie de décision dépond de la technique de l'apprentissage utilisée.

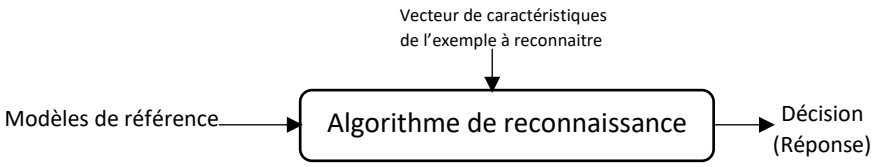


Figure 1.9 étape de décision

Le résultat de cette étape peut être l'une des décisions suivantes :

- Réponse correcte : le système classifie correctement l'exemple à reconnaitre.
- Fausse classification (substitution) : le système affecte à l'exemple à reconnaitre une classe à laquelle il n'appartient pas.
- Ambiguïté (confusion) : le système trouve plus d'une solution. Pour enlever l'ambiguïté, une étape de post-traitement est nécessaire. Celle-ci consiste, par exemple, à calculer un poids pour chaque classe.
- Rejet : le système ne peut pas décider, car aucune classe n'est suffisamment proche à l'exemple à reconnaitre. C'est le cas d'un mot qui n'appartient pas au vocabulaire considéré. Une telle décision est possible grâce à un ou plusieurs seuils de rejet, généralement fixés de façon empirique dans la phase d'apprentissage.

L'erreur de substitution est l'erreur la plus grave comparée à l'erreur de rejet. En effet, pour un système critique par exemple, il est préférable de ne pas décider que de faire une fausse décision, qui pourrait causer de graves conséquences.

1.4.6 Evaluation des performances d'un système de RdF

La dernière phase dans le processus de développement d'un SRAP est l'évaluation de sa qualité. Pour cette fin, une base de test contenant plusieurs exemples de chaque classe est utilisée. La base de test doit être différente de celle d'apprentissage. Typiquement, environ 80% des données disponibles pour le développent d'un système de RdF sont réservées à l'apprentissage, et 20% au test. La base de données utilisée étant limitée, et elle doit être partitionnée judicieusement dans une partie d'apprentissage et une autre de test, car la qualité du système dépond beaucoup de cette partition. En effet, si la base d'apprentissage est de faible taille, le classifieur résultant ne sera pas très robuste et aura une faible capacité de généralisation (reconnaitre de nouveaux exemples inconnus). D'autre part, si la base de test est de petite taille, la confiance dans le résultat d'évaluation sera faible.

Plusieurs méthodes de partitionnement existent. Ces méthodes diffèrent par la façon dont elles utilisent les exemples disponibles comme bases d'apprentissage et de test. Si le nombre d'exemples disponibles est extrêmement important, toutes ces méthodes sont susceptibles de conduire au même résultat d'évaluation. Dans ce qui suit, nous donnons un résumé de ces méthodes :

- La méthode de resubstitution : Tous les exemples disponibles sont utilisés à la fois pour l'apprentissage et pour le test ; la base d'apprentissage et celle de test sont les mêmes. Cette méthode assure un bon apprentissage mais une confiance faible dans le résultat d'évaluation. En effet, pour une bonne évaluation, il faut éviter d'utiliser la base d'apprentissage pour le test, et ce, pour ne pas produire une vue optimiste de l'évaluation.
- La méthode « Holdout » : La moitié des données est utilisée pour l'apprentissage et le reste pour le test. Les parties de test et d'apprentissage sont indépendantes. L'inconvénient de cette méthode est que différents partitionnements donnent différents résultats d'évaluation.
- La méthode « Leave one out » : Un classifieur est conçu en utilisant $(n-1)$ exemples et évalué en utilisant l'exemple qui reste. La méthode permet d'utiliser un maximum de données pour l'apprentissage et elle est très utilisée lorsque les bases de données sont de tailles insuffisantes. Son inconvénient est qu'elle exige un calcul important, impliquant plusieurs classifieurs différents qui doivent être entrainés et testés.
- La méthode de validation croisée (n-fold cross validation) : Il s'agit d'un compromis entre « Holdout » et « Leave one out ». Elle divise les données en P , où $(1 \leq P \leq n)$ sous bases différentes, $(P-1)$ sous bases sont utilisées pour l'apprentissage et le reste pour le test.

- La méthode de rééchantillonnage « Bootstrap » : Cette méthode rééchantillonne les données disponibles avec remplacement pour générer un certain nombre d'ensembles de données « simulées » (généralement des centaines) de la même taille que l'ensemble d'apprentissage initial. Elle est utilisable si la quantité de données disponible est faible.

Il devient désormais courant d'utiliser trois au lieu de deux bases de données : une pour l'apprentissage, une pour la validation et une pour le test. La base de test est invisible pendant le processus d'apprentissage. La base de validation peut être considérée comme un pseudo-test. Nous continuons le processus d'apprentissage jusqu'à ce que l'amélioration des performances sur la base d'apprentissage ne soit plus accompagnée d'une amélioration des performances sur la base de validation. À ce stade, l'apprentissage doit être arrêté afin d'éviter le sur-apprentissage (voir cours Techniques d'apprentissage en IA, M1 SI).

Selon les besoins de l'application, trois aspects principaux peuvent être considérés durant l'évaluation d'un système de RdF : Ses performances, sa robustesse et sa complexité. Par la suite, nous allons aborder en détails chacun de ces aspects.

1.4.6.1 Les performances

L'aspect performance est le plus considéré pour évaluer la qualité d'un système de RdF. Le choix d'une métrique dépend essentiellement de l'application. Les métriques les plus utilisées sont :

- La matrice de confusion : C'est une matrice de taille NxN, où N est le nombre de classes. Un élément a_{ij} d'une matrice de confusion représente le nombre d'exemples de la classe i qui ont été affectés à la classe j. La diagonale indique, donc, le nombre d'exemples correctement reconnus. Elle est appelée « matrice de confusion », car elle permet de repérer facilement où le système confond deux classes.

- Taux de reconnaissance ou exactitude (recognition rate ou accuracy) : Il est basé sur le nombre de prédictions erronées. C'est le pourcentage des exemples correctement reconnus.

$$Taux\_Rec = Rec\_Rate = \frac{nombre\ d'exemples\ correctement\ reconnus}{nombre\ total\ d'exemplestestés}$$

- Taux d'erreur (Error rate) : Il est basé sur le nombre de prédictions correctes. C'est le pourcentage des exemples non correctement reconnus.

$$Taux\_Erreur = Error\_Rate = \frac{nombre\ d'exemples\ non\ correctement\ reconnus}{nombre\ total\ d'exemplestestés}$$

Exemple : Le tableau 1-3 montre un exemple de matrice de confusion pour un problème de classification bi-classes (2 classes).

Tableau 1-2 Matrice de confusion d'un problème bi-classes

		Classe reconnue	
		Classe 1	Classe 2
Classe actuelle	Classe 1	a_{11}	a_{12}
	Classe 2	a_{21}	a_{22}

Le nombre d'exemples correctement reconnus = $a_{11} + a_{22}$

Le nombre d'exemples correctement reconnus = $a_{12} + a_{21}$

Le nombre total d'exemples testés (taille de la base de test) = $a_{11} + a_{22} + a_{12} + a_{21}$

Donc :

$$Taux\_Rec = Rec\_Rate = \frac{a_{11} + a_{22}}{a_{11} + a_{22} + a_{12} + a_{21}}$$

$$Taux\_Erreur = Error\_Rate = \frac{a_{12} + a_{21}}{a_{11} + a_{22} + a_{12} + a_{21}}$$

Il existe d'autres métriques telles que, la précision, la sensibilité, la spécificité, …

1.4.6.2 La robustesse des SRAP

La notion de robustesse a été introduite par Box (Box, 1979). Elle est définie comme étant la capacité du système de demeurer stable face à la variabilité des données et aux perturbations de l'environnement.

1.4.6.3 La complexité

Un troisième aspect à évaluer dans un système de RdF est sa complexité. Celle-ci peut être mesurée en termes de ressources mémoire ou de temps de calcul de l'apprentissage et de la reconnaissance. Les algorithmes de reconnaissance, gourmands en temps de calcul et de mémoire, fonctionnent aujourd'hui sur des composants numériques de base et ne requièrent plus de carte ou de matériel spécialisé, et ce, grâce au développement rapide de la microélectronique conduisant à une augmentation en puissance des processeurs standards et des mémoires qui, selon la loi de Moore (Moore, 1965), double environ tous les deux ans.

1.4.7 Schéma Conception/Utilisation d'un système de RdF

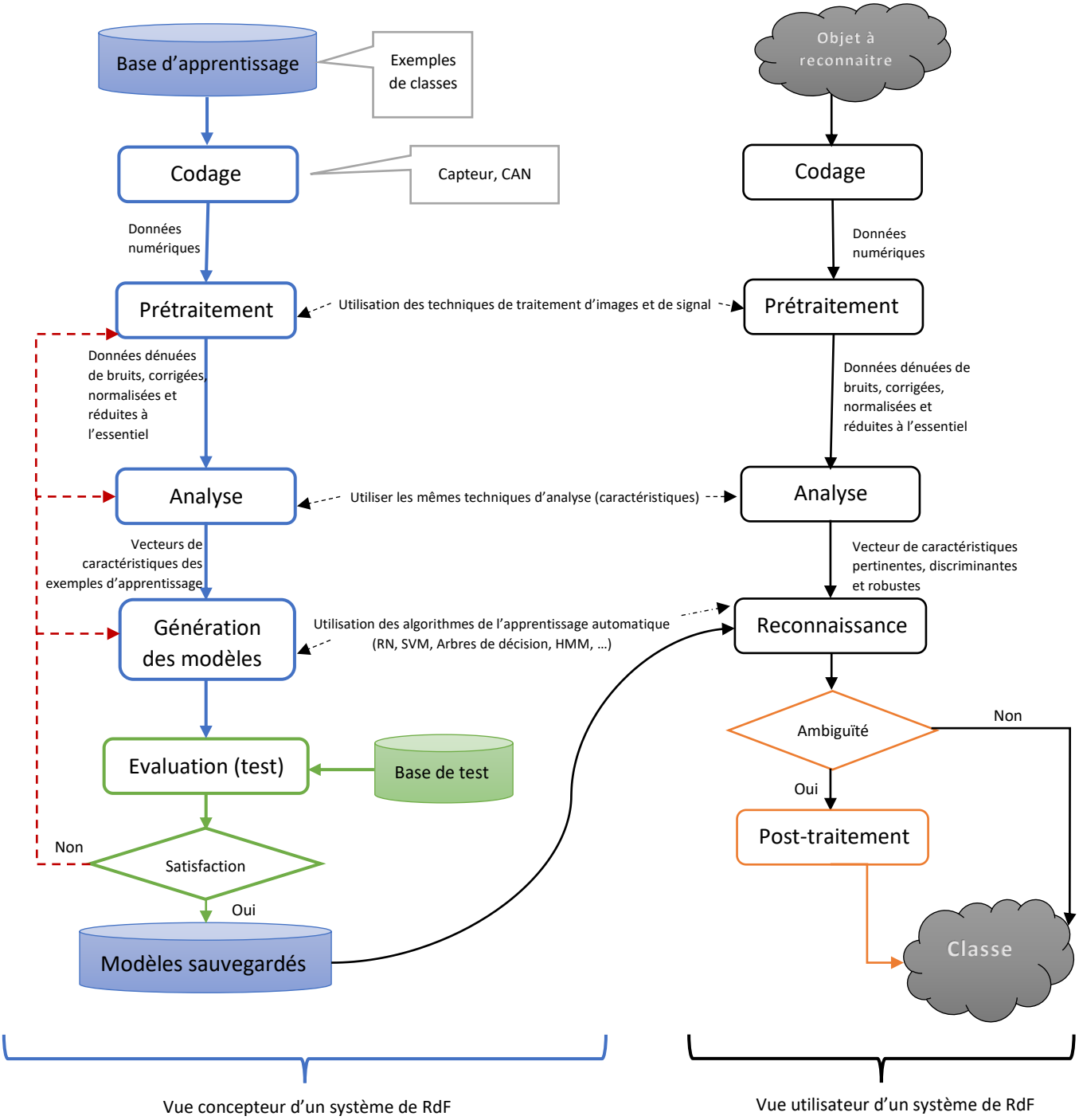


Figure 1.10 Un Schéma détaillé résumant le processus de conception et celui d'utilisation d'un système de RdF

1.4.8 Un exemple concret

http://www.byclb.com/TR/Tutorials/neural_networks/ch1_1.htm

Considérons l'exemple suivant. Dans une usine de conditionnement du poisson, on veut automatiser le processus de tri des poissons entrants sur un tapis roulant en fonction des espèces. Pour ce faire on essaye de séparer le bar/loup du saumon en utilisant la détection optique.

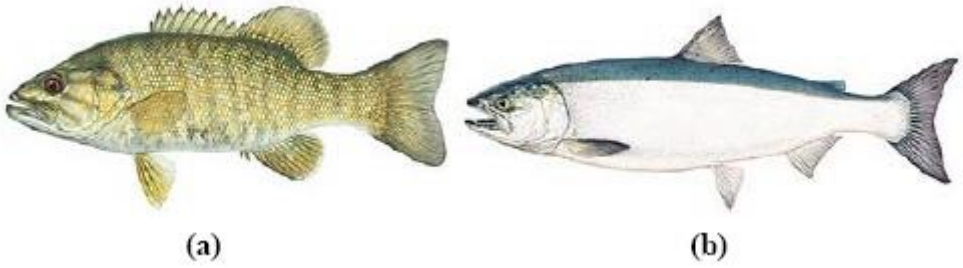


Figure 1.11 Formes à classifier : (a) Bar, (b) Saumon

Nous installons une caméra (voir Figure 1.11), prenons quelques exemples d'images et commençons à noter quelques différences physiques entre les deux types de poissons, longueur, légèreté, largeur, nombre et forme des nageoires, position de la bouche, etc. , et ceux-ci suggèrent des caractéristiques à explorer pour une utilisation dans notre classifieur. On remarque également des bruits ou des variations dans les images, des variations d'éclairage, et de la position du poisson sur le convoyeur, voire statique du fait de l'électronique de la caméra elle-même.

