

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Hadj Lakhdar Batna
Faculté de Technologie
Département d'Électronique



Mémoire

**Présenté pour l'obtention du diplôme de
MAGISTER en Électronique**

OPTION

Traitement de signal

Par

Dris Abdelhalim

Thème

Authentification des produits alimentaires en utilisant les méthodes de l'intelligence artificielle

Soutenu devant le jury composé de :

Dr. Louchene Ahmed	Prof.	Université de Batna	Président
Dr. Nabil Benoudjit	Prof.	Université de Batna	Rapporteur
Dr. Benzid Rédha	Prof.	Université de Batna	Examineur
Dr. Arar Djemai	Prof.	Université de Batna	Examineur
Dr. Hassam Abdelouahab	M.C.A.	Université de Sétif	Examineur
Dr. Douak Fouzi	Docteur		Invité

Je dédie ce travail à :
La mémoire de ma mère,
Mes frères et mes sœurs,
Ma femme
Tous mes amis sans exception.

Remerciements

Je tiens premièrement à remercier Allah le tout puissant de m'avoir donné le courage et la patience pour terminer ce travail.

*Je remercie ensuite mon cher encadreur Monsieur **Nabil Benoudjit**, professeur à l'université de Batna, pour m'avoir honoré par son encadrement, ses conseils précieux, sa patience et ses nobles valeurs humaines.*

*Je remercie également mon co-encadreur Dr. **Fouzi Douak**, pour ses conseils, ses lectures enrichissantes et son suivi continu.*

Mes remerciements vont également aux membres de jury :

Monsieur Louchene Ahmed, Professeur à l'Université de Batna, qui a bien voulu accepter de présider ce jury.

Monsieur Benzid Rédfha, Professeur à l'Université de Batna, qui a bien voulu examiner ce travail.

Monsieur Arar Djemai, Professeur à l'Université de Batna, qui a bien voulu examiner ce travail.

Monsieur Hassam Abdelouahab, Maître de Conférences classe (A) à l'université de Sétif, qui a bien voulu examiner ce travail.

Je remercie également l'ensemble des personnes qui ont contribué à l'aboutissement de ce travail.

Mr. DRIS Abdelhalim

Table des matières

Liste de figures	i
Liste des tableaux	iii

Introduction Générale.....	1
-----------------------------------	----------

Chapitre I

Méthodes linéaires et non linéaires

I.1. Introduction	5
I.2. Modèles linéaires et non linéaires	5
I.2.1. Analyse en composantes principales	5
I.2.2. Régression linéaire	6
I.2.3. Régression ridge	8
I.2.4. Régression ridge à noyau (KRR).....	9
I.2.5. Réseaux de neurones à fonction de base radiale (RBFN)	10
I.2.5.1. Principe	11
I.2.5.2. Apprentissage des modèles RBFN.....	12
I.2.6. Machine à vecteurs de support	12
I.2.6.1. Principe général.....	13
I.2.6.2. Cas linéairement séparable.....	13
I.2.6.3. Marge maximale.....	14
I.2.6.4. Recherche de l'hyperplan optimal	15
I.2.6.5. Cas non séparable.....	17
I.2.6.6. Choix de la fonction noyau	19
I.2.6.7. Marge souple	19
I.2.6.8. Optimisation des paramètres:.....	20
I.3. Classification multiclasse:	20
I.3.1. Un contre tous:.....	21
I.3.2. Un-contre-un (1vs1):	22
I.4. Technique de validation:	24
I.4.1. Validation simple:.....	24
I.4.2. Monte-Carlo validation Croisé:	24
I.4.3. K-fold cross-validation:	24
I.5. Conclusion:.....	25

Chapitre II

Description des bases de données

II.1. Introduction	27
II.2. La spectroscopie	27
II.2.1. Moyen infrarouge (MIR)	28
II.2.2. Proche infrarouge (PIR) :	29

II.2.2.1. Performances de la spectroscopie infrarouge:	29
II.2.2.2. Principe	29
II.2.2.3. Images multispectrales:.....	30
II.3. Description des bases de données réelles:	31
II.3.1. Base de données huile d'olive:	31
II.3.2. Images multispectrales (bases de données Fraise et Poivrons):	34
II.3.3. Base de données poisson	37
II.4. Conclusion:.....	39

Chapitre III

Etude expérimentale

III.1. Introduction	41
III.2. Les résultats expérimentaux :	41
III.2.1. Évaluation des performances :	41
III.2.1.1. Taux de classification:.....	41
III.2.1.2. Matrice de confusion	41
III.2.2. Optimisation des paramètres des modèles RBFN, SVM et KRR	41
III.2.3. Les résultats d'apprentissage:	42
III.2.4. Les résultats de test:.....	56
III.2.4.1. Résultats de la base de donnée Huile d'olive.....	56
III.2.4.1.1. Application RBFN	56
III.2.4.1.2. Application SVM.....	57
III.2.4.1.3. Application KRR	57
III.2.4.2. Résultats de la base de donnée Fraise.....	58
III.2.4.2.1. Application RBFN	58
III.2.4.2.2. Application SVM.....	58
III.2.4.2.3. Application KRR	59
III.2.4.3. Résultats de la base de donnée Poivron.....	59
III.2.4.3.1. Application RBFN	59
III.2.4.3.2. Application SVM.....	59
III.2.4.3.3. Application KRR	60
III.2.4.4. Résultats de la base de donnée Poisson	60
III.2.4.4.1. Application RBFN	60
III.2.4.4.2. Application SVM.....	61
III.2.4.4.3. Application KRR	61
III.2.5. Etude comparative	62
III.2.6. Réduction du nombre de variables	63
III.2.6.1. Réduction de dimension de la base de données huile d'olive.....	65
III.3. Conclusion.....	69
Conclusion Générale	71

Bibliographie

Liste de figures

Figure I.1: Modèle de la régression linéaire de dimension "1".....	7
Figure I.2: Illustration de la méthode de moindres carrés.....	8
Figure I.3: Présentation connexionniste d'un réseau RBFN	11
Figure I.4: Les hyperplans séparateurs pour un ensemble de point linéairement séparable.....	14
Figure I.5: L'hyperplan optimal.....	15
Figure I.6: Exemple simple de transformation d'un problème séparable vers un problème non séparable.....	18
Figure I.7: Approche une contre reste avec des zones d'indécision.....	21
Figure I.8: Résolution des cas d'indécision dans la méthode un conte tous.....	22
Figure I.9: Approche un contre un.....	23
Figure II.1: Région de spectre électromagnétique.....	28
Figure II.2: Illustration d'une image multispectrale.....	30
Figure II.3: Les spectres de la base de données huile d'olive.....	32
Figure II.4: Les spectres de l'ensemble d'apprentissage de la base de donnés huile d'olive.....	32
Figure II.5: Les spectres de l'ensemble de test de la base de données huile d'olive.....	33
Figure II.6: Répartition des 3 classes pour la base de données huile d'olive.....	33
Figure II.7: L'image RVB du Fraise et Poivron.....	34
Figure II.8: L'image RVB du Fraise et Poivron montre les pixels utilisé.....	35
Figure II.9: Les spectres de l'ensemble d'apprentissage de la base de données Fraise.....	35
Figure II.10: Répartition des 2 classes pour la base de données Fraise.....	36
Figure II.11: Les spectres de l'ensemble d'apprentissage de la base de données Poivron.....	36
Figure II.12: Répartition des 2 classes pour la base de données Poivron.....	37
Figure II.13: Images des poissons de la base de données Poisson.....	38
Figure II.14: différents mesures prises sur le corps du poisson.....	38
Figure II.15: Répartition des 7 classes pour la base de données Poisson.....	39
Figure III.1: Evolution de taux de classification de validation en fonction du nombre de neurone dans la couche cachée du classificateur RBFN sur la base huile d'olive.....	43
Figure III.2: Evolution de taux de classification de validation en fonction du nombre de neurone dans la couche cachée du classificateur RBFN sur la base fraise.....	43
Figure III.3: Evolution de taux de classification de validation en fonction du nombre de neurone dans la couche cachée du classificateur RBFN sur la base poivron.....	44
Figure III.4: Evolution de taux de classification de validation en fonction du nombre de neurone dans la couche cachée du classificateur RBFN sur la base poisson.....	44
Figure III.5: Evolution de taux de classification de validation en fonction de sigma du classificateur RBFN sur la base huile d'olive.....	45
Figure III.6: Evolution de taux de classification de validation en fonction de sigma du classificateur RBFN sur la base fraise.....	46
Figure III.7: Evolution de taux de classification de validation en fonction de sigma du classificateur RBFN sur la base poivron.....	46
Figure III.8: Evolution de taux de classification de validation en fonction de sigma du classificateur RBFN sur la base poisson.....	47
Figure III.9: Evolution de taux de classification de validation en fonction de C du	

classificateur SVM sur la base huile d'olive.....	48
Figure III.10:Evolution de taux de classification de validation en fonction de C du classificateur SVM sur la base fraise.....	48
Figure III.11:Evolution de taux de classification de validation en fonction de C du classificateur SVM sur la base poivron.....	49
Figure III.12:Evolution de taux de classification de validation en fonction de C du classificateur SVM sur la base Poisson.....	49
Figure III.13:Evolution de taux de classification de validation en fonction de sigma du classificateur SVM sur la base huile d'olive.....	50
Figure III.14:Evolution de taux de classification de validation en fonction de sigma du classificateur SVM sur la base fraise.....	50
Figure III.15:Evolution de taux de classification de validation en fonction de sigma du classificateur SVM sur la base poivron.....	51
Figure III.16:Evolution de taux de classification de validation en fonction de sigma du classificateur SVM sur la base Poisson.....	51
Figure III.17:Evolution de taux de classification de validation en fonction de lambda du KRR sur la base Huile d'olive.....	52
Figure III.18:Evolution de taux de classification de validation en fonction de lambda du KRR sur la base Fraise.....	52
Figure III.19:Evolution de taux de classification de validation en fonction de lambda du KRR sur la base Poivrons.....	53
Figure III.20:Evolution de taux de classification de validation en fonction de lambda du KRR sur la base Poissons	53
Figure III.21:Evolution de taux de classification de validation en fonction de sigma du KRR sur la base Huile d'olive.....	54
Figure III.22:Evolution de taux de classification de validation en fonction de sigma du KRR sur la base Fraise	54
Figure III.23:Evolution de taux de classification de validation en fonction de sigma du KRR sur la base Poivrons.....	55
Figure III.24:Evolution de taux de classification de validation en fonction de sigma du KRR sur la base Poissons.....	55
Figure III.25:Exemple des deux longueurs d'onde consécutives 12 et 14 nm qui sont très colinéaires.....	64
Figure III.26: Exemple des deux longueurs d'onde 1400 et 2500 nm qui ne sont pas colinéaires.....	65
Figure III.27:Évolution du taux de classification en fonction de nombre de composants principales du RBFN sur la base de données Huile d'olive.....	66
Figure III.28:Évolution du taux de classification en fonction de nombre de composants principales du SVM sur la base de données Huile d'olive.....	67
Figure III.29:Évolution du taux de classification en fonction de nombre de composants principales du KRR sur la base de données Huile d'olive.....	68
Figure III.30:illustration des résultats obtenus avec les trois classificateurs sur les quatre bases de données.....	69