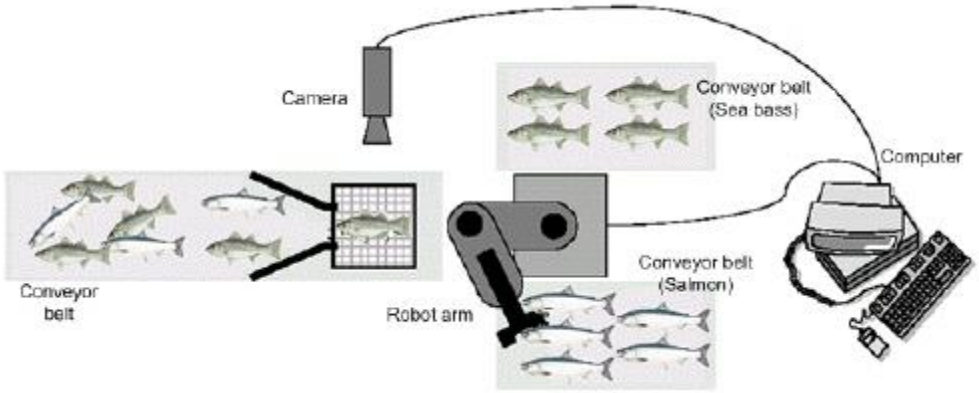


Figure 1.12 Le système de conditionnement du poisson

Comme il existe vraiment des différences entre la population de bar et celle de saumon, nous les considérons comme ayant des modèles différents, des descriptions différentes, qui sont généralement de forme mathématique. Le but de l'approche de classification des formes est

de mettre des hypothèses sur les classes de ces modèles, de traiter les données détectées pour éliminer le bruit et, pour tout forme détectée, de choisir le modèle qui correspond le mieux.

Dans le système prototype (Figure 1.11), d'abord, la caméra capture une image du poisson (étape de codage). Ensuite, les signaux de la caméra sont prétraités pour simplifier les opérations ultérieures sans perdre les informations pertinentes. En particulier, on peut utiliser une opération de segmentation dans laquelle les images de différents poissons sont en quelque sorte isolées les unes des autres et de l'arrière-plan (on peut ajuster automatiquement le niveau de lumière moyen ou limiter l'image pour supprimer l'arrière-plan de la bande transporteuse). Les informations d'un seul poisson sont ensuite envoyées à un extracteur de caractéristiques, dont le but est de réduire les données en mesurant certaines caractéristiques ou propriétés (étape d'analyse). Ces caractéristiques sont ensuite transmises à un classifieur qui évalue les informations présentées et prend une décision finale quant à l'espèce (étape de décision).

Avant de classifier les poissons en bar ou saumon, on doit disposer d'un ensemble d'apprentissage contenant plusieurs exemples pour chaque classe. A partir de cet ensemble, on peut établir des modèles (hypothèses) qui seront par la suite utilisés pour classifier des nouveaux exemples de poissons. En examinant l'ensemble d'apprentissage, plusieurs hypothèses peuvent être faites :

a. Les bars sont généralement plus longs que les saumons : Alors la longueur devient une caractéristique évidente, et on peut essayer, dans ce cas, de classer le poisson simplement en voyant si la longueur l d'un poisson dépasse une certaine valeur critique l^* ou non. Cette valeur (seuil) l^* peut être facilement calculée en examinant les exemples de l'ensemble d'apprentissage.

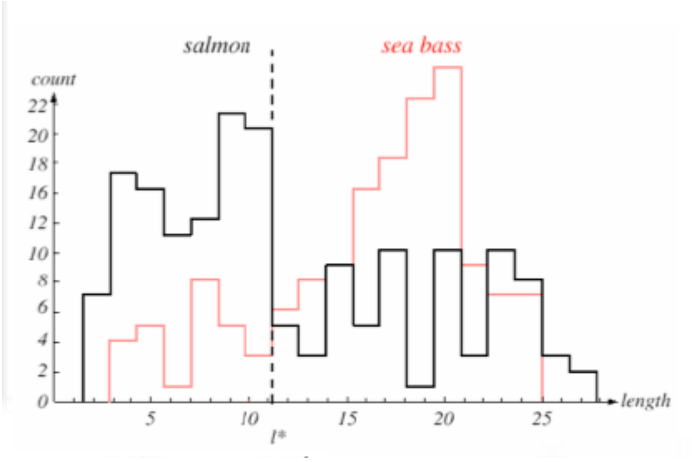


Figure 1.13 Histogrammes de la longueur pour les deux classes. La valeur l^* entraînera le plus petit nombre d'erreurs

Les histogrammes de la figure 1.12 confirment l'affirmation selon laquelle le bar est un peu plus long que le saumon, en moyenne, mais il est clair que ce critère unique est assez pauvre ; peu importe la façon dont on choisit l^*, nous ne pouvons pas séparer de manière fiable le bar du saumon par la longueur seulement.

b. Les bars sont généralement plus clairs que les saumons : Ainsi, on peut utiliser, comme caractéristique, la luminosité.

Les histogrammes résultants et la valeur critique x^* illustrés dans la figure 1.13 sont beaucoup plus satisfaisants : les classes sont bien mieux séparées (on a moins d'erreurs). Cependant, si les consommateurs acceptent sans trop de problèmes de retrouver dans une boîte de bar un peu de saumon, l'inverse n'est pas vrai !

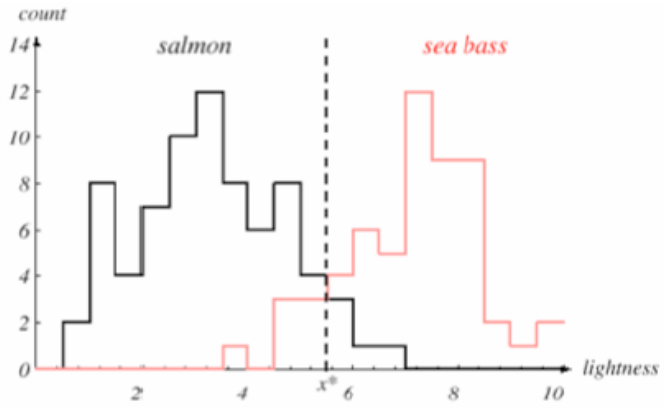


Figure 1.14 Histogrammes de la luminosité pour les deux classes. La valeur l^* entraînera le plus petit nombre d'erreurs

Pour réduire encore le nombre d'erreurs, on peut utiliser plus d'une caractéristique à la fois : la luminosité et la largeur (le bar est généralement plus large d'un saumon). Dans ce cas, l'espace des caractéristiques est $\mathbb{R}^2$. Et les exemples de l'ensemble d'apprentissage peuvent être représentés par le nuage de points illustré dans la figure 1.14.

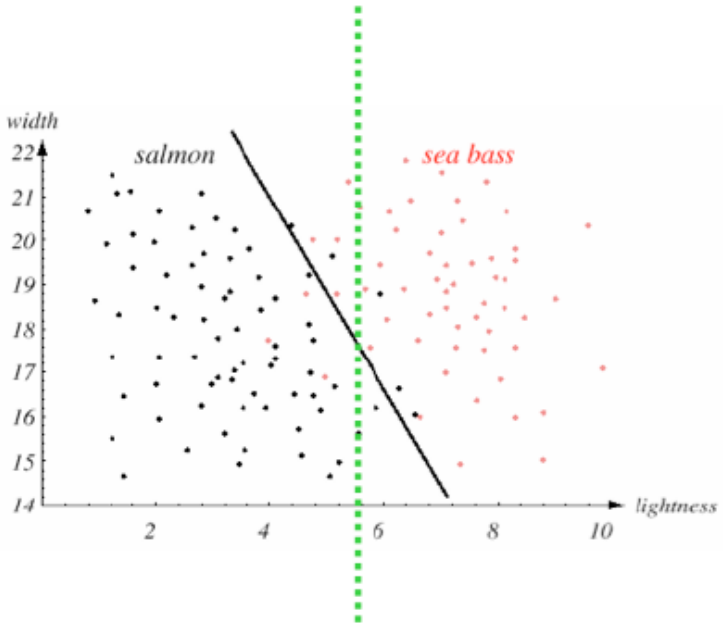


Figure 1.15 L'espace des caractéristiques (luminosité et largeur) de l'ensemble d'apprentissage pour les deux classes Bar et Saumon avec une droite caractérisant la frontière de décision (modèle simple : sous-apprentissage)

Le problème d'apprentissage est maintenant de diviser l'espace des caractéristiques en deux régions, tel que tous les points d'une région, sont des bars, et tous les points de l'autre, sont des saumons. Le graphique de la figure 1.14 suggère la règle suivante pour séparer les poissons : Classifier le poisson comme bar si son vecteur des caractéristiques tombe au-dessus de la frontière de décision indiquée (la droite), et comme saumon dans le cas contraire.

Cette règle semble être bonne pour la classification des exemples et suggère qu'il serait peut-être souhaitable d'incorporer encore plus de caractéristiques. Outre la luminosité et la largeur du poisson, on peut introduire certains paramètres de forme, tel que le placement des yeux. Comment savoir à l'avance laquelle de ces caractéristiques fonctionnera le mieux ? Certaines caractéristiques peuvent être redondantes. Par exemple, si la couleur des yeux de tous les poissons correspondait parfaitement à la largeur, les performances de classification ne vont pas être améliorées en introduisant la couleur des yeux comme caractéristique.

On suppose que les autres caractéristiques sont trop coûteuses à mesurer ou qu'elles apportent peu à la classification, et qu'on est obligé de prendre la décision en fonction des

deux caractéristiques seulement. Si nos modèles étaient extrêmement complexes, notre classifieur aurait une frontière de décision plus complexe que la simple ligne droite. Dans ce cas, tous les exemples d'apprentissage seraient parfaitement séparés, comme le montre la figure 1.15. Avec une telle solution, cependant, notre satisfaction serait prématurée car l'objectif principal de la conception d'un classifieur est de suggérer des actions lorsqu'il est présenté avec de nouveaux exemples, c'est-à-dire des poissons non encore vus. C'est le problème de la généralisation. Il est peu probable que la frontière de décision complexe de la figure 1.15 offre une bonne généralisation ; elle est adaptée aux exemples de l'apprentissage plutôt qu'au véritable modèle de tous les bars et saumons qui devront être séparés (erreur réduite sur l'ensemble d'apprentissage mais importante sur l'ensemble de test). C'est le problème de sur-apprentissage ou over-fitting (pour plus de détail consulter mon cours Techniques d'Apprentissage en IA, M1SI).

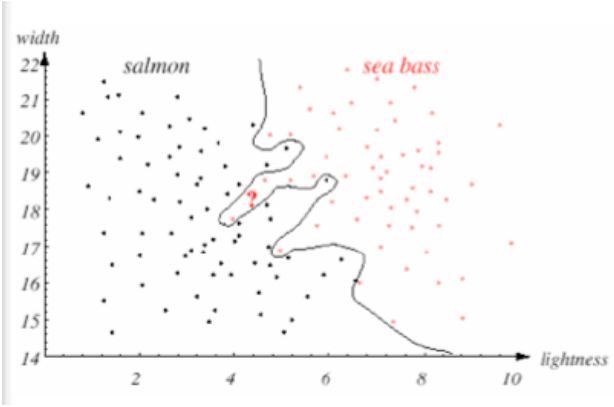


Figure 1.16 L'espace des caractéristiques (luminosité et largeur) de l'ensemble d'apprentissage pour les deux classes Bar et Saumon avec une frontière de décision complexe (sur-apprentissage)

La solution peut être un compromis entre les deux modèles. En effet, même s'il n'est pas le meilleur pour l'ensemble d'apprentissage, un modèle n'est pas aussi complexe comme celui de la figure 1.15 pourrai être meilleur pour des nouveaux exemples de test (meilleur généralisation). La figure 1.16 montre une frontière de décision (modèle) pas trop simple (sous-apprentissage) ni trop complexe (sur-apprentissage) ; c'est un compromis entre les deux : erreur satisfaisante aussi bien sur l'ensemble d'apprentissage que celui de test.

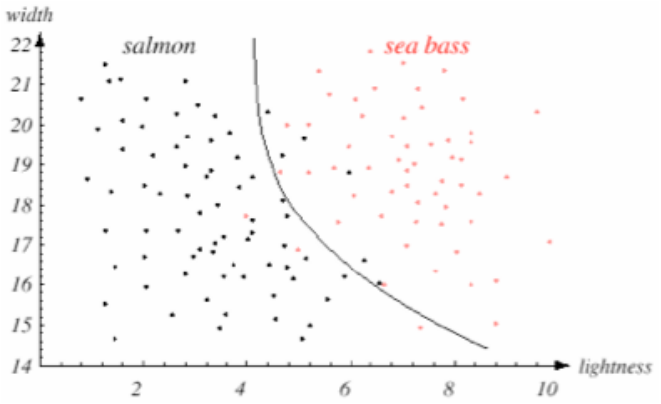


Figure 1.17 L'espace des caractéristiques (luminosité et largeur) de l'ensemble d'apprentissage pour les deux classes Bar et Saumon avec une frontière de décision pas trop complexe ni trop simple (compromis)

2 Les méthodes statistiques bayésiennes

2.1 Introduction

Une forme dans ce type de méthodes est représentée par un ensemble de valeurs numériques souvent réelles. Ces valeurs sont donc regroupées dans des vecteurs de dimension d appartiennent à l'espace des caractéristiques $\mathbb{R}^d$.

Les méthodes bayésiennes supposent que le problème de représentation de formes admet un modèle probabiliste qui permet de prendre une décision de type « plus forte probabilité d'appartenance à une classe ».

La classification dans ce type de méthodes est basée sur la règle de Bayes (d'où leur nom) qui permet d'évaluer des **probabilités à postériori** (c-à-d., après l'observation effective de certains grandeurs) connaissant les distributions de probabilités **conditionnelles** et **à priori** (c-à-d., indépendantes de toute contrainte sur la variable observée).

2.2 La décision bayésienne

On considère un échantillon (exemple) représenté par un vecteur de caractéristiques x. Il s'agit de déterminer la classe w_i à laquelle il appartient. Pour cela, on suppose connaitre :

- L'ensemble des classes $\Omega = \{w_1, w_2, \dots, w_N\}$,
- La probabilité à priori de chaque classe $P(w_i)$ qui est la probabilité d'observation de ses échantillons (exemples),
- Des fonctions de densité des caractéristiques de chaque classe $P(x/w_i)$.

La règle de bayes permet de calculer la probabilité à postériori de chaque classe, c-à-d., la probabilité conditionnée par l'observation de x, soit :

$$P(w/x) = \frac{P(w) * P(x/w)}{P(x)}$$

Comme $\sum_{i=1}^{k} P(w/x) = 1$, alors :

$$P(x) = \sum_{i=1}^{N} P(w_i) * P(x/w_i)$$

Où $P(x)$ est un facteur de normalisation qui peut être ignoré durant la décision puisque c'est le même pour toutes les classes (il ne dépend pas d'une valeur particulière w_i).

Remarque :

- Une décision idéale d_0 est celle qui à toute x de la classe $w \in \Omega$, $d_0(x) = w$

Mais en pratique, il n'y a pas de décision idéale, on cherche alors une décision optimale.

- La décision optimale consiste à choisir la décision $d(x) = w_i$ qui maximise $P(w_i/x)$. C-à-d.,

$$d : \mathbb{R}^d \rightarrow \Omega$$
$$x \longmapsto d(x) = w_i / \forall j \neq i : P(w_i/x) \geq P(w_j/x)$$
$$\Longrightarrow P(w_i) * P(x/w_i) \geq P(w_j) * P(x/w_j)$$

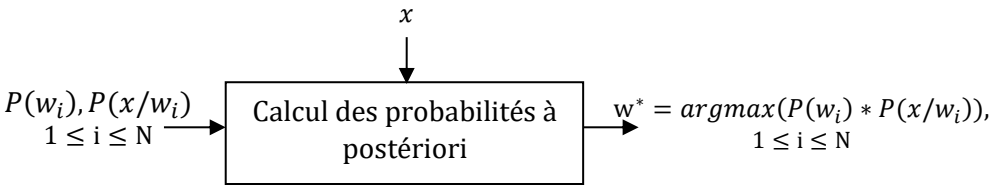


Figure 2.1 Principe de la décision bayésienne

Exemple :

……

2.3 L'apprentissage bayésien

Comme nous l'avons vu dans le cadre de la décision bayésienne, si l'on connait pour chaque classe w_i la probabilité à priori de la classe $P(w_i)$ et d'autre part, la fonction de densités de probabilités $P(x/w_i)$, alors on peut définir grâce à la décision bayésienne un classifieur optimale.

L'apprentissage consiste alors à déterminer $P(w_i)$ et $P(x/w_i)$ à l'aide des données fournies qui sont les échantillons (exemples) de la classe w_i.

2.3.1 Détermination de $P(\mathrm{w})$

$P(w)$ est soit connue à priori (par exemple, lorsque on voit que toutes les classes sont équiprobables), soit estimée facilement et avec précision sur l'ensemble d'apprentissage.

2.3.2 Détermination de $P(x/w)$

Supposons que $P(x/w)$, dépend de x et de m paramètres $\theta = \{\theta_1, \theta_2, \theta_3, \dots, \theta_m\}$. Dans ce cas, $P(x/w)$ devient une fonction de x et de θ. Le problème revient alors à estimer θ à partir du tirage aléatoire des échantillons de w, et ceci successivement pour chaque classe.

Il existe plusieurs méthodes de résolution, la plus fréquemment utilisée est celle du maximum de vraisemblance.

Estimateur vraisemblance : On suppose que $E = \{x_1, x_2, x_3, \dots, x_n\}$, l'ensemble des échantillons d'apprentissage de la classe w, et que tous ces échantillons sont tirés aléatoirement et indépendamment les uns des autres. On peut donc écrire : $P(E, \theta) = \prod_{k=1}^{n} P(x_k, \theta)$: la probabilité d'avoir tous les échantillons est le produit des probabilités individuelles, puisque les échantillons sont indépendants.

Pour le calcul de θ, l'estimateur de vraisemblance suppose de choisir θ qui maximise la vraisemblance d'être dans la classe w.

$P(E, \theta)$ est max, alors $\prod_{k=1}^{n} P(x_k, \theta)$ est max et $\sum_{k=1}^{n} \log P(x_k, \theta)$ est max

Donc, $\frac{\partial}{\partial \theta_m} [\sum_{k=1}^{n} \log P(x_k, \theta)] = 0, \forall\, m$

Estimation de θ en cas de la loi gaussienne : Pour simplifier les calculs, on suppose que l'espace des caractéristiques a une seule dimension.

Dans ce cas, il suffit de calculer, sur la base d'apprentissage, les paramètres de la loi gaussienne $\theta = \{\mu, \sigma\}$ caractérisant chaque classe, puis d'appliquer (dans la phase de reconnaissance) la formule suivante :

$$p(x/w) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Pour trouver θ, on applique la méthode de l'estimateur de vraisemblance :

On cherche donc μ et σ tels que :

$$\begin{cases} \frac{\partial}{\partial \mu}\left[\sum_{k=1}^{n} \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2}\right] = 0 \\ \frac{\partial}{\partial \sigma}\left[\sum_{k=1}^{n} \log \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_k-\mu}{\sigma}\right)^2}\right] = 0 \end{cases}$$

Après développement, on obtient :

$$\begin{cases} \mu = \frac{1}{n}\sum_{k=1}^{n} x_k \\ \sigma^2 = \frac{1}{n}\sum_{k=1}^{n} (x_k - \mu)^2 \end{cases}$$

Chaque classe est représentée donc par un moyen μ et une variance σ.

La vraisemblance $p(x/w_i)$ est calculée durant l'étape de décision suivant la formule :

$$p(x/w_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

2.4 Frontières de décision et fonction discriminantes

Le principe de décision bayésienne partitionne l'espace de caractéristiques en régions de décision. A chaque région correspond une classe. Les frontières entre les régions sont appelées frontières de décision : une frontière de décision sépare deux classes.

Problème : Comment peut-on caractériser la frontière de décision en fonction des distributions $p(w_i)$ et $p(x/w_i)$?

Solution : Utiliser les fonctions discriminantes.

On note d_i la décision bayésienne correspond à $d(x) = w_i$. On peut distinguer deux cas :

2.4.1 Cas de plusieurs classes

Un classifieur peut utiliser des fonctions discriminantes $g_i(x)$ et décider :

$\forall j \neq i, g_i(x) \geq g_j(x) \Longrightarrow d(x) = w_i$ avec g_i est la fonction discriminante caractérisant la classe w_i

Un classifieur bayésien peut utiliser comme fonction discriminante toute fonction $f(g_i(x))$ où f est monotone croissante, en particulier :

$$g_i(x) = p(x/w_i) * p(w_i)$$

$$g_i(x) = \ln p(x/w_i) + \ln p(w_i)$$

2.4.2 Cas de deux classes

Pour discriminer deux classes, on peut remplacer les fonctions $g_1(x)$ et $g_2(x)$ par une seule fonction $g(x) = g_1(x) - g_2(x)$ et décider :

$g(x) > 0 \implies d(x) = w_1$

Le classifieur bayésien peut alors utiliser :

$$g(x) = P(w_1/x) - P(w_2/x)$$

$$g(x) = ln\frac{P(w_1/x)}{P(w_2/x)}$$

$$g(x) = ln\frac{P(x/w_1)}{P(x/w_2)} + ln\frac{P(w_1)}{P(w_2)}$$

3 Les méthodes stochastiques

3.1 Introduction

Les méthodes stochastiques sont des méthodes de modélisation utilisables là où se trouvent le hasard et l'incertitude. Elles permettent l'utilisation des modèles probabilistes basés sur des processus stochastiques évoluant aléatoirement avec le temps pour traiter les problèmes à information incomplète ou incertaine.

On fait appel aux méthodes stochastiques lorsqu'on est en présence de phénomènes qu'il n'est pas possible (ou peu pratique) d'étudier de façon détaillée et déterministe. C'est notamment le cas lorsque les systèmes étudiés présentent une très grande complexité, ou lorsque on ne dispose que d'une connaissance partielle de leurs caractéristiques. Les méthodes stochastiques permettent alors les comportements en moyenne, en modélisant de façon probabiliste les parties d'un système qu'on ne peut pas décrire en détails.

3.2 Le processus stochastique

Un processus stochastique signifie, dans la théorie probabiliste, un processus aléatoire qui peut changer d'état q_i $(i = 1{:}N)$ au hasard aux instants $t = 1,2,\dots,T$. A chaque état, il possède une variable aléatoire $X(t) = q_i$, qui prend ses valeurs dans l'ensemble des observations que l'on fait dans le temps du phénomène analysé. Une évolution du système est donc une suite de transitions d'états à partir d'un état de départ $(X(1) = q_1) : q_1 \to q_2 \to \cdots \to q_T$.

La probabilité que le système passe par la suite d'états (suite de transitions) $q_1, q_2, \dots, q_T$ est calculée comme suit :

$$P(q_1, q_2, \dots, q_T) = P(q_1) * P(q_2/q_1) * P(q_2/q_1, q_2) * \dots P(q_T/q_1, q_2, \dots, q_{T-1})$$

Pour spécifier entièrement un processus stochastique, il suffit de spécifier la loi de probabilité de la première variable aléatoire q_1 (l'état lors de la première observation) et les probabilités conditionnelles $P(q_i/q_1, q_2, \dots, q_{i-1})$.

3.3 Les modèles de Markov cachés

Les modèles de Markov cachés (HMM : Hidden Markov Models) sont des méthodes stochastiques utilisées pour la modélisation et la classification dans de nombreux domaines de reconnaissance de formes et de l'apprentissage automatique, principalement dans le domaine de la reconnaissance de la parole et de l'écriture.

3.3.1 Le processus de Markov

Si l'état d'un processus stochastique au temps t ne dépend que de son état au temps t-1, on dit qu'il vérifie la propriété de Markov et on l'appelle un processus de Markov.

Propriété de Markov :

$$P(q_t/q_1, q_2, \dots, q_{t-1}) = P(q_t/q_{t-1}) \quad (3.1)$$

Un processus de Markov peut être modélisé par un modèle de Markov observable ou caché.

3.3.2 Les modèles de Markov observables

Un modèle de Markov est dit observable si les états sont directement observables.

Un modèle de Markov observable à N états peut être défini par $\lambda = (\Pi, A)$:

- Π est un vecteur à N éléments $\pi_i = P(s_0 = s_i)$, $1 \leq i \leq N$, représentants les probabilités que le processus démarre d'un état donné.
- A est la matrice de transition de taille (NxN) contenant les probabilités $a_{ij} = P(s_j/s_i)$ de passer d'un état à un autre.

Les contraintes stochastiques suivantes doivent être vérifiées :

$$\sum_{i=1}^{N} \pi_i = 1, \quad (3.2)$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \leq i \leq N, \quad (3.3)$$

L'évolution d'un processus de Markov peut être représentée par un graphe orienté et pondéré (automate) qui fait apparaitre la structure du processus selon les règles suivantes :

- Les sommets représentent les états
- Les arcs représentent les transitions. Elles sont pondérées par leurs probabilités.

Exemple : N=3

$$\Pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{bmatrix}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

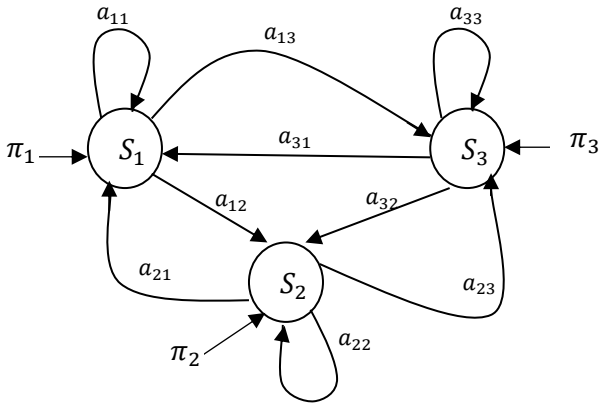


Figure 3.1 Représentations d'un modèle de Markov observable

3.3.3 Les modèles de Markov cachés

Un HMM est un processus doublement stochastique (Rabiner et Juang 1986). Il a deux propriétés principales. Premièrement, il suppose qu'une séquence d'observations $O = (o_1, o_2, \dots, o_T)$ est produite par une séquence d'états cachés $Q = (q_1, q_2, \dots, q_T)$. Autrement dit, la séquence d'états ayant généré la séquence d'observations est cachée à l'observateur, d'où son nom (caché). Deuxièmement, il vérifie la propriété de Markov qui suppose que, l'état du processus au temps t ne dépend que de son état au temps $t - 1$. En se basant sur cette propriété, on peut déduire que la probabilité $P(q_1, q_2, \dots, q_T)$ que le processus passe par la séquence d'états $Q = (q_1, q_2, \dots, q_T)$ peut être calculée comme suit :

$$P(q_1, q_2, \dots, q_T) = P(q_1) * P(q_2/q_1) * \dots * P(q_T/q_{T-1}) \quad (3.4)$$

Formellement, un HMM à N états et M symboles d'observations discrètes qui peuvent être émis par les états au cours du temps est définit par le triplet $\lambda = (\Pi, A, B)$, tel que :

- Π est un vecteur à N éléments $\pi_i = P(s_0 = s_i)$, $1 \leq i \leq N$, représentants les probabilités que le processus démarre d'un état donné.
- A est la matrice de transition de taille (NxN) contenant les probabilités $a_{ij} = P(s_j/s_i)$ de passer d'un état à un autre.
- B est la matrice d'observation de taille (NxM). Un coefficient $b_i(O_j)$ de B représente la probabilité que le symbole O_j soit émis par l'état s_i.

Les contraintes stochastiques suivantes doivent être vérifiées :

$$\sum_{i=1}^{N} \pi_i = 1, \tag{3.5}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \qquad 1 \leq i \leq N, \tag{3.6}$$

$$\sum_{j=1}^{M} b_i(O_j) = 1, \quad 1 \leq i \leq N \tag{3.7}$$

Il est à noter que cette définition concerne les HMM à densité discrète. Pour les HMM à densité continue, où on ne dispose pas de symboles d'observation discrets, la matrice d'observation B est remplacée par les paramètres de la loi de probabilité utilisée pour évaluer les probabilités d'observation. En cas de la loi multi-gaussienne (GMM pour Gaussian Mexture Model), utilisée généralement dans les problématiques de reconnaissance de formes, la matrice B est remplacée par les vecteurs moyens et les matrices de covariance des densités gaussiennes. Chaque densité de probabilité associée à un état i est calculée en appliquant la formule :

$$\mathcal{N}(\mu_i, \Sigma_i, O) = \frac{1}{(2\pi)^{\frac{P}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(O-\mu_i)^T \Sigma_i^{-1} (O-\mu_i)} \tag{3.8}$$

Où ; P est la dimension (nombre d'éléments) du vecteur O, μ_i le vecteur moyen de la fonction de densité pour l'état i, $|\Sigma_i|$ le déterminant de la matrice de covariance de la fonction de densité associée à l'état i, Σ_i^{-1} est l'inverse de la matrice de covariance de l'état i, et T le nombre moyen d'observations (trames) par séquence.