Liste des tableaux

Tableau III.1: Les paramètres optimaux du classificateur RBFN.....	56
Tableau III.2: Les paramètres optimaux du classificateur SVM.....	56
Tableau III.3: Les paramètres optimaux du classificateur KRR.....	56
Tableau III.4: Matrice de confusion obtenue par l'application RBFN sur base de données Huile d'olive.....	57
Tableau III.5: Matrice de confusion obtenue par l'application SVM sur base de données Huile d'olive.....	57
Tableau III.6: Matrice de confusion obtenue par l'application KRR sur base de données Huile d'olive.....	57
Tableau III.7: Matrice de confusion obtenue par l'application RBFN sur base de données Fraise.....	58
Tableau III.8: Matrice de confusion obtenue par l'application SVM sur base de données Fraise.....	58
Tableau III.9: Matrice de confusion obtenue par l'application KRR sur base de données Fraise.....	59
Tableau III.10: Matrice de confusion obtenue par l'application RBFN sur base de données Poivron.....	59
Tableau III.11: Matrice de confusion obtenue par l'application SVM sur base de données Poivrons.....	60
Tableau III.12: Matrice de confusion obtenue par l'application KRR sur base de données Poivrons.....	60
Tableau III.13: Matrice de confusion obtenue par l'application RBFN sur la base de données Poisson.....	61
Tableau III.14: Matrice de confusion obtenue par l'application SVM sur la base de données Poisson.....	61
Tableau III.15: Matrice de confusion obtenue par l'application KRR sur la base de données Poisson.....	62
Tableau III.16: Comparaison des résultats obtenus pour la base de données Fraise.....	63
Tableau III.17: Comparaison des résultats obtenus pour la base de données Poivron.....	63
Tableau III.18: Matrice de confusion obtenue par l'application RBFN sur base de données Huile d'olive à dimension réduite.....	66
Tableau III.19: Matrice de confusion obtenue par l'application SVM sur base de données Huile d'olive à dimension réduite.....	67
Tableau III.20: Matrice de confusion obtenue par l'application KRR sur base de données Huile d'olive à dimension réduite.....	68
Tableau III.21: résultats de classification de quatre bases de données obtenus avec les trois classificateurs.....	69

Introduction générale

Introduction générale

Les problèmes d'authenticité dans la filière agroalimentaire ne sont pas récents et remontent même aux civilisations grecque et romaine [1]. L'authenticité d'un produit peut être définie par sa conformité à sa définition.

Lorsque l'on prend le cas des produits alimentaires, plusieurs problématiques d'authenticité se posent : conformité à l'espèce ou à la variété, au contenu présumé, à son origine naturelle ou artificielle, à son mode de culture biologique ou conventionnelle, la non-adultération, ou encore l'origine géographique. La raison majeure de la fraude alimentaire est économique. Il s'agit de générer des profits supérieurs avec des produits de moins bonne qualité.

Pour des questions de sûreté et d'équité commerciale, la réglementation de l'industrie agroalimentaire évolue en permanence. Les mesures législatives ont pour but d'assurer au consommateur la loyauté de l'information : dénomination, caractéristiques, provenance des produits. La détection et la prévention des fraudes, par les autorités compétentes, utilisent des contrôles adaptés au produit et au paramètre suspect.

Par ailleurs, l'industrie, toujours pressée par les délais, souhaite s'affranchir des contrôles de la qualité de sa filière : de la réception de la matière première au produit fini en sortie d'usine, en passant par la maîtrise de la qualité du procédé. Pour des questions économiques de rentabilité ces contrôles doivent être menés de manière rapide, simple, sûre et peu coûteuse.

Ainsi, les méthodes d'analyse rapides (par exemple, la spectroscopie de masse, spectroscopie de la lumière, de l'imagerie hyperspectral) pour contrôler un nombre important de paramètres sur des produits à différents stades de leur fabrication sont une aide précieuse, génératrice de profits [2].

L'objectif de ce mémoire est l'authentification rapide et fiable des produits alimentaires par des modèles mathématique [3]. L'authentification de propriétés chimiques contenues dans un produit alimentaire, sera hautement souhaitable pour les fabricants et pour ceux qui sont impliqués dans la protection de la marque car un produit alimentaire peut être vendu avec un prix élevé alors qu'il est considéré comme produit alimentaire contrefait [4].

Le problème de la classification par l'intermédiaire de l'information spectrale peut être adressée à l'aide de nombreuses techniques différentes, du traditionnel K-plus proche voisin ou linéaire (ou quadratique) analyse discriminante à base de PLS, ou des approches plus sophistiquées telles que les machines à vecteurs de support ou séparateurs à vaste marge (en

anglais Support Vector Machine, SVM) et les réseaux de neurones artificiels. Dans de nombreux cas, les spectres disponibles, indépendamment de leur origine (par exemple, la spectroscopie de masse, spectroscopie de la lumière, de l'imagerie hyperspectral), nécessite plusieurs prétraitements avant que toute méthode de classification puisse être appliquée. Pour la classification supervisée, l'analyse en composantes principales (ACP) est couramment utilisée pour réduire la dimension des données des spectres. Après la réduction des variables, plusieurs types d'algorithmes sont appliqués pour construire un modèle de classification par exemple les réseaux de neurones à fonction de base radiale (en anglais Radial Basis Function Neural network RBFN), les machines à vecteur de support SVM, régression ridge à noyau (en anglais Kernel Ridge Regression KRR).

Quatre bases de données sont utilisées dans ce mémoire, les spectres proches infrarouges (en anglais Near InfraRed NIR) (Huile d'olive) et les images multispectrales représentant des fruits et des légumes (Fraise et poivron), et la quatrième base de données est représentée par des mesures morphologiques prises sur le corps du poisson.

L'organisation générale du mémoire est décrite ci-dessous :

Le chapitre I est dédié à la présentation générale des modèles linéaires tel que L'analyse en composantes principales (ACP), la régression linéaire multiple (en anglais Multiple Linear Regression MLR) et la régression ridge (RR), et des modèles non linéaires tel que le ridge régression à noyau (KRR), les réseaux de neurones à fonction de base radiale (RBFN) et les machines à vecteur de support(SVM).

Le chapitre II est consacré à la description des bases de données. Dans un premier temps il présente une description de la spectrophotométrie et les images multispectrales, dans un deuxième temps une description des bases de données réels.

Le chapitre III présente une étude expérimentale sur les différentes bases de données, ainsi qu'une étude comparative des performances obtenues lors de l'utilisation de différents classificateurs. La deuxième partie de ce chapitre est consacré à l'étude du problème de grande dimension.

La dernière partie de ce mémoire est consacrée à la conclusion générale où nous montrons les performances des classificateurs étudiés dans ce travail pour chaque base de données, suivi

par une étude de la sensibilité des classificateurs à la malédiction de la dimensionnalité. Et nous terminons par quelques perspectives telles que l'utilisation du classificateur LWKRR (en anglais Local Weighted Kernel Ridge Regression), la stratégie de fusion, et l'optimisation des paramètres avec l'algorithme génétique.

Chapitre I

Méthodes linéaires et non linéaires

I.1. Introduction

Ce chapitre va aborder les différentes notions qui seront utilisées dans ce travail. Dans un premier temps, une présentation des modèles linéaires (ACP, régression linéaire et régression ridge) et des modèles non linéaires (SVM, RBFN et KRR) par leur principe ainsi que l'optimisation des paramètres de chaque modèle. Et par la suite Nous mettrons en évidence les techniques de classification multiclass. Deux technique sont étudiées la première un contre un et la deuxième un contre tous. Et nous terminons ce chapitre par les techniques de validations telles que la validation simple, la validation croisée et k-fold validation.

I.2. Modèles linéaires et non linéaires intelligence

I.2.1. Analyse en composantes principales

L'analyse en composantes principales, est une technique dont le but est d'expliquer la structure covariante d'un jeu de données multivariées par la contraction du jeu de données dans un espace de plus petite dimension dont les nouvelles variables sont indépendantes (non corrélées) [5]. La base du système est orthogonale. On détermine les vecteurs propres de la matrice carrée R résultant de la multiplication de la matrice du jeu de données X par sa transposée X^T . Les vecteurs propres sont appelés composantes principales, et sont orthogonaux entre eux, leur covariance étant nulle. Géométriquement, cette transformation équivaut à une rotation des axes dans l'espace des m dimensions (variables d'origine) dans un nouvel espace à p dimensions dont les axes décrivent la variance maximum des données. L'origine des axes se trouve alors au centre du nuage de points dans l'espace des mesures. Le premier axe principal sera celui qui rendra compte de la variance la plus grande (les valeurs propres étant classées par ordre de valeurs décroissantes). La variance du nouveau jeu de données est fonction des valeurs des éléments de U , la somme des carrés des éléments de chaque vecteur propre sera égale à 1, soit par exemple pour le premier vecteur propre

$$u_{11}^2 + u_{12}^2 + u_{13}^2 + u_{14}^2 + \dots + u_{1m}^2 = 1 \quad \text{I.1}$$

Pour obtenir les coordonnées des données dans le nouvel espace ("principal components scores" Y_{ij}), on projette chaque vecteur individu (x_1, \dots, x_m) de X (avec $i=1:n$, nombre d'individus, et $j=1:m$, nombre de variables initiales) sur chaque composante principale en le multipliant par les éléments (u_j) de chaque vecteur propre U (composante principale).

Considérons la projection d'un individu x_i sur l'axe principal 1 :

$$y_{i1} = \sum_{j=1}^m u_{1j} \cdot x_{ij} \quad \text{I.2}$$

Ce qui donne d'une manière générale exprimée sous forme matricielle

$$X \cdot U = Y_r \quad \text{I.3}$$

où Y_r est la matrice résultante des données dans le nouvel espace, X la matrice $n \times m$ des données de départ, et U la matrice carrée contenant les composants des vecteurs propres. On peut en donner la forme éclatée plus claire:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ x_{31} & x_{32} & \cdots & x_{3m} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \bullet \begin{bmatrix} u_{11} & \cdots & u_{1m} \\ \vdots & \vdots & \vdots \\ u_{m1} & \cdots & u_{mm} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ y_{31} & y_{32} & \cdots & y_{3m} \\ \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & \cdots & y_{nm} \end{bmatrix} \quad \text{I.4}$$

Bien que pour m variables d'origine, m composantes principales soient calculées à fin de l'ACP, il est souvent possible de ne retenir que les 3 ou 4 premières qui à elles seules recèlent la majeure partie de l'information du jeu de données [6]. Combien de composantes principales retenir est subjectif et reste à la discrétion de l'utilisateur. Pour aider à la décision, on peut supprimer les composantes qui n'amènent que peu d'information supplémentaire à celles déjà retenues, où bien se fixer un pourcentage d'information à atteindre, 90% par exemple.

I.2.2. Régression linéaire

Le but de la régression linéaire est de trouver la relation de détermination de Y par k variables explicatives X_1, \dots, X_k , où k est le nombre de variables, la relation la plus simple à envisager est le modèle linéaire suivant:

$$Y = b_0 + b_1 X_1 + \dots + b_k X_k + e = Y' + e \quad \text{I.5}$$

L'écart " e " traduit l'inadéquation du modèle par rapport à la vraie valeur de Y , il s'appelle le résidu de la régression. Y' est la variable estimée de Y par le modèle linéaire. b_0, b_1, \dots, b_k sont les coefficients de régression. Le modèle de la méthode MLR peut s'écrire sous la forme matricielle suivante [7, 8]:

$$Y = Xb + e \quad \text{I.6}$$

Où X est la matrice des variables explicatives x_1, x_2, \dots, x_k augmentée avec une colonne de 1, " b " est le vecteur des paramètres inconnus b_0, b_1, \dots, b_k du modèle. Prenons comme

exemple la régression linéaire de la dimension "1", qui a l'avantage d'être plus aisément représentable (voir par exemple La Figure I.1): sont donnés des couples de nombre réels (X_i, Y_i) (c'est l'échantillon) et l'on essaie d'y ajuster une droite au nuage de points (X_i, Y_i) (c'est le modèle) quelle droite peut-on alors choisir, qui s'ajustera au mieux sur le nuage de points? Une réponse classique sera donnée par la méthode des moindres carrés. Il s'agit de considérer pour une droite quelconque, la somme des carrés des différences entre les observations Y_i et les projections verticales de ces observations sur la droite. On considère ensuite la droite qui minimise la somme de ces carrés, d'où le nom de la méthode. Une illustration de cette méthode est donnée dans la Figure I.2.

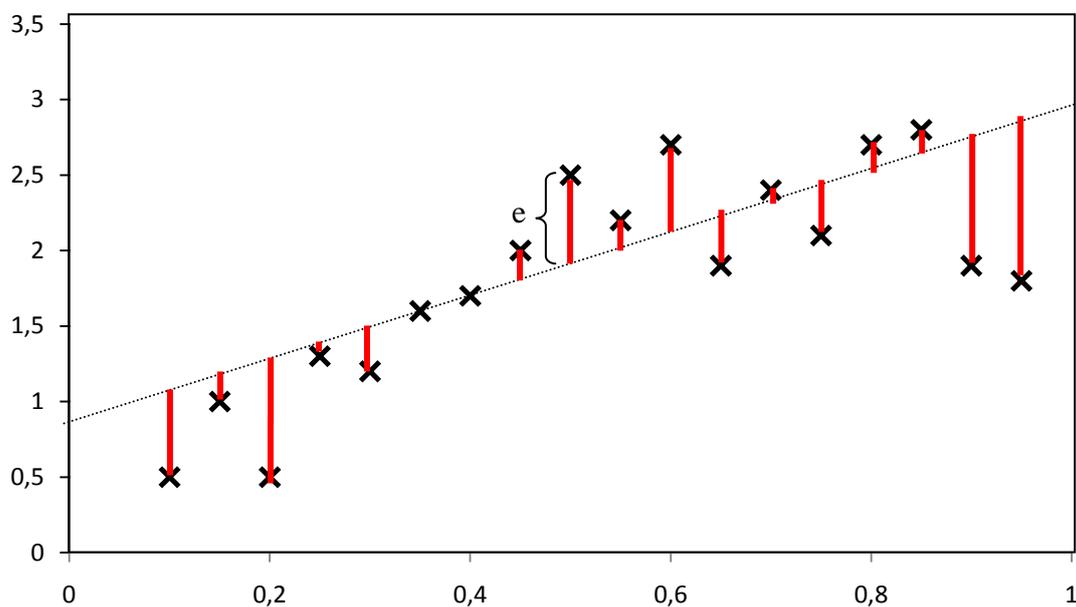


Figure I.1. Modèle de la régression linéaire de dimension "1"

(L'échantillon est représenté par les points, la fonction f est représentée par la droite en pointillés. Le bruit est alors représenté par les segments rouges verticaux).

La distance "e" dans la Figure I.1 représente l'erreur de prédiction d'un exemple.

L'estimation du meilleur modèle consiste à trouver la fonction qui minimise les résidus dont la méthode d'estimation la plus couramment utilisée est celle des moindres carrés résultant de la minimisation de la somme des carrés des résidus.

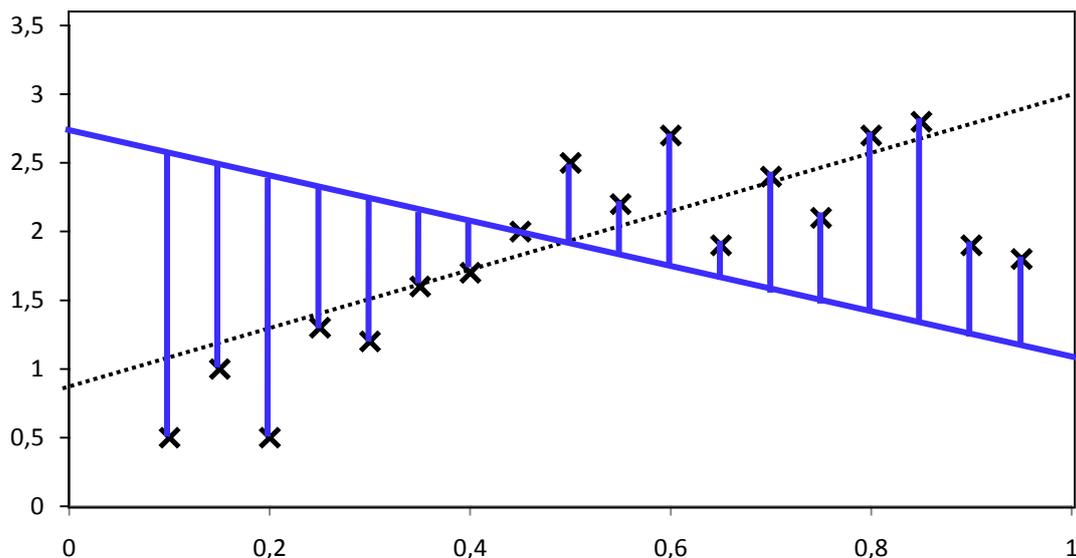


Figure I.2: Illustration de la méthode de moindres carrés.

L'échantillon est représenté par les points, la droite ayant généré les données est la droite en pointillés. La droite de régression est la droite bleue et les erreurs sont les segments bleus verticaux. On cherche donc à minimiser la somme des carrés des longueurs des segments bleus.

Généralisons l'exemple précédent. Supposons que l'on observe n points X_i , chaque point ayant k coordonnées (avec $k < n$). Pour chaque point X_i , on observe un nombre réel Y_i . On peut donc former la matrice X , de taille $n \times k$, dont la ligne i est constituée des coordonnées de X_i ainsi que le vecteur y dont la i ème coordonnée est Y_i . Le problème de régression se pose donc ainsi: trouver un vecteur β (de taille k) tel que le vecteur $X\beta$ approche au mieux Y . La méthode de moindres carrés vise alors à chercher une solution qui minimise la quantité [9].

$$L(w, b) = \sum_{i=1}^n (y_i - \langle w, x_i \rangle - b)^2 \quad \text{I.7}$$

La solution à ce problème de minimisation se trouve aisément et l'on obtient comme valeur de w :

$$\hat{w} = (X^T X)^{-1} X^T Y \quad \text{I.8}$$

I.2.3. Régression ridge

La régression ridge est une méthode de résolution des problèmes de régression linéaire mal conditionnés. Ces problèmes souffrent, dans le calcul de la matrice inverse $(X^T X)^{-1}$, d'instabilité numérique en présence de bruit dans les données [10].

Comme il a été vu précédemment, la régression par la méthode des moindres carrés consiste à faire le calcul suivant:

$$\hat{w} = (X^T X)^{-1} X^T Y \quad \text{I.8}$$

Le problème, c'est que parfois la matrice $X^T X$ est singulière [11, 12] : l'estimateur obtenu a alors une variance très élevée. On peut contourner ce problème de singularité en ajoutant une constante à toutes les valeurs propres de la matrice $X^T X$, et en calculant :

$$\hat{w} = (X^T X + \lambda I)^{-1} X^T Y \quad \text{I.9}$$

Où I est la matrice identité et λ est un paramètre de régularisation positif [13]. Nous devons résoudre un système de k équations linéaires à k inconnus. Cette solution permet de réduire la variance, en revanche l'estimateur obtenu est biaisé.

On peut aussi présenter la régression ridge comme une régression pénalisée où on cherche w qui minimise:

$$L(w, b) = \lambda \langle w, w \rangle + \sum_{i=1}^l (\langle w, x_i \rangle + b - y_i)^2 \quad \text{I.10}$$

L'ajout du terme pénalisant permet d'éviter que les coefficients de \hat{w} soient trop grands. Le paramètre λ règle le compromis entre l'attache aux données et la pénalité. Une petite valeur de λ signifie que la solution s'ajustera très près des données sans causer de pénalités importantes. Dans le cas contraire, une forte valeur induit que l'ajustement sera sacrifié s'il requiert des coefficients importants. Le biais ainsi introduit dans l'estimateur favorise les solutions usant de petits coefficients, l'effet sera de lisser la solution alors que des coefficients importants rendent la solution plus "variable".

I.2.4. Régression ridge à noyau (KRR)

La régression ridge à noyau est une méthode de régression non linéaire qui consiste à appliquer au vecteur d'entrée une transformation non linéaire de l'espace d'origine vers un autre espace de grande dimension tel que : $K(x, z) = \langle \phi(x), \phi(z) \rangle$ où x et z sont deux points de l'espace d'origine et $\phi(x)$ et $\phi(z)$ leur transformation [9].