Les probabilités d'observation $b_i(O_t)$ sont calculées comme une somme pondérée des fonctions de densité gaussienne $\mathcal{N}(\mu_i, \Sigma_i, O_t)$ associées à l'état i.

Dans la littérature, les HMM qui utilisent une distribution multi-gaussiennes des probabilités d'observation sont souvent qualifiés de HMM/GMM.

Un HMM peut être également représenté par un graphe orienté et pondéré, tel que, les sommets représentent les états, les arcs indiquent les transitions entre états, et les poids des arcs correspondent aux probabilités de transition. La matrice de transition permet de définir la topologie du modèle. Un HMM peut avoir une topologie gauche-droite ou ergodique. Dans le modèle gauche-droite (cf. figure 3.2(a)), seul le bouclage sur le même état ou les transitions vers les états d'indices supérieurs sont autorisées ; les probabilités de transition $a_{ij}, avec\ j < i$ sont donc nulles. Quant à la topologie ergodique, toutes les transitions sont autorisées (cf. figure 3.2(b)).

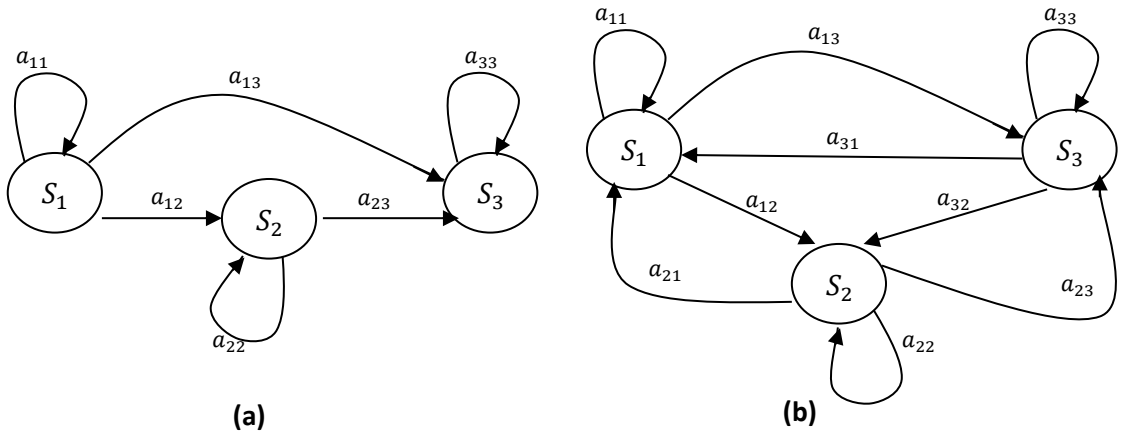


Figure 3.2 Représentation graphique d'un HMM à 3 états (a) Modèle gauche-droite, (b) modèle ergodique

3.4 Application des modèles de Markov cachés en RdF

Contrairement aux autres méthodes de classification, où une forme est représentée par un vecteur de caractéristiques, les HMM nécessitent une représentation particulière. En effet, une forme est représentée sous forme d'une suite de vecteurs de caractéristiques $O = (o_1, o_2, \dots, o_T)$ appelée séquence d'observations. Chaque vecteur (observation) correspond à une portion de la forme. Par exemple, en reconnaissance de la parole, un signal vocal est découpé en tranches de taille fixe appelées frames, à partir de chaque frame est calculé un vecteur de caractéristiques.

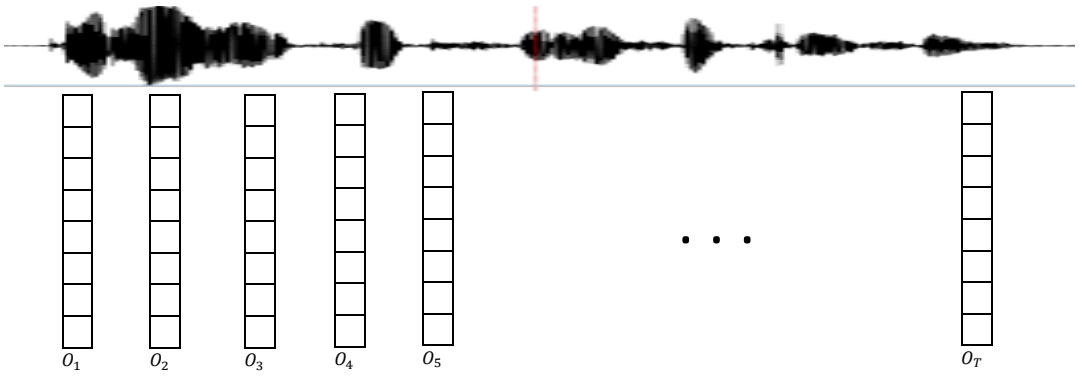


Figure 3.3 Représentation d'un signal de la parole sous forme d'une séquence d'observations

3.4.1 Les trois problèmes à résoudre par les modèles de Markov cachés

L'utilisation des HMM pour des applications réelles nécessite la résolution de trois problèmes fondamentaux qui sont bien définis par Rabiner (Rabiner, 1988 ; Rabiner, 1989) comme suit :

- Problème d'évaluation (calcul de vraisemblance) : Étant donné la séquence d'observation $O = (o_1, o_2, \dots, o_T)$ et un modèle $\lambda = (\Pi, A, B)$, comment peut-on calculer efficacement $P\ (O/\lambda)$, la probabilité d'observer la séquence O, étant donné le modèle λ ?
- Problème de décodage (reconnaissance) : Étant donné la séquence d'observation $O = (o_1, o_2, \dots, o_T)$ et le modèle λ, comment choisir la séquence d'états $Q = (q_1, q_2, \dots, q_T)$ optimale ayant généré O ?
- Problème de réestimation (apprentissage) : Comment ajuster les paramètres du modèle $\lambda = (\Pi, A, B)$ pour maximiser la vraisemblance $P\ (O/\lambda)$?

Nous allons, dans les sous sections qui suivent, présenter les solutions de ces trois problèmes.

3.4.1.1 Solution du problème d'évaluation

On veut calculer la probabilité de la séquence d'observation, $O = (o_1, o_2, \dots, o_T)$, étant donné le modèle λ, c-à-d., la vraisemblance $P\ (O/\lambda)$. La solution est l'algorithme Forward-Backward. Cet algorithme considère que l'observation peut se faire en 2 étapes :

- L'émission de la suite d'observations $(o_1, o_2, \dots, o_t)$ et la réalisation de l'état s_i au temps t : Forward.
- L'émission de la suite d'observations $(o_{t+1}, o_{t+2}, \dots, o_T)$ en partant de l'état s_i au temps t : Backward.

On considère la variable Forward, $\alpha_t(i)$, définie par :

$$\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = s_i/\lambda) \tag{3.9}$$

Cela correspond à la probabilité d'observer la séquence d'observations partielle $O = (o_1, o_2, \dots, o_t)$ et d'être dans l'état S_i au temps t, étant donné le modèle λ. On peut calculer $\alpha_t(i)$ en appliquant la procédure Forward de manière itérative, comme suit :

Algorithme Forward :

1. Initialisation

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad 1 \le i \le N \tag{3.10}$$

2. Itérations

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \quad 1 \le t \le T-1, \quad 1 \le j \le N \tag{3.11}$$

3. Terminaison

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.12}$$

La première étape permet d'initialiser la variable $\alpha_1(i)$ en tant que probabilité conjointe de l'état S_i et l'observation initiale o_1. L'étape d'itérations, qui est au cœur du calcul Forward, montre comment l'état s_i peut être atteint au temps $t+1$ à partir des N états possibles s_i $(1 \leq i \leq N)$, au temps t. Etant donné que $\alpha_t(i)$ est la probabilité que l'événement conjoint que les observations $o_1, o_2, \dots, o_t$ soient observées et que l'état au temps t soit s_i, le produit $\alpha_t(i) * a_{ij}$ est alors la probabilité de l'événement conjoint que $o_1, o_2, \dots, o_t$ soient observés et l'état s_j soit atteint au temps $t+1$ via l'état s_i au temps t. La somme de ce produit sur tous les N états possibles s_i $(1 \leq i \leq N)$, au temps t a, pour résultat, la probabilité de s_j à l'instant $t + 1$ avec toutes les observations partielles précédentes qui l'accompagnent. Une fois que cela est fait et que s_j est connu, il est facile de voir que $\alpha_{t+1}(j)$ est obtenu en prenant en compte l'observation de o_{t+1} dans l'état j, c-à-d., en multipliant la quantité additionnée par la probabilité $b_j(o_{t+1})$. Le calcul de l'équation (équation 3.11) est fait pour tous les états $j, 1 \leq j \leq N$, pour un t donné. Le calcul est ensuite itéré pour $t = 1, 2, \dots, T-1$. Enfin, l'étape de terminaison indique le calcul de $P(O/\lambda)$ en tant que la somme des variables terminales en Forward, $\alpha_T(i)$. Cela est justifié puisque, par définition nous avons,

$$\alpha_T(i) = P(o_1\, o_2,\ \dots\ o_T, q_T = s_i\ /\lambda) \tag{3.13}$$

Et, par conséquent, P (O/λ) est simplement la somme des $\alpha_T(i)$.

De la même façon, on peut considérer la variable Backward, définie comme suit :

$$\beta_t(i) = P(o_{t+1}\, o_{t+2},\ \dots\ o_T/q_t = s_i, \lambda) \tag{3.14}$$

Elle correspond à la probabilité de la séquence d'observations partielle de $t+1$ à la fin (T), étant donnés l'état s_i au temps t et le modèle λ. La variable $\beta_t(i)$ peut être calculée de manière itérative, comme suit :

Algorithme Backward :

1. Initialisation

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \tag{3.15}$$

2. Itérations

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\ \beta_{t+1}(j), \quad t = T-1, T-2, \dots, 1, \quad 1 \leq i \leq N \tag{3.16}$$

L'étape d'initialisation définit arbitrairement $\beta_T(i)$ comme étant 1 pour tout i. L'étape d'itérations montre que, pour être dans l'état S_i au temps t et prendre en compte la séquence d'observation à partir de l'instant $t+1$, il faut considérer tous les états possibles, s_j au temps $t+1$, en tenant compte de la transition de s_i à s_j (le terme a_{ij}), ainsi que l'observation o_{t+1} dans l'état j (le terme $b_j(o_{t+1})$), puis considérer la séquence d'observations partielle restante de l'état j (le terme $\beta_{t+1}(j)$).

Il est possible de combiner les deux procédures Forward et Backward pour résoudre le problème d'évaluation. Dans ce cas, la vraisemblance $P(O/\lambda)$ est calculée comme suit :

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_t(i) * \beta_t(i)\,, 1 \le t \le T, 1 \le j \le N \tag{3.17}$$

On peut également considérer les deux cas particuliers :

Si $t = 0$:

$$P(O/\lambda) = \sum_{i=1}^{N} \pi_i \beta_1(i)\,, 1 \le j \le N \tag{3.18}$$

Si $t = T$:

$$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i)\,, 1 \le j \le N \tag{3.18}$$

3.4.1.2 Solution du problème de décodage

Ce problème est également appelé problème de reconnaissance lorsqu'il est appliqué en RdF. Il existe plusieurs façons de résoudre ce problème, à savoir la recherche de la séquence d'états optimale associée à la séquence d'observations donnée. La difficulté réside dans la définition de la séquence d'état optimale, car il existe plusieurs critères d'optimalité possibles. Le critère le plus utilisé est de rechercher la meilleure séquence d'états (chemin), c-à-d., de maximiser $P(Q/0,\lambda)$, ce qui est équivalent à maximiser $P(Q,0\,/\lambda)$. Pour trouver cette séquence d'états optimale, il existe une technique formelle basée sur des méthodes de programmation dynamique. Cette technique est appelée algorithme de Viterbi.

Algorithme de Viterbi : Pour trouver la séquence optimale d'états, $Q = \{q_1, q_2, \dots, q_T\}$, ayant généré la séquence d'observations donnée $O = \{o_1, o_2, \dots, o_T\}$, il faut définir la quantité :

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_T} P[q_1 q_2 \dots q_t = i, o_1, o_2, \dots, o_t/\lambda] \tag{3.19}$$

c.à.d., que $\delta_t(i)$ est la probabilité la plus élevée d'un chemin unique, ayant généré les t premières observations et que l'état au temps t est s_i. Par induction on a :

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] . \; b_j(o_{t+1}) \tag{3.20}$$

Pour récupérer la séquence d'états, on doit garder trace de l'argument qui a maximisé l'équation 3.20, pour chaque t et j. Cela est fait via un tableau $\psi_t(j)$. La procédure complète permettant de trouver la meilleure séquence d'états peut, donc, être définie comme suit :

1. Initialisation

$$\delta_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N \tag{3.21a}$$

$$\psi_1(i) = 0 \tag{3.21b}$$

2. Itérations

$$\delta_t(j) = \max_{1 \leq i \leq N}\left[\delta_{t-1}(i) a_{ij}\right] b_j(o_t), \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \tag{3.22a}$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \leq i \leq N}\left[\delta_{t-1}(i) a_{ij}\right], \quad 2 \leq t \leq T, \quad 1 \leq j \leq N \tag{3.22b}$$

3. Terminaison

$$P^* = \max_{1 \leq i \leq N}[\delta_T(i)] \tag{3.23a}$$

$$q_T^* = \operatorname*{argmax}_{1 \leq i \leq N}[\delta_T(i)] \tag{3.23b}$$

4. Retour-arrière (Back-tracking)

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1 \tag{3.24}$$

Il est à noter que l'algorithme de Viterbi est similaire (à l'exception de l'étape de retour-arrière : Backtracking) dans la mise en œuvre à la procédure Forward. La principale différence réside dans la maximisation de l'équation 3.22a par rapport aux états précédents utilisés à la place de la sommation dans l'équation 3.11.

En RdF, l'algorithme de Viterbi peut être allégé en ignorant le calcul des $\psi_t(i)$ et l'étape du Backtracking, car les états de la séquence optimale ne sont liés à aucun phénomène physique, ce qui nous intéresse est la probabilité du chemin optimal et pas le chemin lui-même.

3.4.1.3 Solution du problème de réestimation

Ce problème est également appelé problème d'apprentissage lorsqu'il est appliqué en RAP. C'est le problème le plus difficile à résoudre, car, il n'existe aucun moyen connu de trouver

analytiquement le modèle qui maximise la probabilité de la séquence d'observations. En fait, étant donné que toute séquence d'observations finie est un exemple de la base d'apprentissage, il n'existe pas de moyen optimal d'estimer les paramètres du modèle. On peut, cependant, choisir $\lambda = (\Pi, A, B)$ de telle sorte que $P(O/\lambda)$ soit maximisé localement en utilisant des techniques de gradient ou une procédure itérative, telle que l'algorithme de Baum-Welch qui est une implémentation de la méthode Expectation-Maximisation (EM).