Le problème d'optimisation correspondant à la régression ridge à noyau est le suivant:

$$\text{Minimise} \quad \lambda \|w\|^2 + \sum_{i=1}^l \zeta_i^2 \quad \text{où} \quad \zeta_i = y_i - \langle w \bullet \phi(x_i) \rangle \quad \text{I.11}$$

Ceci peut se résoudre par la méthode de Lagrange où le lagrangien est donné par:

$$L(w, \zeta, \alpha) = \lambda \|w\|^2 + \sum_{i=1}^l \zeta_i^2 + \sum_{i=1}^l \alpha_i (y_i - \langle w \cdot \phi(x_i) \rangle - \zeta_i) \quad \text{I.12}$$

En annulant les dérivées partielles du lagrangien on obtient:

$$w = \frac{1}{2\lambda} \sum_{i=1}^l \alpha_i \phi(x_i) \quad \text{I.13}$$

$$\zeta_i = \frac{\alpha_i}{2}$$

En réinjectant ces valeurs dans l'équation I.13 on obtient:

$$\alpha = 2\lambda(K + \lambda I)^{-1} y \quad \text{I.14}$$

La fonction de prédiction finale est:

$$f(x) = k'(K' + \lambda I)^{-1} y \quad \text{I.15}$$

Pour résoudre un problème de classification la fonction de prédiction finale de régression ridge à noyau est donnée par:

$$f(x) = \text{signe}(k'(K' + \lambda I)^{-1} y) \quad \text{I.16}$$

Où K est la matrice de Gram associée au noyau k sur l'ensemble d'apprentissage où $K_{ij} = \langle \phi(x_i) \cdot \phi(x_j) \rangle$, k est un vecteur où $k_i = \langle \phi(x_i) \cdot \phi(x_i) \rangle$, $i = 1, \dots, n$, n est le nombre d'échantillons d'apprentissage, I est la matrice d'identité et λ est le paramètre de ridge $\lambda \in \mathfrak{R}$ $\lambda > 0$. Le choix typique de la fonction noyau est la fonction à base radiale (Gaussienne) [14,15], qui donne les composants de la matrice noyau par la relation suivante:

$$K_{ij} = K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \quad \text{I.17}$$

où σ représente la largeur de noyau.

I.2.5. Réseaux de neurones à fonction de base radiale (RBFN)

Les réseaux de neurones à fonctions de base radiales (RBFN) sont des modèles connexionnistes simples à mettre en œuvre et assez intelligibles. Ils sont très utilisés pour la régression et la discrimination (classification). Leur propriétés théoriques et pratiques ont été étudiées en détail depuis la fin des années 80 ; il s'agit certainement, avec le Perceptron multicouche, du modèle connexionniste le mieux connu.

I.2.5.1. Principe

Une fonction de base radiale (RBFN) est une fonction ϕ symétrique autour d'un centre μ_j :

$\phi_j(x) = \phi_j(\|x - \mu_j\|)$, où $\|\cdot\|$ est une norme [16]. Par exemple, la fonction gaussienne est une

RBFN avec la norme euclidienne et $\phi(r) = \exp(-\frac{r^2}{2\sigma^2})$. En général, les RBFN sont

paramétrées par σ qui correspond à la « largeur » de la fonction :

$$\phi_j(x) = \phi_j(\|x - \mu_j\|, \sigma_j) \quad \text{I.18}$$

Un modèle ou réseau RBFN calcule une combinaison linéaire de fonctions radiales de centres

$$y(x) = \sum_{j=1}^N \lambda_j \phi_j(\|x - \mu_j\|, \sigma_j) \quad \text{I.19}$$

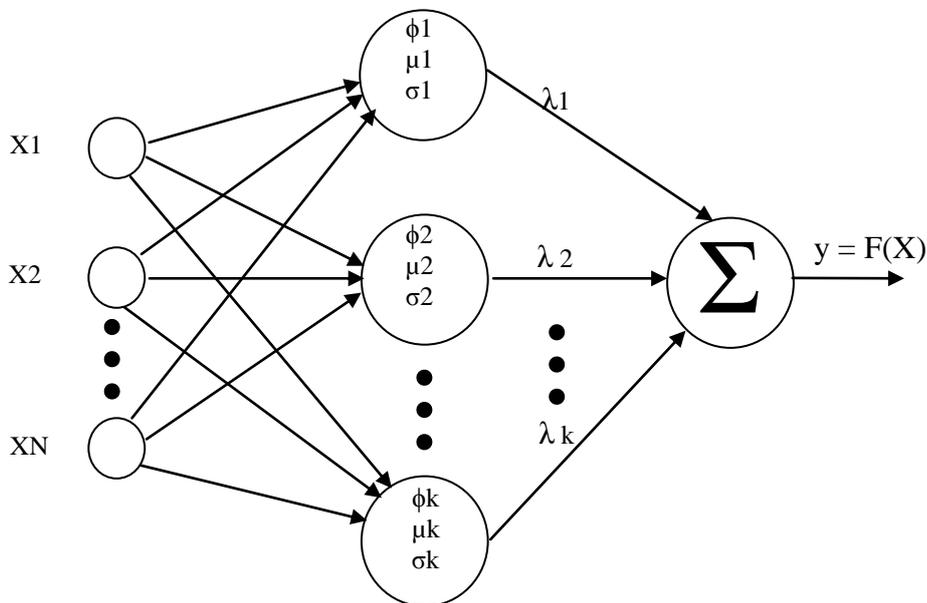


Figure: I.3 Représentation connexionniste d'un réseau RBFN.

A gauche la couche d'entrée x , au milieu les centres RBFN, à droite la sortie y . Pour traiter les problèmes de discrimination à C classes, on pourra utiliser C sorties y_1, \dots, y_c

On distingue trois couches (Figure I.3) : entrée x , fonctions radiales, sortie y , et on distingue aussi trois jeux de paramètres : les centres μ_j , les largeurs σ_j et les poids λ_j . Les combinaisons linéaires de Gaussiennes sont utilisées depuis les années 60 pour construire des interpolations ou approximations de fonctions [17]. A la fin des années 80, la présentation de ces modèles comme des réseaux connexionnistes a suscité un regain d'intérêt [18, 19] motivé en grande partie par la possibilité d'utiliser un algorithme d'apprentissage très rapide (sans

recourir à des techniques d'optimisation non linéaire comme dans le cas du Perceptron multicouche), donnant en général des résultats voisins des meilleurs modèles connexionnistes. L'apprentissage des modèles RBFN est supervisé, les RBFN sont utilisés pour résoudre tant des tâches de discrimination (en général en choisissant $i \in \{-1,1\}^C$) ou des tâches de régression ou prévision de signal (monovarié $y_i \in \mathfrak{R}$ ou multivarié $y_i \in \mathfrak{R}^C$).

I.2.5.2. Apprentissage des modèles RBFN

L'apprentissage d'un modèle RBFN consiste à déterminer son architecture (le nombre N de fonctions radiales) et à fixer les valeurs des paramètres.

Une caractéristique intéressante des modèles RBFN est que l'on peut diviser les paramètres en trois groupes : les centres μ , les largeurs σ et les poids λ . L'interprétation de chaque groupe permet de proposer un algorithme d'apprentissage séquentiel, simple et performant.

Dans un réseau de neurones artificiel RBFN il ya quatre paramètres principaux à régler.

- ❖ Le nombre de neurones N dans la couche cachée.
- ❖ La position des centres des Gaussiennes μ de chacun des neurones.
- ❖ La largeur des Gaussiennes σ .
- ❖ Les poids des connexions λ entre les neurones de la couche cachée et les neurones de la couche de sortie

La plupart des utilisateurs déterminent empiriquement la valeur de N en recourant à des techniques de validation croisée.

On estime les positions des centres μ_j à l'aide d'un algorithme non supervisé de type k-means.

Une fois ces paramètres fixés, il est possible de calculer les poids λ_j optimaux par une méthode de régression linéaire telle que :

$$y(x) = \sum_{j=1}^N \lambda_j \phi_j(\|x - c_j\|, \sigma_j) \quad \text{I.20}$$

I.2.6. Machine à vecteurs de support

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais Support Vector Machine, SVM) sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression. Les SVM sont une généralisation des classificateurs linéaires.

Les SVM ont été développés dans les années 1990 à partir des considérations théoriques de

Vladimir Vapnik sur le développement d'une théorie statistique de l'apprentissage (la Théorie de Vapnik-Chervonenkis) [20]. Les SVM ont rapidement été adoptés pour leur capacité à travailler avec des données de grandes dimensions, le faible nombre d'hyper paramètres, leurs garanties théoriques, et leurs bons résultats en pratique.

Les SVM ont été appliqués à de très nombreux domaines (bio-informatique, recherche d'information, vision par ordinateur, finance [21]...). Selon les données, la performance des machines à vecteurs de support est de même ordre, ou même supérieure, à celle d'un réseau de neurones ou d'un modèle de mixture Gaussienne.

I.2.6.1. Principe général

Cette technique est une méthode de classification à deux classes qui tente de séparer les exemples positifs des exemples négatifs dans l'ensemble des exemples. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Cela garantit une généralisation du principe car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être situés d'un côté ou l'autre de la frontière [22]. L'intérêt de cette méthode est la sélection de vecteurs supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas, ce qui peut être considéré comme un avantage pour cette méthode.

On se limite pour l'instant à un problème de discrimination à deux classes, c'est-à-dire $y \in \{1, -1\}$, le vecteur d'entrée x étant dans un espace X muni d'un produit scalaire. On peut prendre par exemple $X = \mathfrak{R}^N$.

I.2.6.2. Cas linéairement séparable

Le cas simple est le cas d'une fonction discriminante linéaire, obtenue par combinaison linéaire du vecteur d'entrée $x = (x_1, \dots, x_N)^T$, avec un vecteur de poids $w = (w_1, \dots, w_N)$

$$h(x) = W^T x + W_0 \tag{I.21}$$

Il est alors décidé que x est de classe 1 si $h(x) \geq 0$ et de classe -1 sinon. C'est un classificateur linéaire [23].

La frontière de décision $h(x) = 0$ est un hyperplan, appelé hyperplan séparateur, ou séparatrice. Le but d'un algorithme d'apprentissage supervisé est d'apprendre la fonction $h(x)$

par le biais d'un ensemble d'apprentissage : $\{(x_1, l_1), (x_2, l_2), \dots, (x_p, l_p)\} \subset \mathbb{R}^N \times \{-1, 1\}$

où les l_k sont les labels, p est la taille de l'ensemble d'apprentissage, N la dimension des vecteurs d'entrée. Si le problème est linéairement séparable, on doit alors avoir :

$$l_k h(x_k) \geq 0 \quad 1 \leq k \leq p \quad \text{autrement dit:} \quad l_k (W^T x_k + W_0) \geq 0 \quad 1 \leq k \leq p$$

I.2.6.3. Marge maximale

On se place désormais dans le cas où le problème est linéairement séparable. Même dans ce cas simple, le choix de l'hyperplan séparateur n'est pas évident. Il existe en effet une infinité d'hyperplans séparateurs, dont les performances en apprentissage sont identiques (le risque empirique est le même), mais dont les performances en généralisation peuvent être très différentes. Pour résoudre ce problème, il a été montré [22] qu'il existe un unique hyperplan optimal, défini comme l'hyperplan qui maximise la marge entre les échantillons et l'hyperplan séparateur.

Il existe des raisons théoriques à ce choix. Vapnik a montré [22] que la capacité des classes d'hyperplans séparateurs diminue lorsque leur marge augmente.

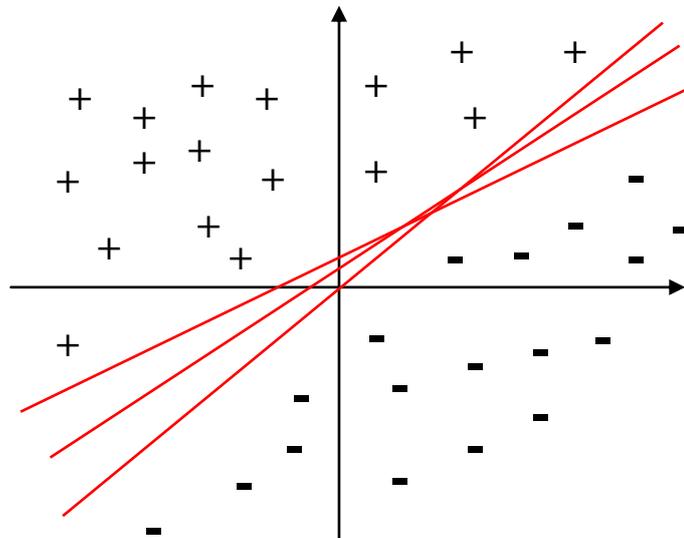


Figure I.4: Infinité d'hyperplans séparateurs pour un ensemble de points linéairement séparables.

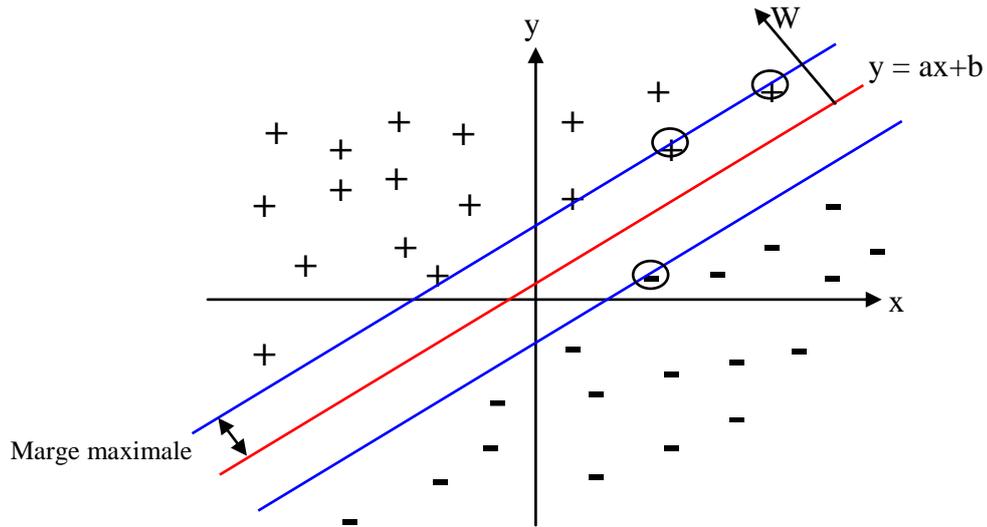


Figure I.5: L'hyperplan optimal (en rouge) avec la marge maximale

Les échantillons entourés sont des vecteurs supports. La marge est la distance entre l'hyperplan et les échantillons les plus proches. Ces derniers sont appelés vecteurs supports. L'hyperplan qui maximise la marge est donné par :

$$\arg \max_{w, w_0} \min_k \left\{ \|x - x_k\| : x \in \mathfrak{R}^N, w^T x + w_0 = 0 \right\} \quad \text{I.22}$$

Il s'agit donc de trouver w et w_0 remplissant ces conditions, afin de déterminer l'équation de l'hyperplan séparateur :

$$h(x) = W^T x + W_0 = 0 \quad \text{I.23}$$

I.2.6.4. Recherche de l'hyperplan optimal

La marge est la plus petite distance entre les échantillons d'apprentissage et l'hyperplan séparateur qui satisfait la condition de séparabilité (à savoir $l_k(w^T x_k + w_0) \geq 0$ comme expliqué précédemment). La distance d'un échantillon x_k à l'hyperplan est donnée par sa projection orthogonale sur l'hyperplan :

$$\frac{l_k(W^T x_k + W_0)}{\|W\|} \quad \text{I.24}$$

L'hyperplan séparateur (w, w_0) de marge maximale est donc donné par :

$$\arg \max_{w, w_0} \left\{ \frac{1}{\|W\|} \min_k [l_k(w^T x_k + w_0)] \right\} \quad \text{I.25}$$

Afin de faciliter l'optimisation, on choisit de normaliser w et w_0 , de telle manière que les

échantillons à la marge (x_{marge}^+ pour les vecteurs supports sur la frontière positive, et x_{marge}^- pour ceux situés sur la frontière opposée) satisfassent [21]:

$$\begin{cases} w^T x_{marge}^+ + w_0 = 1 \\ w^T x_{marge}^- + w_0 = -1 \end{cases} \quad \text{I.26}$$

$$\text{D'où pour tous les échantillons, } k = 1, \dots, p \quad l_k (w^T x_k + w_0) \geq 1 \quad \text{I.27}$$

Cette normalisation est parfois appelée la forme canonique de l'hyperplan, ou hyperplan canonique [21].

Avec cette mise à l'échelle, la marge vaut désormais $\frac{1}{\|w\|}$, il s'agit donc de maximiser $\|w\|^{-1}$.

La formulation dite primale des SVM s'exprime alors sous la forme suivante Minimiser $\frac{1}{2} \|w\|^2$

sous les contraintes $l_k (w^T x_k + w_0) \geq 1$.

Ceci peut se résoudre par la méthode classique des multiplicateurs de Lagrange, où le lagrangien est donné par :

$$L(w, w_0, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{k=1}^p \alpha_k \{l_k (w^T x_k + w_0) - 1\} \quad \text{I.28}$$

Le lagrangien doit être minimisé par rapport à w et w_0 , et maximisé par rapport à α .

En annulant les dérivées partielles du lagrangien, selon les conditions de Kuhn-Tucker, on obtient :

$$\begin{cases} \sum_{k=1}^p \alpha_k l_k x_k = w^* \\ \sum_{k=1}^p \alpha_k l_k = 0 \end{cases} \quad \text{I.29}$$

En réinjectant ces valeurs dans l'équation I.29, on obtient la formulation duale :

$$\text{Maximiser } \tilde{L}(\alpha) = \sum_{k=1}^p \alpha_k - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j l_i l_j x_i^T x_j \quad \text{I.30}$$

sous les contraintes $\alpha_k > 0$, et $\sum_{k=1}^p \alpha_k l_k = 0$.

Ce qui donne les multiplicateurs de Lagrange optimaux α_k^* .

Afin d'obtenir l'hyperplan solution, on remplace w par sa valeur optimale w^* , dans l'équation de l'hyperplan $h(x)$, ce qui donne:

$$h(x) = \sum_{k=1}^p \alpha_k^* l_k (x \cdot x_k) + w_0 \quad \text{I.31}$$

Il y a trois remarques intéressantes à faire à propos de ce résultat:

- ❖ La première découle de l'une des conditions de Kuhn-Tucker, qui donne :

$$\alpha_k [l_k h(x_k) - 1] = 0 \quad 1 \leq k \leq p \quad \text{I.32}$$

$$\text{D'où} \begin{cases} \alpha_k = 0 \\ l_k h(x_k) = 1 \end{cases}$$

Les seuls points pour lesquels les contraintes du lagrangien sont actives sont donc les points tels que $l_k h(x_k) = 1$, qui sont les points situés sur les hyperplans de marges maximales. En d'autres termes, seuls les vecteurs supports participent à la définition de l'hyperplan optimal.

- ❖ La deuxième remarque découle de la première. Seul un sous-ensemble restreint de points est nécessaire pour le calcul de la solution, les autres échantillons ne participant pas du tout à sa définition. Ceci est donc efficace au niveau de la complexité. D'autre part, le changement ou l'agrandissement de l'ensemble d'apprentissage à moins d'influence que dans un modèle de mélanges gaussiens par exemple, où tous les points participent à la solution. En particulier, le fait d'ajouter des échantillons à l'ensemble d'apprentissage qui ne sont pas des vecteurs supports n'a aucune influence sur la solution finale.

- ❖ La dernière remarque est que l'hyperplan solution ne dépend que du produit scalaire entre le vecteur d'entrée et les vecteurs supports. Cette remarque est l'origine de la deuxième innovation majeure des SVM : le passage par un espace de redescription grâce à une fonction noyau.

I.2.6.5. Cas non séparable

La notion de marge maximale et la procédure de recherche de l'hyperplan séparateur telles que présentées pour l'instant ne permettent de résoudre que des problèmes de discrimination linéairement séparables. C'est une limitation sévère qui condamne à ne pouvoir résoudre que des problèmes jouets, ou très particuliers. Afin de remédier au problème de l'absence de séparateur linéaire, l'idée des SVM est de reconsidérer le problème dans un espace de dimension supérieure, éventuellement de dimension infinie. Dans ce nouvel espace, il est alors probable qu'il existe un séparateur linéaire.