Les formules de réestimation de l'algorithme Baum-Welch

Soit $\xi_t(i,j)$, la probabilité d'être dans l'état s_i au temps t, et l'état s_j au temps $t+1$, compte tenu du modèle et de la séquence d'observations, c.à.d., :

$$\xi_t(i,j) = P\left[q_t = s_i, q_{t+1} = s_j / O, \lambda\right] \tag{3.25}$$

Il est clair, à partir des définitions des variables Forward et Backward, qu'il est possible de réécrire $\xi_t(i,j)$ sous la forme :

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1})\, \beta_{t+1}(j)}{P(O/\lambda)} \tag{3.26}$$

Nous avons précédemment défini $\gamma_t(i)$ comme étant la probabilité d'être dans l'état s_i au temps t, compte tenu de la séquence d'observation et du modèle. On peut donc relier $\gamma_t(i)$ à $\xi_t(i,j)$ en faisant la somme sur j, ce qui donne :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{3.27}$$

Si on fait la somme des $\gamma_t(i)$ pour $t = 1$ à $t = T-1$, on obtient une grandeur qui peut être interprétée comme le nombre espéré (dans le temps) de visites de l'état S_i ou bien le nombre de transitions espérées à partir de l'état s_i. De la même façon, la somme de $\xi_t(i,j)$ sur t (de $t = 1$ à $t = T-1$) peut être interprétée comme étant le nombre espéré de transitions entre l'état s_i et l'état s_j.

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{le nombre espéré de visites de l'état } s_i \tag{3.28a}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{le nombre espéré de transitions entre l'état } s_i \text{ et l'état } s_j \tag{3.28b}$$

En utilisant les formules ci-dessus (et le concept de comptage des occurrences d'événements), on peut donner une méthode de réestimation des paramètres d'un HMM. Un ensemble de formules de réestimation raisonnables pour Π, A et B sont :

$$\bar{\pi} = le\ nombre\ espéré\ de\ visites\ de\ l'état\ s_i au\ temps\ 1$$

$$= \gamma_1(i) \tag{3.29a}$$

$$\bar{a}_{ij} = \frac{\text{le nombre de transitions espérées entre l'état } s_i \text{ et l'état } s_j}{\text{le nombre de transitions espérées à partir de l'état } s_i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.29b}$$

$$\bar{b}_j(k) = \frac{le\ nombre\ de\ fois\ espéré\ de\ visiter\ l'état\ s_j\ et\ en\ observant\ le\ symbole\ v_k}{le\ nombre\ de\ fois\ espéré\ \ de\ visiter\ l'état\ s_j}$$

$$= \frac{\sum_{t=1}^{T} \gamma_t(j),\ \ o_t = v_k}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.29c}$$

En cas des HMM à densités continues, comme c'est montré plus haut, la matrice d'observation est remplacée par les paramètres de la loi multi-gaussienne utilisée. Dans ce cas, les paramètres à estimer sont les vecteurs moyens et les matrices de covariance. Les probabilités d'observation $\bar{b}_j(k)$ sont calculées suivant la formule suivante :

$$\bar{b}_j(k) = \sum_{m=1}^{M} \bar{c}_{jm}\, \mathcal{N}(\bar{\mu}_j, \bar{\Sigma}_j, O_t = v_k), \tag{3.30}$$

Avec,

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)} \tag{3.31}$$

$$\bar{\mu}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)\, O_t}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{3.32}$$

$$\bar{\Sigma}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \,.\, (O_t - \mu_{jk})(O_t - \mu_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)} \tag{3.33}$$

où (.)' désigne une transposition vectorielle et $\gamma_t(j,k)$ est la probabilité d'être dans l'état j à l'instant t et la $k^{ième}$ composante du mélange représentant O_t, c'est-à-dire,

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{i=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}\mathcal{N}(\mu_{jk}, \bar{\Sigma}_{jk}, O_t)}{\sum_{m=1}^{M} c_{jm}\mathcal{N}(\mu_{jm}, \bar{\Sigma}_{jm}, O_t)}\right] \tag{3.34}$$

Ces formules de réestimation seront utilisées, par la suite, dans une procédure itérative appelée algorithme de Baum-Welch, qui n'est qu'une implémentation de la méthode *Expectation-Maximisation* (EM) pour les modèles de Markov cachés.

La méthode d'Expectation-Maximisation EM

Etant donnée une séquence d'observations O, l'apprentissage d'un HMM, comme nous l'avons vu précédemment, consiste à réestimer les paramètres du modèle λ en maximisant la vraisemblance $P(O/\lambda)$. L'estimation du maximum de vraisemblance, (ML : pour Maximum Likelihood en anglais), est la méthode standard pour estimer les paramètres d'un modèle probabiliste. Selon cette méthode, le modèle estimé à partir de O et λ est le modèle λ'_{ML} tel que :

$$\lambda'_{ML} = \underset{\lambda}{\text{argmax}}(\log \mathrm{P}(\mathrm{O}/\lambda)) \tag{3.35}$$

où $log\ P(O/\lambda)$ est le logarithme de vraisemblance de la séquence observée O sachant le modèle λ. Si on suppose que la séquence O est générée par la suite d'états cachés Q, la maximisation de $log\ P(O/\lambda)$ revient alors à maximiser la quantité suivante :

$$\log \frac{\mathrm{P}(\mathrm{O},\mathrm{Q}/\lambda)}{\mathrm{P}(\mathrm{Q}/\mathrm{O},\lambda)} = \log \mathrm{P}(\mathrm{O},\mathrm{Q}/\lambda) - \log\ \mathrm{P}(\mathrm{O},\mathrm{Q}/\lambda) \tag{3.36}$$

Donc, on peut écrire :

$$\lambda'_{ML} = \underset{\lambda}{\text{argmax}}\ (\log \mathrm{P}(\mathrm{O},\mathrm{Q}/\lambda) - \log\ \mathrm{P}(\mathrm{O},\mathrm{Q}/\lambda)) \tag{3.37}$$

Vu que la séquence d'états Q est cachée, il est impossible de résoudre directement le problème du maximum de vraisemblance. Plusieurs solutions approximatives ont été proposées dans la littérature, telles que, la méthode d'Expectation-Maximisation (Dempster et al., 1977), la méthode du gradient (Levinson, 1983), les méthodes variationnelles (Jordan et al., 1999) et autres. Nous nous intéressons ici, à la méthode d'EM qui est celle la plus communément utilisée. Celle-ci considère les variables non observées (les états cachés) comme données manquantes et les remplace par leurs espérances de vraisemblance.

L'algorithme de Baum-Welch (Baum & Petrie, 1966 ; Baum & Eagon, 1967 ; Baum, 1972) est l'implémentation de la méthode d'EM pour les modèles de Markov cachés. Cet algorithme permet de réestimer, petit à petit, les paramètres d'un modèle selon la procédure itérative suivante :

Initialisation :

- Choisir un modèle initial λ_0.

Itérations :

- Étape d'évaluation de l'espérance (E) : on calcule l'espérance de la vraisemblance des données manquantes en tenant compte des dernières variables observées
- Étape de maximisation (M) : on effectue une mise à jour des paramètres en maximisant la vraisemblance trouvée à l'étape E. A la fin de cette étape on obtient un nouveau modèle$\lambda_i$qui sera réutilisé comme point de départ d'une nouvelle phase d'évaluation de l'espérance. La mise à jour des paramètres se fait par les formules de réestimation de Baum-Welch.
- Répéter E et M jusqu'à convergence ou bien atteindre un nombre maximal d'itérations.

L'algorithme EM converge de manière sûre vers un optimum éventuellement local. Le résultat final dépend de l'initialisation. Pour cette raison, dans l'étape d'initialisation, les paramètres du modèle initial (les probabilités de départ Π, les probabilités de transition A et les fonctions de densité d'observation B), ainsi que la structure du modèle (le nombre d'états et le nombre de gaussiens par états), doivent être soigneusement réglés.

3.4.2 Processus général de la reconnaissance

Etant donnés un ensemble de N classes $\Omega = \{w_1, w_2, \dots, w_N\}$ et une base d'apprentissage contenant plusieurs exemples de chaque classe. Après avoir fait l'analyse des exemples de la base d'apprentissage pour en extraire les séquences d'observations (x_j^i : exemple j de la classe i), ces dernières sont fournies au module d'apprentissage qui consiste à appliquer l'algorithme de Baum-Welch (implémentation de l'algorithme Expectation-Maximisation pour les HMM) afin de construire un HMM λ_i pour chaque classe $w_i, 1 \leq i \leq N$. Lorsque on doit reconnaitre un nouvel exemple, il est d'abord paramétrisé en utilisant la même méthode d'analyse utilisée lors de l'apprentissage, le résultat est une séquence d'observations $O = (o_1, o_2, \dots, o_T)$, qui sera ensuite fournie au module de reconnaissance pour calculer la vraisemblance $P(O/\lambda_i)$ par rapport à tous les modèles entrainés en appliquant l'algorithme Forward-Backward, ou la probabilité du meilleur chemin dans chacun des modèles en appliquant l'algorithme de Viterbi. La classe d'appartenance est celle dont le modèle est le plus vraisemblable.

La reconnaissance de formes selon les HMM se pratique suivant le processus illustré sur la figure suivante :

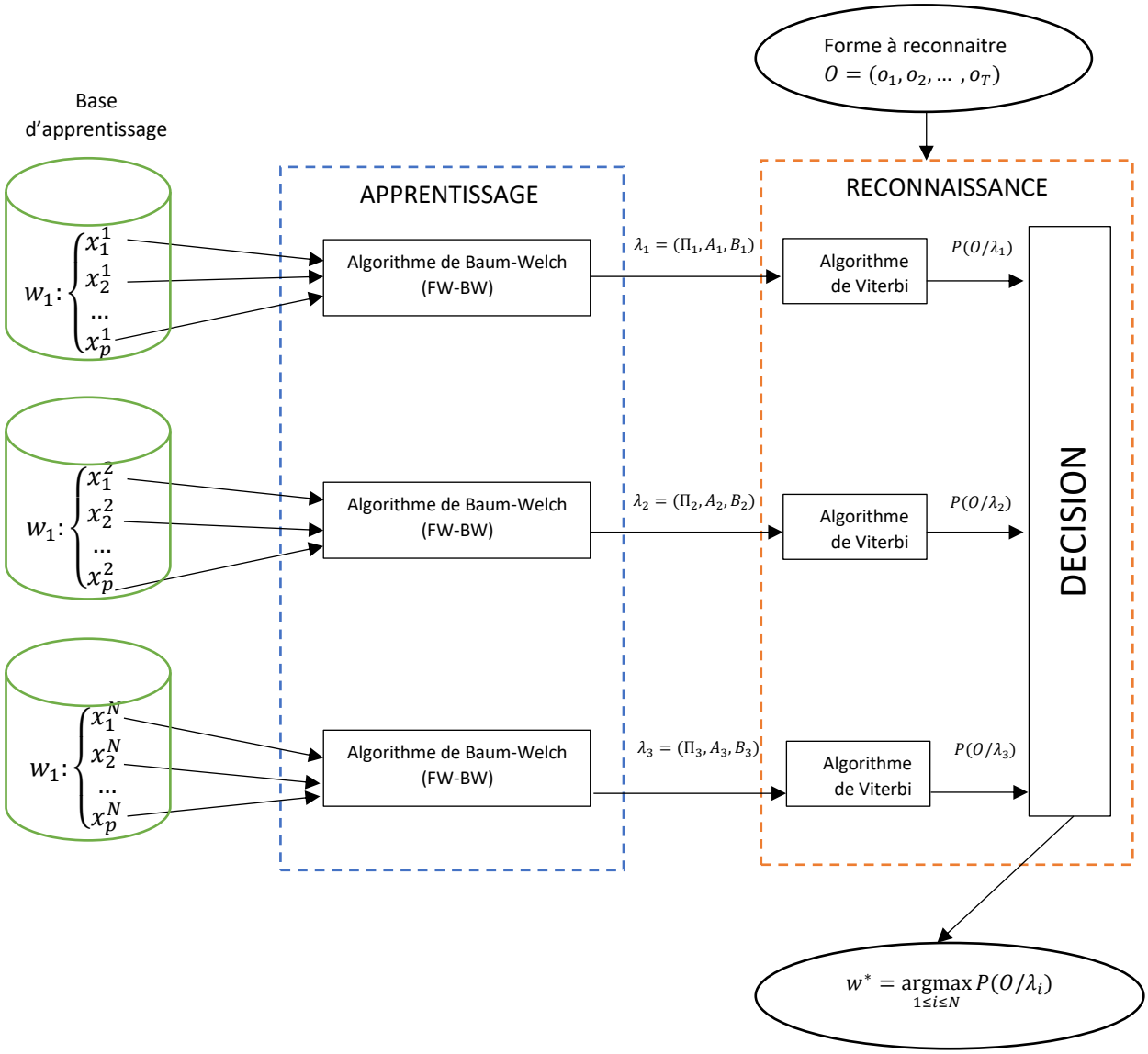


Figure 3.4 Processus de la RdF par les HMM

Université 20 août 1955- SKIKDA
Faculté des sciences
Département de l'informatique

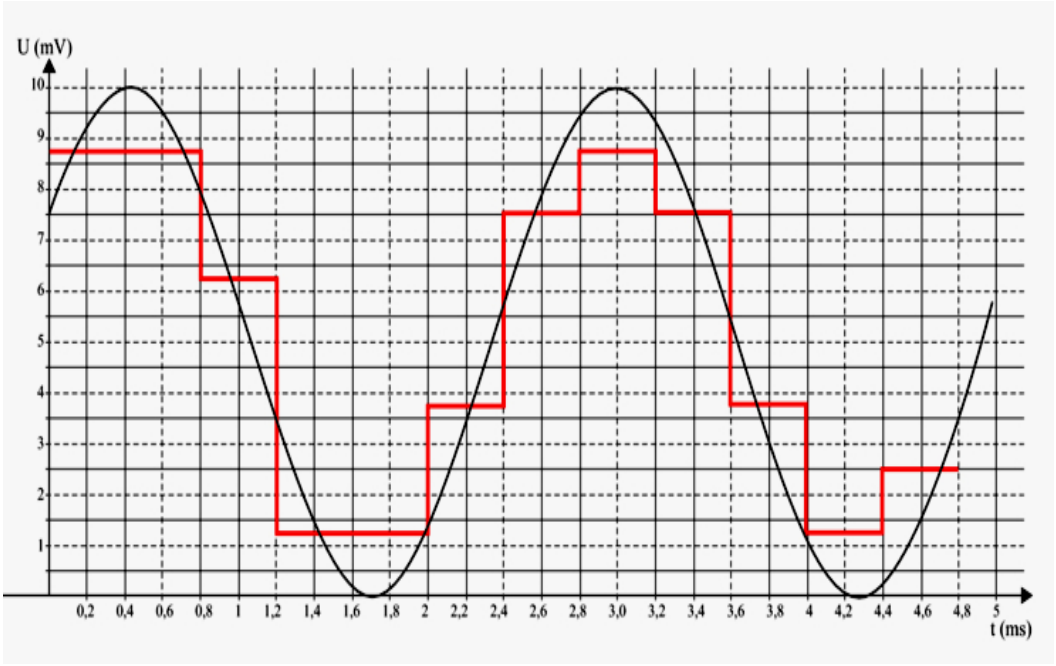
Niveau : M2 SI (2018/2019)
Matière : Techniques de RdF
Enseignante : S.HAZMOUNE

SÉRIE DE TD N°1

LE CODAGE DES DONNÉES

EXERCICE 1

On a numérisé un son, qui a une fréquence de 392 Hz. Le signal ci-dessous est le signal reconstruit à partir de sa version numérique.