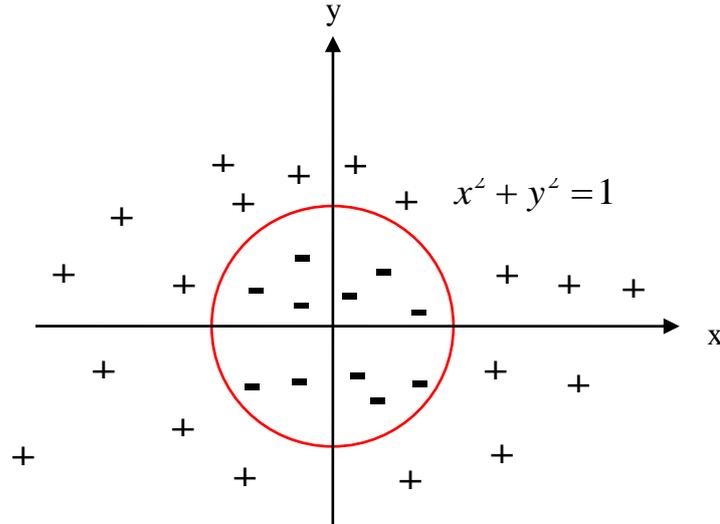


Figure I.6: Exemple simple de transformation

Le problème n'est pas linéairement séparable en coordonnées cartésiennes, par contre en coordonnées polaires, le problème devient linéaire. Il s'agit ici d'un exemple très simple, l'espace de redescription étant de même dimension que l'espace d'entrée.

Plus formellement, on applique aux vecteurs d'entrée x une transformation non-linéaire ϕ . L'espace d'arrivée $\phi(x)$ est appelé espace de redescription. Dans cet espace, on cherche alors l'hyperplan $h(x)$ qui vérifie $l_k h(x_k) > 0$ Pour tous les points x_k de l'ensemble d'apprentissage, c'est-à-dire l'hyperplan séparateur dans l'espace de redescription. $h(x)$ est donné par la relation suivante:

$$h(x) = w^T \phi(x) + w_0 \quad \text{I.33}$$

En utilisant la même procédure que dans le cas sans transformation, on aboutit au problème d'optimisation suivant :

$$\text{Maximiser} \quad \tilde{L}(\alpha) = \sum_{k=1}^p \alpha_k - \frac{1}{2} \sum_{ij} \alpha_i \alpha_j l_i l_j \phi(x_i)^T \phi(x_j) \quad \text{I.34}$$

sous les contraintes $\alpha_k > 0$, et $\sum_{k=1}^p \alpha_k l_k = 0$

Le problème de cette formulation est qu'elle implique un produit scalaire entre vecteurs dans l'espace de redescription [24], de dimension élevée, ce qui est coûteux en termes de calculs. Pour résoudre ce problème, on utilise une astuce connue sous le nom de Kernel trick, qui consiste à utiliser une fonction noyau, qui vérifie [25]:

$$K(x_i, x_j) = \phi(x_i)^T \cdot \phi(x_j) \quad \text{I.35}$$

d'où l'expression de l'hyperplan séparateur en fonction de la fonction noyau :

$$h(x) = \sum_{k=1}^p \alpha_k^* l_k K(x_k, x) + w_0 \quad \text{I.36}$$

L'intérêt de la fonction noyau est double :

- Le calcul se fait dans l'espace d'origine, ceci est beaucoup moins coûteux qu'un produit scalaire en grande dimension.
- La transformation ϕ n'a pas besoin d'être connue explicitement, seule la fonction noyau intervient dans les calculs. On peut donc envisager des transformations complexes, et même des espaces de redescription de dimension infinie.

I.2.6.6. Choix de la fonction noyau

En pratique, on ne connaît pas la transformation ϕ , on construit plutôt directement une fonction noyau. Celle-ci doit respecter certaines conditions, elle doit correspondre à un produit scalaire dans un espace de grande dimension. Le théorème de Mercer explicite les conditions que K doit satisfaire pour être une fonction noyau : elle doit être symétrique, semi-définie positive [26].

L'exemple le plus simple de fonction noyau est le noyau linéaire :

$$K(x_i, x_j) = x_i^T \cdot x_j \quad \text{I.37}$$

On se ramène donc au cas d'un classificateur linéaire, sans changement d'espace. L'approche par Kernel trick généralise ainsi l'approche linéaire. Le noyau linéaire est parfois employé pour évaluer la difficulté d'un problème.

Des noyaux usuels employés avec les SVM sont [26]:

- le noyau polynomial $K(x_i, x_j) = (x_i^T \cdot x_j + 1)^d$
- le noyau Gaussien $K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$

I.2.6.7. Marge souple

En général, il n'est pas non plus possible de trouver une séparatrice linéaire dans l'espace de redescription. Il se peut aussi que des échantillons soient mal étiquetés, et que l'hyperplan séparateur ne soit pas la meilleure solution au problème de classement.

En 1995, Corinna Cortes et Vladimir Vapnik proposent une technique dite de marge souple [27], qui tolère les mauvais classements. La technique cherche un hyperplan séparateur qui minimise le nombre d'erreurs grâce à l'introduction de variables ressort ζ_k (*slack variables* en anglais), qui permettent de relâcher les contraintes sur les vecteurs d'apprentissage :

$$l_k(w^T x_k + w_0) \geq 1 - \zeta_k \quad \zeta_k \geq 0 \text{ et } 1 \leq k \leq p \quad \text{I.38}$$

Avec les contraintes précédentes, le problème d'optimisation est modifié par un terme de pénalité, qui pénalise les variables ressort dont les variables sont trop grandes:

$$\text{Minimiser} \quad \frac{1}{2} \|w\|^2 + C \sum_{k=1}^p \zeta_k \quad C > 0 \quad \text{I.39}$$

Où "C" est une constante qui permet de contrôler le compromis entre nombre d'erreurs de classement, et la largeur de la marge. Elle doit être choisie par l'utilisateur, en général par une recherche exhaustive dans l'espace des paramètres, en utilisant par exemple la validation croisée sur l'ensemble d'apprentissage. Le choix automatique de ce paramètre de régularisation est un problème statistique majeur.

I.2.6.8. Optimisation des paramètres:

Le paramètre principal à optimiser dans l'algorithme des SVM est le paramètre de régularisation "C". Il exprime le compromis entre deux objectifs conflictuels : la minimisation du nombre d'échantillons mal prédits lors de la phase d'entraînement et la maximisation de la marge. Lorsque "C" devient grand, seul le nombre d'échantillons mal classés est considéré. Au contraire, lorsque "C" tend vers 0, la marge est maximisée sans tenir compte du nombre d'échantillons mal classés, ce qui peut conduire à des solutions aberrantes. On cherche donc une valeur intermédiaire en faisant varier "C". Le choix s'effectue sur la base des performances des modèles obtenus en validation croisée, par exemple.

Les paramètres définissant le kernel sont également à optimiser. Par exemple, si on utilise des fonctions radiales de base, il faut déterminer σ qui correspond à la largeur de la Gaussienne.

I.3. Classification multiclasse:

Le modèle SVM dans leur origine biclasses. Cependant, les problèmes du monde réel sont dans la plupart des cas multiclasse, Dans ces cas, on ne cherche pas à affecter un nouvel exemple à l'une de deux classes mais à l'une parmi plusieurs, c.-à-d. que la décision n'est plus biclasses et un seul hyperplan ne suffit plus.

Le modèle SVM multiclasse, réduisent le problème multiclasse à une composition de plusieurs hyperplans biclasses permettant de tracer les frontières de décision entre les différentes classes [28, 29]. Ces méthodes décomposent l'ensemble d'exemples en plusieurs sous-ensembles représentant chacun un problème de classification biclasses. Pour chaque problème un hyperplan de séparation est déterminé par la méthode SVM biclasses. On

construit lors de la classification une hiérarchie des hyperplans biclasses qui est parcourue de la racine jusqu'à une feuille pour décider de la classe d'un nouvel exemple. On trouve dans la littérature plusieurs méthodes de décomposition:

I.3.1. Un contre tous:

C'est la méthode la plus simple et la plus ancienne. Selon la formulation de Vapnik [30], elle consiste à déterminer pour chaque classe k un hyperplan $H_k(w_k, b_k)$ la séparant de toutes les autres classes. Cette classe k est considérée comme étant la classe positive (+1) et les autres classes comme étant la classe négative (-1), ce qui résulte, pour un problème à K classes. Un hyperplan H_k est défini pour chaque classe k par la fonction de décision suivante :

$$H_k(x) = \text{signe}(\langle w_k, x \rangle + b_k) = \begin{cases} +1 \rightarrow f_k(x) > 0 \\ 0 \rightarrow \text{SINON} \end{cases} \quad \text{I.40}$$

La valeur retournée de l'hyperplan permet de savoir si x appartient à la classe k ou non. Dans le cas où il n'appartient pas à k ($H_k(x) = 0$), nous n'avons aucune information sur l'appartenance de x aux autres classes. Pour le savoir, on présente x à tous les hyperplans, ce qui donne la fonction de décision de l'équation (I.42) suivante:

$$k^* = \arg_{1 \leq k \leq K} \text{Max}(H_k(x)) \quad \text{I.41}$$

Si une seule valeur $H_k(x)$ est égale à 1 et toutes les autres sont égales à 0, on conclut que x appartient à la classe k . Le problème est que l'équation (I.42) peut être vérifiée pour plus d'une classe, ce qui produit des régions d'ambiguïté, et l'exemple x est dit non classifiable. La Figure I.7 représente un cas de séparation de 3 classes.

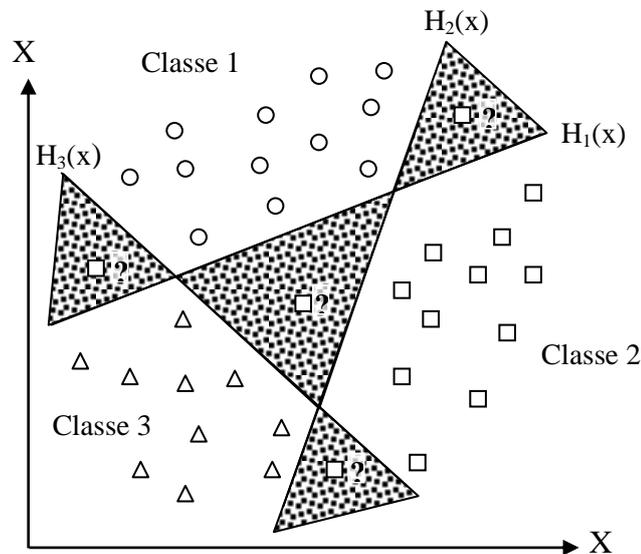


Figure I.7 – Approche un-contre-tous avec des zones d'indécision.

Pour surmonter cette situation, la méthode un-contre-tous utilise le principe de "le gagnant prend tout" ("winner-takes-all") : la classe k retenue est celle qui maximise $f_k(x) = \langle w_k, x \rangle + b_k$ de l'équation (I.43).

$$k^* = \arg_{1 \leq k \leq K} \text{Max}(\langle w_k, x \rangle + b_k) \quad \text{I.42}$$

Géométriquement interprétée, tout nouvel exemple x est affecté à la classe dont l'hyperplan est le plus loin de x , parmi les classes ayant $H(x) = 1$ (Figure I.8).

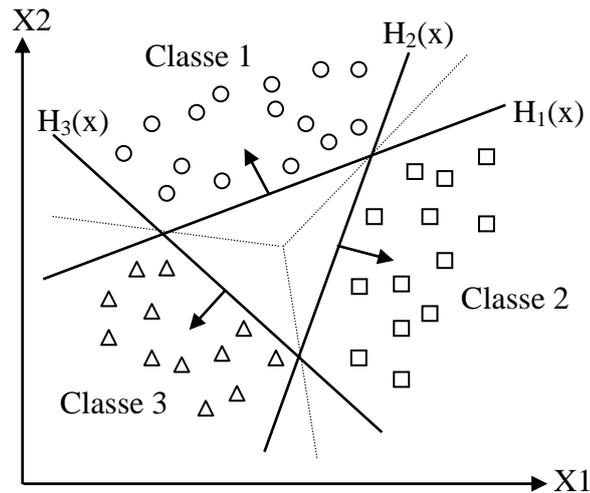


Figure I.8 – Résolution des cas d'indécision dans la méthode 1vsR

La méthode un-contre-tous peut être utilisée pour découvrir même les cas de rejet où un exemple n'appartient à aucune des K classes. Pour cela, on prend les deux fonctions de décision les plus élevées, puis on calcule leur différence, si elle est au-dessous d'un certain seuil, l'exemple est rejeté.

Souvent, la méthode un-contre-tous est critiquée à cause de son asymétrie [21], puisque chaque hyperplan est entraîné sur un nombre d'exemples négatifs beaucoup plus important que le nombre d'exemples positifs.

La méthode un-contre-un suivante est une méthode symétrique qui corrige ce problème.

I.3.2. Un-contre-un (1vs1):

Cette méthode, appelée aussi "pairwise", revient à Kner et ses co-auteurs [31] qui l'ont proposée pour les réseaux de neurones. Elle consiste à utiliser un classificateur pour chaque paire de classes. Au lieu d'apprendre K fonctions de décisions, la méthode 1vs1 discrimine chaque classe de chaque autre classe, ainsi $K(K-1)/2$ fonctions de décisions sont apprises.

Pour chaque paire de classes (k, s) , la méthode un contre un définit une fonction de décision biclasses $h_{ks} : R \rightarrow \{-1, +1\}$. L'affectation d'un nouvel exemple se fait par liste de vote. On teste un exemple par le calcul de sa fonction de décision pour chaque hyperplan. Pour chaque test, on vote pour la classe à laquelle appartient l'exemple (classe gagnante). On définit la fonction de décision biclasses $H_{ks}(x)$ de l'équation I.44.

$$H_{ks}(x) = \text{signe}(f_{ks}(x)) = \begin{cases} +1 \rightarrow f_{ks}(x) > 0 \\ 0 \rightarrow \text{SINON} \end{cases} \quad \text{I.43}$$

Sur la base des $K(k-1)/2$ fonctions de décision biclasses, on définit K autres fonctions de décision (équation I.45) :

$$H_k(x) = \sum_{s=1}^m H_{ks}(x) \quad \text{I.44}$$

Un nouvel exemple est affecté à la classe la plus votée. La règle de classification d'un nouvel exemple x est donnée par l'équation I.46 :

$$k^* = \arg_{1 \leq k \leq K} (\text{Max} H_k(x)) \quad \text{I.45}$$

Malheureusement, la fonction I.46 peut être vérifiée pour plusieurs classes, ce qui produit des zones d'indécisions. La méthode de vote affecte dans ce cas, un exemple aléatoirement à l'une des classes les plus votées.

La Figure I.9 représente un exemple de classification de trois classes avec la zone d'indécision.

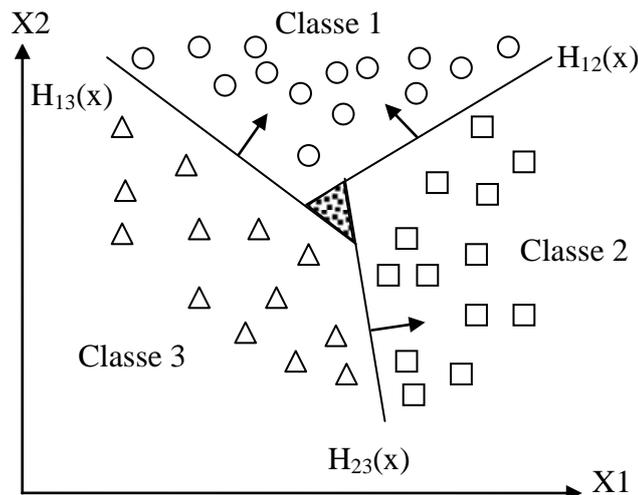


Figure I.9 – Approche un-contre-un

Bien que La méthode un contre un utilise, pour l'entraînement, un nombre plus important d'hyperplans que la méthode un contre tous, elle est souvent plus rapide. Cela est du, d'une

part, au nombre limité d'exemples utilisés pour entraîner chaque hyperplan, et d'autre part, à la simplicité des problèmes à résoudre. En effet, chaque deux classes prises à part sont moins chevauchées que toutes les classes.

I.4. Technique de validation:

I.4.1. Validation simple:

Une première technique de validation consiste tout simplement à diviser les données disponibles en deux ensembles (les ensembles d'apprentissage et de validation), sans qu'une donnée ne soit commune [32]. Un nombre non négligeable de données est bien entendu nécessaire dans l'ensemble de validation pour estimer correctement l'erreur de généralisation, ce qui réduit d'autant le nombre de données disponibles pour l'apprentissage. Souvent on garde 2/3 des données dans l'ensemble d'apprentissage pour en réserver 1/3 pour la validation. Les ensembles d'apprentissage et de validation sont tirés aléatoirement parmi les N données disponibles.

I.4.2. Monte-Carlo validation Croisé:

L'inconvénient majeur de la technique de validation simple est que ses résultats dépendent intimement du tirage aléatoire des données, entre les deux ensembles d'apprentissage et de validation [32]. Un remède possible consiste alors à effectuer plusieurs tirages de ce type, à estimer l'erreur de généralisation pour chacun de ces tirages, et à retenir la moyenne de ces estimations.

I.4.3. K-fold cross-validation:

Si l'augmentation du nombre de tirages (et donc du nombre d'évaluations dans la méthode de Monte-Carlo cross-validation) permet d'utiliser successivement les mêmes données, tantôt pour l'apprentissage, tantôt pour la validation, leur caractère aléatoire empêche toute certitude quant au fait que chaque donnée soit, au moins approximativement, utilisée le même nombre de fois pour l'apprentissage ou pour la validation. Seule une valeur prohibitive de J permettrait d'atteindre cette certitude.

Une variante consiste alors à découper l'ensemble des N données disponibles en K sous-ensembles disjoints de taille approximativement identiques. On peut alors obtenir K évaluations, en utilisant à chaque fois $K-1$ sous-ensembles pour l'apprentissage et un sous-ensemble pour la validation. Chacun des K sous-ensemble joue alors une et une seule fois le

rôle d'ensemble de validation. L'estimation de l'erreur de généralisation est alors calculée par la moyenne des erreurs de généralisation de K évaluations [32].

I.5. Conclusion:

Dans ce chapitre, nous avons mis en évidence les concepts de bases concernant les méthodes linéaires et non linéaires. Certaines approches sont basées sur des modèles mathématiques qui trouvent la relation entre les mesures acquises et la prédiction de la classe. D'autres approches sont basées sur des techniques d'interpolation appliqué sur l'ensemble d'apprentissage de la base de données pour trouver le meilleur modèle de prédiction des classes des échantillons de test.

Chapitre II

Description des bases de données

II.1. Introduction

L'évaluation de la qualité nutritionnelle des aliments, implique un grand nombre de déterminations analytiques de différents constituants. Les méthodes d'analyse "classiques", souvent utilisées, sont généralement longues à mettre en œuvre et assez complexes et demande plusieurs heures de travail et d'un personnel qualifié.

Au cours des dernières années, la spectrométrie dans le proche infrarouge (en anglais Near infrared NIR) s'est implantée dans de nombreux laboratoires d'analyses de routine. Cette technique d'analyse a l'avantage d'être très rapide, très reproductible, utilisant une faible quantité d'échantillon sans préparation particulière. Toutefois, sa mise en œuvre présente certaines difficultés et impose certaines règles d'utilisation. Les quelques inconvénients rencontrés ne diminuent en rien les nombreux avantages que l'utilisation de cette technique apporte. L'information spectrale obtenue est importante et permet diverses applications [33].

II.2. La spectroscopie

Les méthodes spectroscopiques sont largement utilisées pour l'analyse qualitative et quantitative de composés de produits agroalimentaires [2, 34]. En général, quand les signaux sont bien définis et non superposés, on peut à partir de l'intensité du signal, calculer une concentration. Cependant, lors de l'analyse de matrices complexes, les signaux ne sont pas toujours bien séparés. Il n'est donc pas évident d'identifier et de quantifier les différents composants. Afin de rester dans le premier objectif de l'analyse spectrale, qui est d'obtenir un résultat d'analyse rapide sur différents paramètres, il est alors nécessaire d'utiliser des méthodes d'extraction d'information rapide.

Les techniques spectroscopiques peuvent être considérées comme des réponses d'absorption d'énergie en fonction de la fréquence du rayonnement concerné [34, 35] (Figure II.1).

La spectroscopie infrarouge permet une acquisition rapide du signal. En effet, l'échantillon nécessite très peu ou pas de préparation, et peut généralement être placé sur un porte-échantillon ou être mesuré en ligne. De plus, l'instrumentation est multi-analytique.

Ainsi, plusieurs constituants peuvent être déterminés à partir d'un seul spectre. Ensuite, ces techniques n'utilisent pas ou peu de solvant par rapport aux techniques d'analyse classique, qui peuvent poser des problèmes pour l'environnement. Un échantillon va absorber partiellement et sélectivement le rayonnement auquel il est soumis. Cette absorption est caractéristique des propriétés chimiques et physico-chimiques du produit considéré.