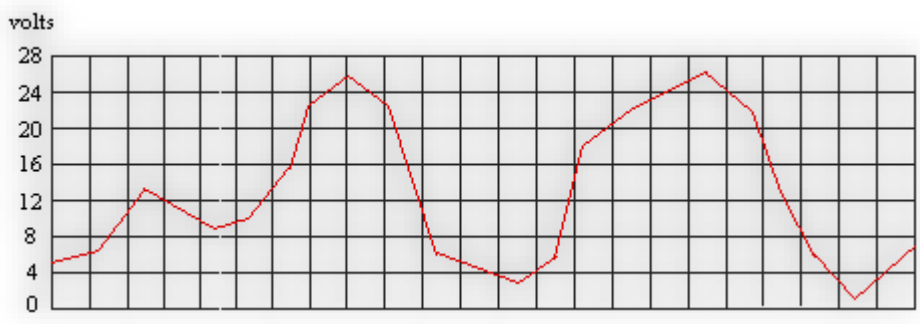


Donner :

1. La fréquence d'échantillonnage
2. Le pas de quantification
3. La résolution de quantification
4. Le nombre de bits de quantification

EXERCICE 2

Soit le signal audio suivant :



Le codage étant effectué sur 8 niveaux et l'échantillonnage étant défini sur la figure ci-dessus, en déduire le codage binaire de ce signal.

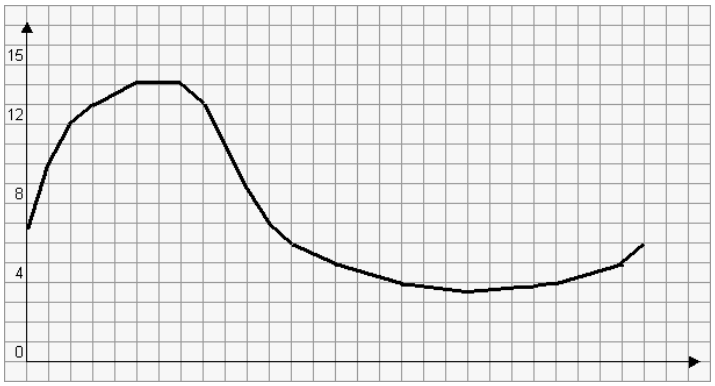
EXERCICE 3

Pour numériser un son, on utilise une fréquence d'échantillonnage de 22 KHz et on code le un codage de valeurs sur 8 bits. Pour 1 minute de son, quel est la taille correspondant en bits (on suppose qu'il n'y a pas de compression) ?

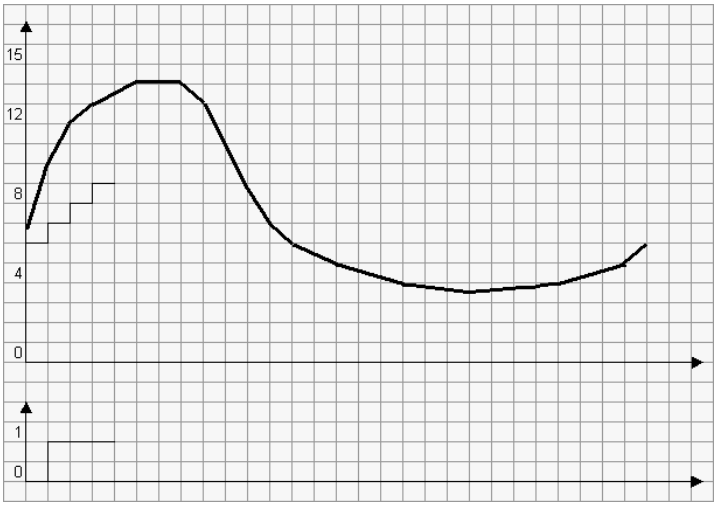
EXERCICE 4

1. Dans le cadre de l'échantillonnage de données analogiques, on peut utiliser le codage ordinaire PCM (Pulse Code Modulation) qui consiste à coder sur n bits chaque valeur mesurée de la donnée (avec approximation de quantification : on va au plus près par exemple).

Soit la donnée analogique suivante que l'on désire coder sur 4 bits (les lignes verticales indiquent les instants d'échantillonnage). En déduire le fichier binaire correspondant.

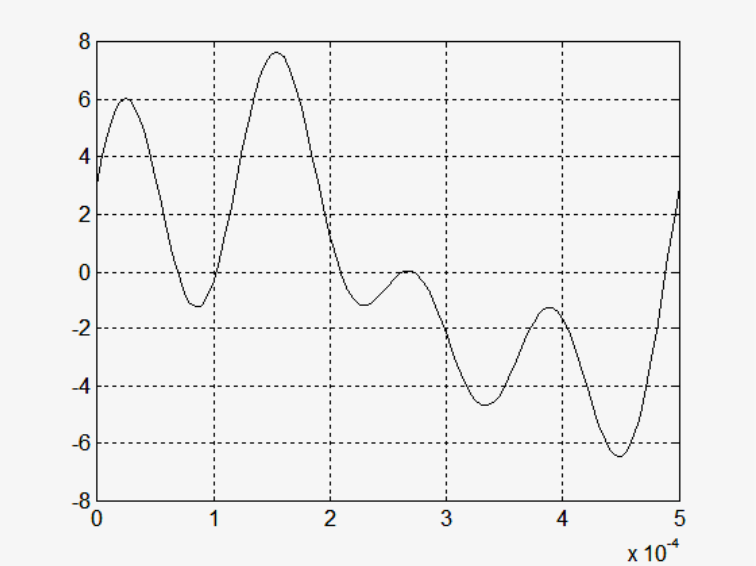


2. On peut aussi utiliser la méthode de codage appelée Modulation Delta. Cette méthode consiste à monter d'un pas de quantification à chaque échantillonnage, vers le haut si on est au-dessous de la courbe analogique, vers le bas si on est au-dessus de la courbe analogique. Le codage résultant est binaire : transition si on change de sens, pas de transition si le sens ne change pas. Le schéma ci-dessous indique le début de codage. Compléter le codage et donner le fichier binaire résultant.



EXERCICE 5

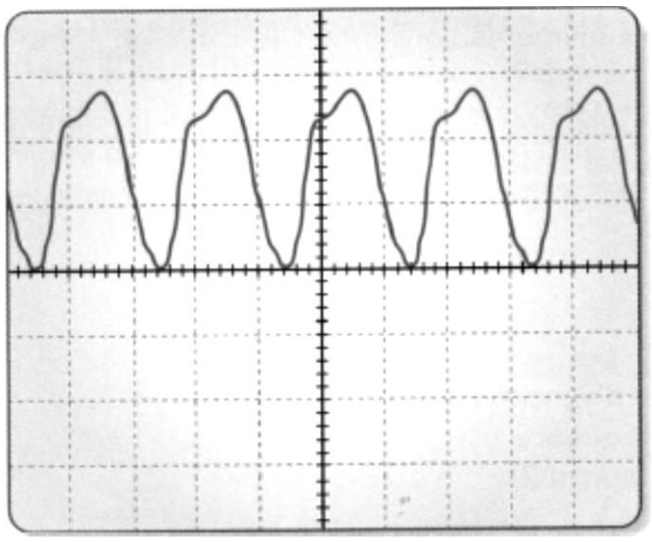
On désire numériser le signal vocal suivant, dont l'amplitude est comprise entre -8 volts et +8 volts. Ce signal est préalablement filtré par un filtre passe bas idéal de fréquence de coupure fc = 10 kHz. La quantification est effectuée sur 8 bits.



1. Proposer une valeur pour la fréquence d'échantillonnage, et représenter les échantillons prélevés sur le signal analogique.
2. Quel est le volume du fichier correspondant à 5 secondes de ce signal ?
3. Quel est le pas de quantification ?
4. Donner les valeurs binaires des quatre premiers échantillons si la quantification est en PCM.

EXERCICE 6

Le signal électrique correspondant à un son affiché sur l'écran d'un oscilloscope analogique est reproduit ci-dessous :



(Sensibilité verticale : 1,0 V/div. Sensibilité horizontale : 2,0 ms/div).

La fréquence maximale du signal est 0.5 KH.

1. Proposer une valeur pour la fréquence d'échantillonnage. Quelle est la durée séparant deux mesures consécutives ?
2. Le CAN étant de 6 bits avec une plage de mesures de 0 V à 10 V.
    a. Quelle est la résolution et le pas de quantification ?
    b. Indiquer les cinq premières valeurs que peut quantifier le convertisseur à partir de 0 V.
3. Le temps t = 0 correspond au bord gauche de l'écran de l'oscilloscope. Donner le codage binaire des 3 premiers échantillons.

Corrigé type : TD n°1 (vérifiez les calculs)

Correction de la serie de TD n°1.

EX01 :

1- La fréquence d'échantillonnage : Fe.

· C'est le nbre d'echantillons prélevés par seconde.

$Fe = \frac{1}{Te}$ % Te : la période d'échantillonnage.

$Fe = \frac{1}{0,2 \times 10^{-3}} = 5 \times 10^{3}$ HZ.

2- Le pas de quantification :

A chaque combien de tensions on peut coder une valeur, il correspond au plus petit intervalle entre 2 valeurs de tension successives.

à partir de la figure on remarque que le pas est **1,25 mv**

3- La résolution de quantification :

C'est le nombre de valeurs entières que l'on peut coder sur n bits.

ici, on ne con pas le nombre de bits, mais on peut trouver la valeur entière maximale que l'on peut coder à partir de la figure.

la valeur max de tension = 8,75 mv

et on a le pas = 1,25 mv (correspond à la valeur entière 1).

Donc : 1,25 mv ⟶ 1 ; 8,75 mv ⟶ X } $X = \frac{8,75}{1,25} = 7$

donc, **la résolution = 7 + 1 = 8** (le zéro).

c-à-d., on peut coder 8 valeurs entières de 0 à 7.

4- Le nombre de bits de quantification :

Résolution : $8 = 2^n \Rightarrow$ **n = 3 bits**

C'est le nombre de valeurs entières que l'on peut coder sur n bits.

EXO2 :

Rés = 8 ⇒ nbre de bits de quantification = 3

Avant de déduire le codage binaire, il faut d'abord déterminer les échantillons à coder. Ces derniers correspondent aux intersections de la courbe (signal) avec les lignes verticales.

Les échantillons sont donc :

$e_0 = 5v$, $e_1 = 6v$, $e_2 = 11v$, $e_3 = 12v$, ...

on cherche les valeurs entières que l'on doit coder. Pour cela, on divise par le pas de quantification.

Rés = 8 ⇒ val_max = 7.

val max = 7 ⟶ 28 v
1 ⟶ pas
} pas = $\frac{28}{7} = 4v$

e_0 : $\frac{5}{4} \simeq 1 = (001)_2$.
e_1 : $\frac{6}{4} \simeq 2 = (010)_2$.
e_2 : $\frac{11}{4} \simeq 3 = (011)_2$
e_3 : $\frac{12}{4} = 3 = (011)_2$
⋮

} le codage binaire du signal est : 001010011011....

EXO3 :

Fe = 22 KHZ , nombre de bits = 8.

* Le nombre d'échantillons codés :

1 seconde ⟶ 22×10^3 échantillons.
60 secondes ⟶ N
} $N = 60 \times 22 \times 10^3 = 1320 \times 10^3$

* La taille en bits :

1 échantillons ⟶ 8 bits
$N = 1320 \times 10^3$ ⟶ taille
} taille = $1320 \times 10^3 \times 8$

taille = 10560000 bits

EX04 :

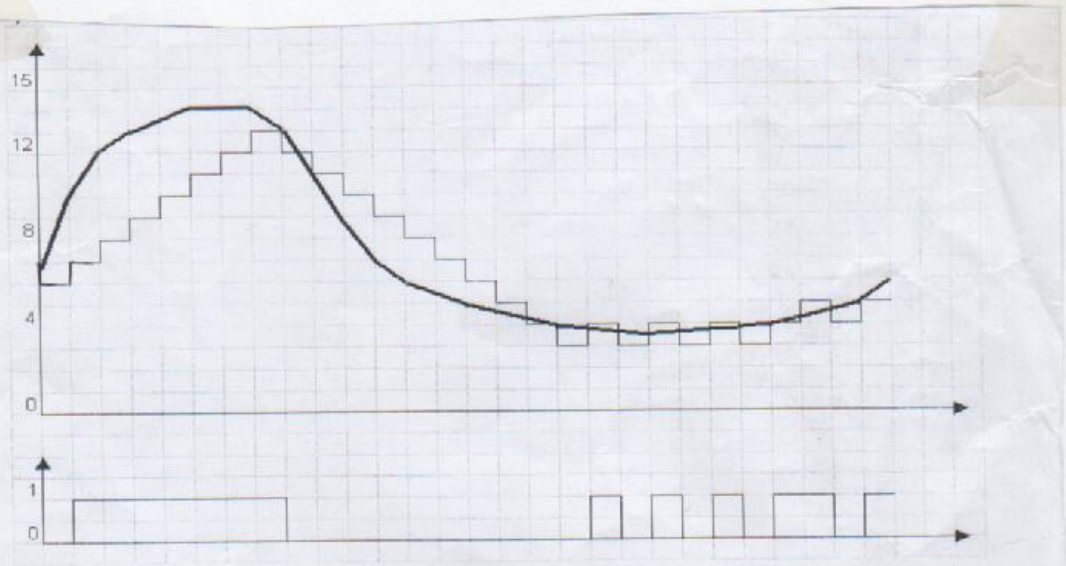
1 - Le fichier binaire correspondant au signal analogique : nbr bits = 4

$e_0 = 7 = (0111)_2$
$e_1 = 10 = (1010)_2$
$e_2 = 12 = (1100)_2$
⋮

} ⇒ le signal binaire : 0111 1010 1100 ....

RQ : ici on n'a pas divisé par le pas, parceque la valeur max de la tension peut être codée sur 4 bits. Par conséquent, le pas de quantification est 1.

2 -



Le codage est donc :

0111111100000000001010101101

EXO5: f_c = 10 KHZ , nbre de bits = 8.

1- Fe = ?

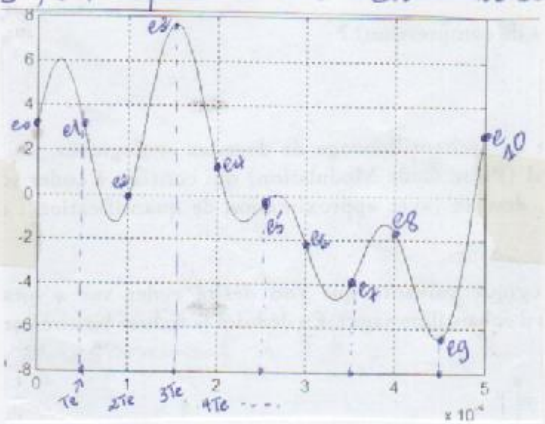
Selon la règle de Shanon : $Fe \geq 2 \times F_{max}$ ($F_{max} = f_c$)

Donc, on peut prendre $Fe = 2 \times 10\,KHZ = \boxed{2 \times 10^4\,HZ}$.

pour représenter les échantillons prélevés, il faut calculer Te

$$Te = \frac{1}{Fe} = \frac{1}{2 \times 10^4} = \boxed{0{,}5 \times 10^{-4}\,s}$$

$\Rightarrow$ A' chaque $0{,}5 \times 10^{-4}$ s, on représente un échantillons.



2- Le volume du fichier

nbr_échs = ?

1 s $\longrightarrow$ 2×10^4 echo. } nbr_échs = $5 \times 2 \times 10^4 = 10^5$ éch

5 s $\longrightarrow$ nbr_échs

1 éch $\longrightarrow$ 8 bits } Volume = 8×10^5 bits

10^5 écho $\longrightarrow$ Volume

3- Le pas de quantification :

$$pas = \frac{16}{2^8} \text{ ou } \frac{8}{128} \Rightarrow pas = 0{,}0625\,V$$

4- Les valeurs binaires de 4 premiers échantillons :

On suppose que le codage est en signe et valeur absolue :

(1 bits : signe ; n-1 bits : Val absolue ; signe : 1 : -, 0 : +)

$e_1 = 3$ volts : $\frac{3}{0{,}0625} = 48 = (\underbrace{0}_{\text{signe}}\underbrace{0110000}_{VA})_2$

$e_2 = 4$ volts : $\frac{4}{0{,}0625} = 64 = (01000000)_2$

$e_3 = 0$ volts = $(00000000)_2$ (+) ou $(10000000)_2$ (-)

$e_4 = 7{,}15$ volts : $\frac{7{,}15}{0{,}0625} = 120 = (01111000)_2$

EXO6 :

1- $Fe \geq 2 \times Fmax$ (Règle de Shanon).

$\Rightarrow Fe = 2 \times 0,5\ KHZ = 10^3\ HZ$.

La durée séparant 2 mesures consécutives : Te

$Te = \frac{1}{Fe} = \frac{1}{10^3} = 10^{-3}\ s = 1\ ms$.

2- nbr_bits = 6 bits , tension $\in [0, 10\ v]$.

a- La résolution :

$Rés = 2^n = 2^6 = 64$

pas = ?

$pas = \frac{Val\_max}{Rés} = \frac{10}{64} = 0,156\ v$

b- Les 5 premières valeurs :

$0\ v$, $0,156\ v$, $0,156 \times 2\ v = 0,312$, $0,156 \times 3\ v = 0,468$, $0,156 \times 4\ v = 0,624$

3- Le codage binaire des 3 premiers échantillons :

$e_1 = 1,1\ v$, $e_2 = 0\ v$, $e_3 = 2,4\ v$

on divise par le pas :

$e_1 : \frac{1,1}{0,156} \simeq 7 = (000111)_2$

$e_2 : \frac{0}{0,156} = 0 = (000000)_2$

$e_3 : \frac{2,4}{0,156} \simeq 15 = (001111)_2$

SÉRIE DE TD N°2

LES MÉTHODES STATISTIQUES BAYÉSIENNES

EXERCICE 1

Imaginons deux boîtes de boules.

a. L'une, A, comporte 30 boules rouges et 10 boules vertes.
b. L'autre, B, en comporte 20 de chaque sorte.

On choisit les yeux fermés une boîte au hasard, puis dans cette boîte une boules au hasard. Il se trouve être rouge. De quelle boîte a-t-il le plus de chances d'être issu, et avec quelle probabilité ?

EXERCICE 2

Soit une population de patients. Ces patients doivent être répartis en deux classes, la classe M (pour malade) et la classe S (pour sain). Les individus sont décrits à l'aide de deux attributs logiques T et C. L'attribut T a la valeur vrai lorsque la tension artérielle d'un patient est anormale et l'attribut C a la valeur vrai lorsque le taux de cholestérol d'un patient est anormal. On veut utiliser l'approche bayésienne pour classer les patients au vu de leurs descriptions. Les probabilités suivantes sont connues :

Classe K	S	M
P(k)	0.7	0.3
P(T/k)	0.25	0.7
P(C/k)	0.4	0.7

P(k): la probabilité qu'un élément de la population soit de classe k,
P(T/k) : la probabilité qu'un élément de classe k ait une tension artérielle anormale
P(C/k) : la probabilité qu'un élément de classe k ait un taux de cholestérol anormal

Nous faisons l'hypothèse supplémentaire suivante : les deux attributs T et C sont indépendants.
Question : dans cette population, décrire

1. la règle de décision majoritaire
2. la règle du maximum de vraisemblance
3. la règle de Bayes

EXERCICE 3

On suppose qu'on a deux classes de poissons A et B, et une variable aléatoire X représente le poids des poissons et qui prend ses valeurs dans R. La distribution de X dans chacune des classes suit une loi gaussienne avec les paramètres $\mu_1=2$ et $\Sigma_1=1$ pour A, et $\mu_2=2$ et $\Sigma_2=2$ pour B.

Classifier les échantillons E1 et E2 caractérisés respectivement par X=3 et X=5, en utilisant la règle du maximum de vraisemblance.

EXERCICE 4

I. On dispose de n-échantillons $\{x_1,x_2,...,x_n\}$ tirés aléatoirement et *indépendamment* les uns des autres dans une population de loi exponentielle à un seul paramètre λ.
Sachant que la densité de probabilité pour la loi exponentielle est donnée par : $p(x,\lambda) = \lambda.e^{-\lambda x}$, x est un échantillon quelconque.
Trouver une formule générale pour calculer le paramètre λ (en utilisant la méthode de l'estimateur de *vraisemblance*).

II. On vous donne une population de deux classes C_1 et C_2. On dispose de 5 échantillons de chaque classe. Les échantillons se caractérisent par leurs tailles, et on suppose que la distribution des échantillons dans les classes suit une loi exponentielle.

Les échantillons sont donnés dans le tableau suivant :

Echantillon / Classe	x_1	x_2	x_3	x_4	x_5
C_1	150	120	110	100	105
C_2	180	170	165	160	150

1. Calculez le paramètre λ pour chacune des classes C_1 et C_2.
2. Classifiez l'échantillon x=125 en utilisant la règle du maximum de vraisemblance.
3. Sachant que la probabilité à priori des classes C1 et C2 sont respectivement 0.4 et 0.6, classer x=125 en utilisant la règle de décision de Bayes.

EXERCICE 5

Nous allons utiliser le classifieur de bayes naïf pour essayer de classifier automatiquement des textes en utilisant la fréquence d'apparition des mots dans chaque catégorie de texte. Le but sera de classer les phrases dans deux classes Radio et Télévision. Pour cela, nous disposant de l'échantillon d'apprentissage suivant :

Phrase	Classe
Le programme TV n'est pas intéressant. La TV m'ennuie	Télévision
Les enfants aiment la TV	Télévision
On reçoit la TV par onde radio	Télévision
Il est intéressant d'écouter la radio	Radio
Sur les ondes, les programmes pour enfants sont rares	Radio
Les enfants vont écouter la radio ; c'est rare	Radio

Soit D= {TV, programme, intéressant, enfant, radio, onde, écouter, rare} le dictionnaire utilisé pour représenter les phrases.

1. L'estimation de P(m/k), probabilité d'apparition d'un mot m pour la classe k est donnée par la formule suivante

$$P(m/k) = \frac{N\,m/k\ + 1}{Card(Tk) + Card(D)}$$

Où Nm/k est le nombre d'occurrences du mot m dans l'ensemble des phrases de classe k, Card(D) est la taille du dictionnaire et Card(Tk) le nombre total de mots dans les phrases de classe k.

Calculer les différents P(m/k) à partir de la base d'apprentissage

2. L'estimation de P(k), probabilité d'appartenance d'une phrase à la classe k est donnée par la formule suivante :

$$P(k) = \frac{Nk}{\sum_k Nk}$$

Calculer les différents P(k) à partir de la base d'apprentissage

3. En faisant l'hypothèse que la probabilité d'une phrase (conditionnellement à la classe) est le produit de la probabilité des mots qui la compose, et en utilisant la règle de Bayes, comment sont classées les phrases de la base d'apprentissage ? (si un mot apparait plusieurs fois dans une phrase, la probabilité conditionnelle correspondante apparaitra autant de fois dans le produit)
4. Classer la phrase suivante « j'ai vu la radio de mes poumons à la TV »

EXERCICE 6

On veut concevoir un classifieur de Bayes naïf qui permet d'identifier le genre d'un locuteur (male ou femelle) en se basant sur le seul paramètre F_0 qui est la fréquence fondamentale d'une tranche d'un signal de la parole produit par le locuteur.

Le tableau suivant donne les valeurs moyennes de F_0 pour 14 locuteurs (7 femmes et 7 hommes) participant à la base d'apprentissage.

Locuteur	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Genre	F	F	F	F	F	F	F	M	M	M	M	M	M	M
F_0(HZ)	240	390	282	339	352	260	288	152	173	46	174	143	121	180

On suppose que la distribution de F_0 dans chaque genre suit une loi gaussienne.

1. Proposer un schéma général du système d'identification du genre en précisant clairement les entrées et les sorties de chaque étape.

2. Quel est le résultat de la phase d'apprentissage bayésien sur cette base ?

3. Proposer une fonction discriminante pour ce problème.

4. Identifier le genre d'un locuteur ayant une fréquence fondamentale F_0=200 HZ?

Corrigé type : TD n°2 (vérifiez les calculs)

TD n°2. Méthodes statistiques bayésiennes

EXO 1 :

A → 30 Rouge, 10 Vert ⇒ $P(\text{Rouge}/A) = \frac{30}{40} = \frac{3}{4}$; $P(\text{Vert}/A) = \frac{10}{40} = \frac{1}{4}$

B → 20 Rouge, 20 Vert ⇒ $P(\text{Rouge}/B) = \frac{20}{40} = \frac{1}{2}$; $P(\text{Vert}/B) = \frac{20}{40} = \frac{1}{2}$

% probabilités conditionnées par les classes.

On choisit une boite au hasard ⇒ $P(A) = P(B) = \frac{1}{2}$ % prob à priori.

La boule rouge est tirée de quelle boite ?

On calcule les probabilités à postériori : Règle de Bayes.

$$P(A/\text{Rouge}) = \frac{P(\text{Rouge}/A) \times P(A)}{P(\text{Rouge}/A) \times P(A) + P(\text{Rouge}/B) \times P(B)}$$

$$= \frac{\frac{3}{4} \times \frac{1}{2}}{\frac{3}{4} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}} = \frac{0,75 \times 0,5}{0,75 \times 0,5 + 0,5 \times 0,5} = 0,6$$

$$P(B/\text{Rouge}) = \frac{P(\text{Rouge}/B) \times P(B)}{P(\text{Rouge}/A) \times P(A) + P(\text{Rouge}/B) \times P(B)}$$

$$= \frac{\frac{1}{2} \times \frac{1}{2}}{\frac{3}{4} \times \frac{1}{2} + \frac{1}{2} \times \frac{1}{2}} = \frac{0,5 \times 0,5}{0,75 \times 0,5 + 0,5 \times 0,5} = 0,4.$$

On remarque $P(A/\text{Rouge}) > P(B/\text{Rouge})$ ⇒ La boule x=Rouge est tirée de la boite A avec une probabilité de **0,6**

EX02: T et C sont indépendants $\Rightarrow P(T,C) = P(T) \times P(C)$.

1- La règle de décision majoritaire :

$\forall x,\ w^* = \underset{1 \le i \le N}{\operatorname{argmax}} P(w_i)$, N : le nombre de classes.

On a 2 classes S et M.

2 attributs logiques T et C $\Rightarrow$ 4 cas possibles :

$E_1 = \begin{pmatrix} T \\ C \end{pmatrix}$, $E_2 = \begin{pmatrix} \neg T \\ C \end{pmatrix}$, $E_3 = \begin{pmatrix} T \\ \neg C \end{pmatrix}$, $E_4 = \begin{pmatrix} \neg T \\ \neg C \end{pmatrix}$.

On a $P(S) > P(M) \Rightarrow E_1, E_2, E_3$ et E_4 appartiennent à la classe S.

2- La règle de Bayes : $\forall x, w^* = \underset{1 \le i \le N}{\operatorname{argmax}} P(w_i/x) = \underset{1 \le i \le N}{\operatorname{argmax}} \frac{P(x/w_i) \times P(w_i)}{P(x)}$

$\Rightarrow \forall x, w^* = \underset{1 \le i \le N}{\operatorname{argmax}} P(x/w_i) \times P(w_i)$.

* $E_1 = \begin{pmatrix} T \\ C \end{pmatrix}$:

$P(E_1/S) \times P(S) = P(T,C/S) \times P(S) = P(T/S) \times P(C/S) \times P(S)$

$= 0,25 \times 0,4 \times 0,7 = \boxed{0,07}$

$P(E_1/M) \times P(M) = P(T,C/M) \times P(M) = P(T/M) \times P(C/M) \times P(M)$

$= 0,7 \times 0,7 \times 0,3 = \boxed{0,147}$

$0,147 > 0,07 \Rightarrow \boxed{E_1 \in M}$

* $E_2 = \begin{pmatrix} \neg T \\ C \end{pmatrix}$:

$P(E_2/S) \times P(S) = P(\neg T/S) \times P(C/S) \times P(S)$

$= [1 - P(T/S)] \times P(C/S) \times P(S)$

$= (1 - 0,25) \times 0,4 \times 0,7 = \boxed{0,21}$

$P(E_2/M) \times P(M) = [1 - P(T/M)] \times P(C/M) \times P(M)$.

$= (1 - 0,7) \times 0,7 \times 0,3 = \boxed{0,063}$

$0,21 > 0,063 \Rightarrow \boxed{E_2 \in S}$

De la même façon, on continue avec E_3 et E_4.

2- La règle du max de vraisemblance:

$\forall x,\ w^* = \underset{1 \le i \le N}{\operatorname{argmax}} P(x/w_i)$

* $E1 = \binom{T}{C}$:

$P(E1/S) = P(T/S) \times P(C/S) = 0,25 \times 0,4 = \boxed{0,1}$

$P(E1/M) = P(T/M) \times P(C/M) = 0,7 \times 0,7 = \boxed{0,49}$

$0,49 > 0,1 \Rightarrow \boxed{E1 \in M}$

* $E2 = \binom{\neg T}{C}$:

$P(E2/S) = P(\neg T/S) \times P(C/S) = (1 - P(T/S)) \times P(C/S)$.

$= (1 - 0,25) \times 0,4 = \boxed{0,3}$

$P(E2/M) = P(\neg T/M) \times P(C/M) = (1 - P(T/M)) \times P(C/M)$

$= (1 - 0,7) \times 0,7 = \boxed{0,21}$

$0,3 > 0,21 \Rightarrow \boxed{E2 \in S}$

De la même façon, on continue avec $E3$ et $E4$.

EXO3:

A: $\mu_1 = 2$, $\sigma_1 = 1$.

B: $\mu_2 = 2$, $\sigma_2 = 2$.

Règle du max de vraisemblance:

$\forall x,\ w^* = \underset{1 \le i \le N}{\operatorname{argmax}} P(x/w_i)$. N : le nombre de classes.

E1: $X = 3$.

$P(3/A) = P(3, \mu_1, \sigma_1) = \frac{1}{\sigma_1 \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{3-\mu_1}{\sigma_1}\right)^2}$.

$= \frac{1}{1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{3-2}{1}\right)^2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}} = \boxed{0,241}$

$P(3/B) = P(3, \mu_2, \sigma_2) = \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{3-2}{2}\right)^2} = \frac{1}{2\sqrt{2\pi}} e^{-\frac{1}{8}} = \boxed{0,176}$

$0,241 > 0,176 \Rightarrow \boxed{E1 \in A}$

EXO4:

I- La méthode de l'estimateur de vraisemblance:

principe: choisir λ maximisant $P(E, \lambda)$.

$P(E, \lambda) = \prod_{i=1}^{n} P(x_i, \lambda)$ % les n échantillons sont indépendants.

$P(E, \lambda)$ est max

$\Rightarrow \prod_{i=1}^{n} P(x_i, \lambda)$ est max

$\Rightarrow \sum_{i=1}^{n} \log(\lambda e^{-\lambda x_i})$ est max

$\Rightarrow \frac{\partial}{\partial \lambda}\left[\sum_{i=1}^{n} \log(\lambda e^{-\lambda x_i})\right] = 0$

$\Rightarrow \frac{n}{\lambda} - \sum_{i=1}^{n} x_i = 0 \Rightarrow \boxed{\lambda = \frac{n}{\sum_{i=1}^{n} x_i}}$

II-1 Calcul de λ pour les 2 classes:

C1: $\lambda_1 = 5/[150 + 120 + 110 + 100 + 105] = 0,0085$

C2: $\lambda_2 = 5/[180 + 170 + 165 + 160 + 150] = 0,0060$

2- $X = 125 \in ?$

Selon la règle du maximum de vraisemblance : $\omega^* = \underset{1 \leq i \leq N}{\arg\max} P(X/\omega_i)$

$P(X/C_1) = P(X, \lambda_1) = 0,0085 \times e^{-0,0085 \times 125}$

$= 0,0029$

$P(X/C_2) = P(X, \lambda_2) = 0,006\, e^{-0,006 \times 125}$

$= 0,0028$

$P(X/C_1) > P(X/C_2) \Rightarrow \boxed{X \in C_1}$

Vérifiez les calculs.

3- La règle de Bayes: $\omega^* = \underset{1 \leq i \leq N}{\arg\max} P(x/\omega_i) \times P(\omega_i)$

$P(X, \lambda_1) \times P(C_1) = 0,0029 \times 0,4 = 0,00116$

$P(X, \lambda_2) \times P(C_2) = 0,0028 \times 0,6 = 0,00168$ $\Rightarrow \boxed{X \in C_2}$

EXO5:

1- L'estimation de $P(m/K)$: (Apprentissage).

On a $P(m/K) = \frac{N_{m/K} + 1}{card(T_K) + card(D)}$.

m: un mot appartenant à D.

K = TV:

$P(TV/Télévision) = \frac{4+1}{9+8} = \frac{5}{17}$

↑ le nombre de mots du dictionnaire dans les phrases de la classe Télévision.

$P(programme/Télévision) = \frac{1+1}{17} = \frac{2}{17}$

$P(intéressant/Télévision) = \frac{2}{17}$

$P(enfant/Télévision) = \frac{2}{17}$

$P(radio/Télévision) = \frac{2}{17}$

$P(onde/Télévision) = \frac{2}{17}$

$P(écouter/Télévision) = \frac{1}{17}$

$P(rare/Télévision) = \frac{1}{17}$.

K = Radio:

$P(TV/Radio) = \frac{1}{11+8} = \frac{1}{19}$

$P(programme/Radio) = \frac{2}{19}$

$P(intéressant/Radio) = \frac{2}{19}$

$P(enfant/Radio) = \frac{3}{19}$

$P(radio/Radio) = \frac{3}{19}$

$P(onde/Radio) = \frac{2}{19}$

$P(écouter/Radio) = \frac{3}{19}$

$P(rare/Radio) = \frac{3}{19}$

2- L'estimation de $P(K)$: (Apprentissage)

$P(K) = \frac{N_K}{\sum_K N_K}$, $P(Télévision) = \frac{9}{9+11} = \frac{9}{20}$, $P(Radio) = \frac{11}{20}$

3- classification des phrase de la base d'apprentissage: Bayes.

$w^* = \underset{1 \le i \le N}{argmax} P(x/w_i) \times P(w_i)$, N: nombre de classes.

x = phrase 1:

$P(phrase\,1/Télévision) \times P(Télévision) = P(programme/Télévision) \times P(intéressant/Télévision) \times P(TV/Télévision) \times P(Télévision)$

$= \frac{2}{17} \times \frac{2}{17} \times \frac{5}{19} \times \frac{9}{20}$

Exos. (suite)

$P(\text{phrase 1} / \text{Radio}) \times P(\text{Radio}) =$

$= P(\text{programme} / \text{Radio}) \times P(\text{intéressant} / \text{Radio}) \times P(\text{TV} / \text{Radio}) \times P(\text{Radio})$.

$\times P(\text{TV} / \text{Radio})$

$= \frac{2}{19} \times \frac{1}{19} \times \frac{2}{19} \times \frac{1}{19} \times \frac{11}{20}$

On remarque que $P(\text{phrase 1} / \text{Télévision}) \times P(\text{Télévision}) >$
$P(\text{phrase 1} / \text{Radio}) \times P(\text{Radio})$

Donc, phrase 1 $\in$ Télévision.

De la même façon, on continue avec les autres phrases

⋮

4. classification d'une nouvelle phrase de Test:

$P(X / \text{Télévision}) \times P(\text{Télévision}) =$

$P(\text{radio} / \text{Télévision}) \times P(\text{TV} / \text{Télévision}) \times P(\text{Télévision})$

$= \frac{2}{17} \times \frac{5}{17} \times \frac{9}{20}$

$P(X / \text{Radio}) \times P(\text{Radio}) =$

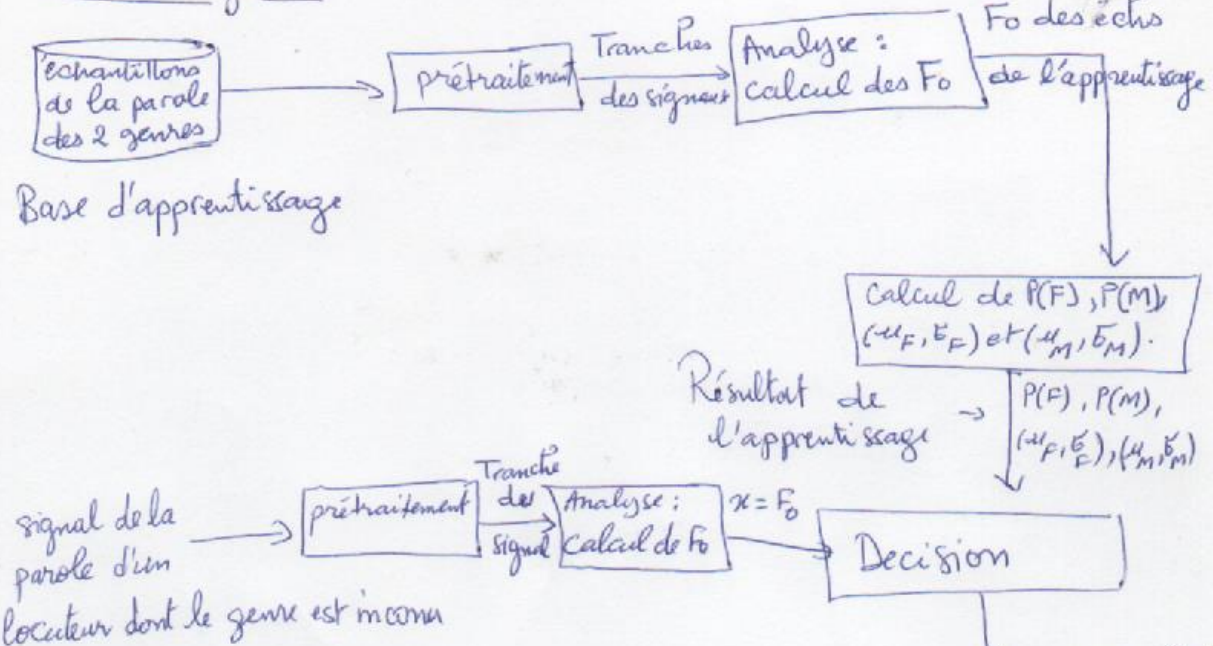
$= P(\text{radio} / \text{Radio}) \times P(\text{TV} / \text{Radio}) \times P(\text{Radio})$

$= \frac{3}{19} \times \frac{1}{19} \times \frac{11}{20}$

Donc, X $\in$ Télévision.

EXO6:

1 - Schéma général :



Avec, $P(x/\omega)$ est calculée en utilisant la formule : $P(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$

2- Le résultat de la phase d'apprentissage :

$P(F)$, $P(M)$, μ_F, μ_M, σ_F, σ_M.

$P(F) = P(M) = \frac{7}{14} = 0{,}5$

$\mu_F = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{7} \times (240 + 390 + 282 + 339 + 352 + 260 + 288)$

$= 307{,}28 \simeq 307$ Hz (pour simplifier les calcul)

$\sigma_F^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \mu_F)^2 = \frac{1}{7}\left((240-307)^2 + (390-307)^2 + \dots + (288-307)^2\right)$

$= 2517{,}42.$

$\mu_M = \frac{1}{7}(152 + 173 + 46 + 174 + 143 + 121 + 180) = 141{,}28$ Hz

$\simeq 141$ Hz

$\sigma_M^2 = \frac{1}{7}\left((152-141)^2 + (173-141)^2 + \dots + (180-141)^2\right) = 1883{,}42$

Exo6 (suite):

3- Fonction discriminante:

Selon le classifieur de Bayes;

Si $\forall i \neq j : P(w_i) \times P(x/w_i) > P(w_j) \times P(x/w_j)$ alors $w^* = w_i$

on peut donc choisir comme fonction discriminante :

$g(x) = P(w_1) \times P(x/w_1) - P(w_2) \times P(x/w_2)$.

ou $g(x) = \ln \left[P(w_1) \times P(x/w_1) - P(w_2) \times P(x/w_2) \right]$

$$g(x) = \ln \frac{P(w_1)}{P(w_2)} + \ln \frac{P(x/w_1)}{P(x/w_2)} \quad , \quad w_1 = F \text{ et } w_2 = M$$

On a $P(F) = P(M) = 0,5$, $\boxed{P(x/w) = P(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}}$

$\mu_F = 307$, $\sigma_F = 2517,42$

$\mu_M = 141$, $\sigma_M = 1883,42$.

Donc $g(x) = \ln \frac{0,5}{0,5} + \ln \frac{\frac{1}{\sqrt{2517,42 \times 2\pi}} e^{\frac{-1}{2}\frac{(x-307)^2}{2517,42}}}{\frac{1}{\sqrt{1883,42 \times 2\pi}} e^{\frac{-1}{2}\frac{(x-141)^2}{1883,42}}}$

$$= 0 - \ln(\sqrt{2517,42 \times 2\pi}) - \frac{1}{2} \times \frac{(x-307)^2}{2517,42} + \ln(\sqrt{1883,42 \times 2\pi}) + \frac{1}{2} \times \frac{(x-141)^2}{1883,42}$$

Si $g(x) > 0 \Rightarrow F$

$g(x) < 0 \Rightarrow M$

$$= -4,83 - 0,00019(x-307)^2 + 4,68 + 0,00026(x-141)^2$$

$$\boxed{g(x) = -0,15 - 0,00019(x-307)^2 + 0,00026(x-141)^2}$$

4- Identification du genre du locuteur ayant une $F_0 = 200$ Hz

$x = 200$ Hz

$g(x) = g(200) = -0,15 - 0,00019(200-307)^2 + 0,00026(200-141)^2$

$= -1,42 < 0$

$\Rightarrow$ **genre est Male**