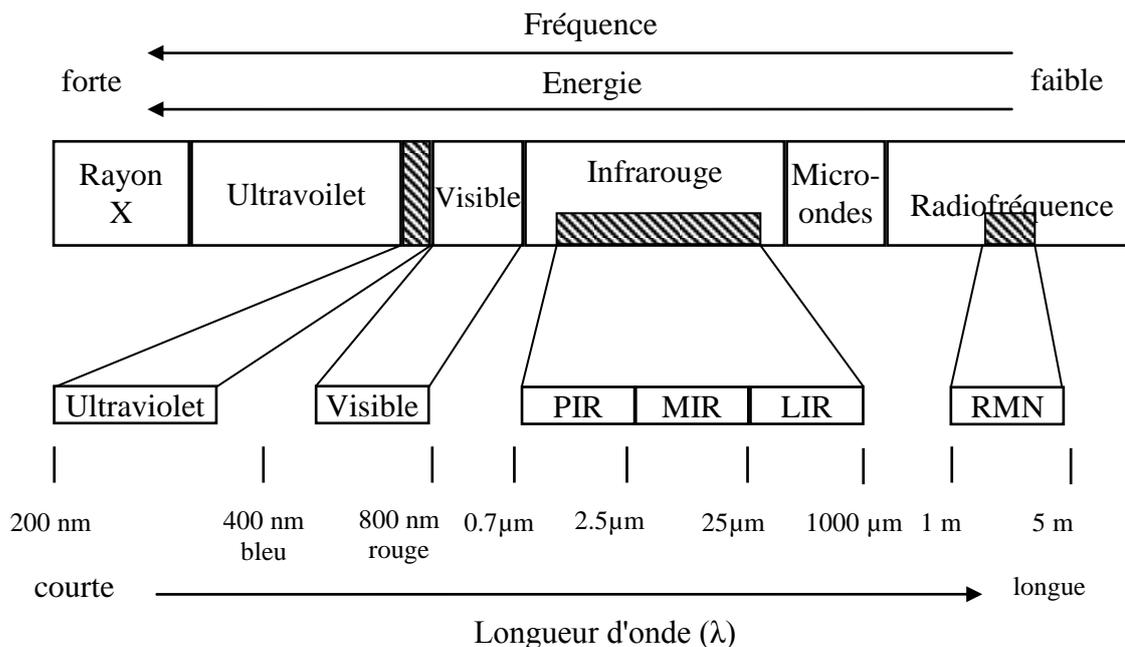


Figure II.1 : Région du spectre électromagnétique montrant la relation entre les longueurs d'onde, l'énergie, les fréquences et les techniques spectroscopiques

II.2.1. Moyen infrarouge (MIR)

En spectroscopie infrarouge, on s'intéresse aux différentes vibrations des liaisons. La spectroscopie MIR est caractérisée par une résolution fine des pics qui lui confère plusieurs avantages. Tout d'abord, la finesse des raies permet d'identifier facilement le type de liaison rencontré et même de quantifier un composé en fonction de l'intensité des signaux. Un autre avantage important des spectres MIR est qu'ils apportent des informations sur l'état physico-chimique de différents composants comme la cristallinité, l'oxydation, ou encore la phase, qui peuvent être utiles pour étudier les changements causés par le traitement ou le stockage [34]. Les applications MIR en agroalimentaire se développent, suite à l'apparition d'appareils basés sur la transformée de Fourier et de nouveaux dispositifs de présentation des échantillons plus pratiques [2]. Plusieurs problèmes d'authentification ont été étudiés par ce type de méthode ; par exemple la quantification des adultérants de l'huile d'olive [36], ou encore la discrimination des différents types de fruits dans les purées [37] ou les confitures [38], et enfin la discrimination des différentes variétés de café [39, 40]. Au niveau du contrôle qualité de la production, une étude polonaise [41] s'est intéressée à l'évolution de la teneur en phospholipides lors de la production d'huile de colza.

II.2.2. Proche infrarouge (PIR) :

En spectroscopie PIR, les raies sont plus larges et donc la résolution des pics est moins bonne. Il n'est donc plus possible de quantifier un composé en fonction de l'intensité d'un pic et l'on fait souvent appel à la chimiométrie pour développer des modèles plus complexes [2, 42]. Cette technique étant moins sensible, elle est plus robuste et moins affectée par les problèmes d'humidité que la spectroscopie MIR. De nombreuses applications ont été développées pour l'analyse des produits agroalimentaires et agricoles. Par exemple, les différentes variétés de blé peuvent être discriminées [43, 44] et la détermination de plusieurs paramètres permet même d'en estimer la valeur. De même, les huiles de différentes espèces végétales ont pu être séparées par analyse en composantes principales [45]. Enfin, les techniques PIR permettent d'étudier l'authenticité des produits comme par exemple l'authenticité du jus d'orange [46], ou du ginseng [47], ou encore la pureté de la matière grasse dans le chocolat [48].

II.2.2.1. Performances de la spectroscopie infrarouge:

Elle est **rapide** : quelques minutes suffisent pour recueillir le spectre d'absorption d'un échantillon, qui servira à la prédiction de sa composition, ceci comparé avec la très longue durée d'une analyse de composition chimique (plusieurs jours) !

Elle est **non destructive** : l'échantillon est récupéré intact après analyse. Cette propriété est particulièrement importante pour des échantillons très peu abondants ou que l'on doit conserver pour des analyses ultérieures.

II.2.2.2. Principe

On appelle « infrarouge » le rayonnement correspondant aux longueurs d'onde directement supérieures à celles du spectre de la lumière visible. Conventionnellement les limites du proche infrarouge se situent entre 800 et 2500 nm.

L'absorption des rayonnements par les échantillons dépend de la composition de la matière organique. En effet les liaisons chimiques sont capables d'absorber les rayonnements correspondant à certaines fréquences particulières. C'est cette propriété des liaisons chimiques de la matière organique que va utiliser la spectroscopie dans le proche infrarouge pour établir un lien entre l'absorption de la lumière et la composition de l'échantillon.

Pour faire cette analyse, l'échantillon va être éclairé à différentes fréquences (ou longueurs d'onde). L'absorption de la lumière à chacune de ces longueurs d'onde constitue le «spectre»

de l'échantillon. Ce spectre peut être constitué de plusieurs centaines de longueurs d'onde pour chacune desquelles on a mesuré l'absorption de la lumière.

Cette richesse d'information constitue l'avantage et la difficulté de l'analyse NIR : beaucoup d'informations sont présentes dans un spectre, mais elles sont complètement emmêlées ! Pour surmonter cette difficulté il faut faire appel à des méthodes statistiques complexes, qui vont permettre de relier les spectres et les analyses chimiques

II.2.2.3. Images multispectrales:

Une image multispectrale est une image qui regroupe plusieurs plans d'images appelés bandes spectrales.

Une image multispectrale ayant un nombre de D bandes est illustrée en Figure II.2. Les composantes du vecteur X sont les valeurs du niveau de gris des pixels qui se trouvent dans la même position dans les différentes bandes. Le vecteur X est de dimension D , et le nombre de ces vecteurs est égal à la résolution de la bande. Par exemple, une image modélisée dans l'espace RVB (Rouge Vert Bleu) est considérée comme une image multispectrale à trois bandes R, V et B.

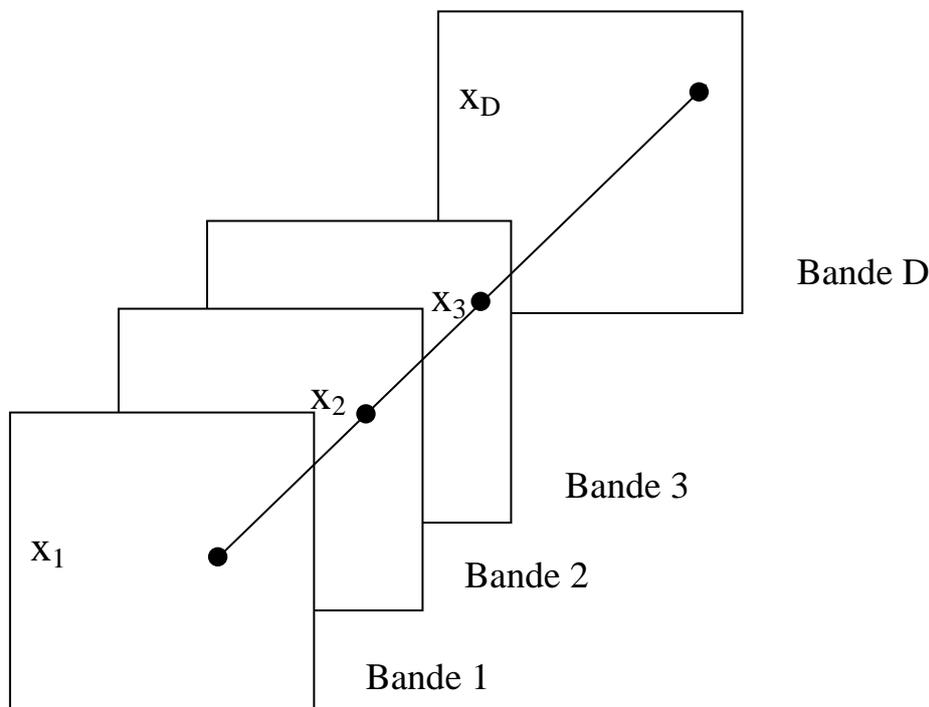


Figure II.2 Illustration d'une image multispectrale.

II.3. Description des bases de données réelles:

La validation de toutes les méthodes utilisées dans ce travail est réalisée sur quatre bases de données réelles liées à l'industrie agroalimentaire. La première base de données est celle d'huile d'olive [49], la deuxième et la troisième bases de données, sont les base de données des images multispectrales représentant des fruits et des légumes (Fraise et poivron) [50, 51], et la quatrième base de données est constituée de sept espèces de poissons capturés du même lac in Finlande [52].

II.3.1. Base de données huile d'olive:

Dans la base de données huile d'olive le problème consiste à discriminer les échantillons de l'huile d'olive de celle non huile d'olive, cette base de données consiste de 120 spectres de huile mesurés en proche infrarouge avec 4001 variables spectrales. Parmi les échantillons analysés, 71 (59 %) échantillons sont de divers types et qualités de huile d'olive et 44 (37 %) échantillons non-huile d'olive et 5 (4 %) échantillons non-huile d'olive mélangé avec huile d'olive.

Les spectres la collection des spectres de la base de données huile d'olive est représentées dans la Figure II.3, les figures II.4 et II.5 représentent les spectres de l'ensemble d'apprentissage et l'ensemble de test respectivement, la Figure II.6 présente la répartition des trois classes de la base de données huile d'olive dans l'espace de l'ACP.

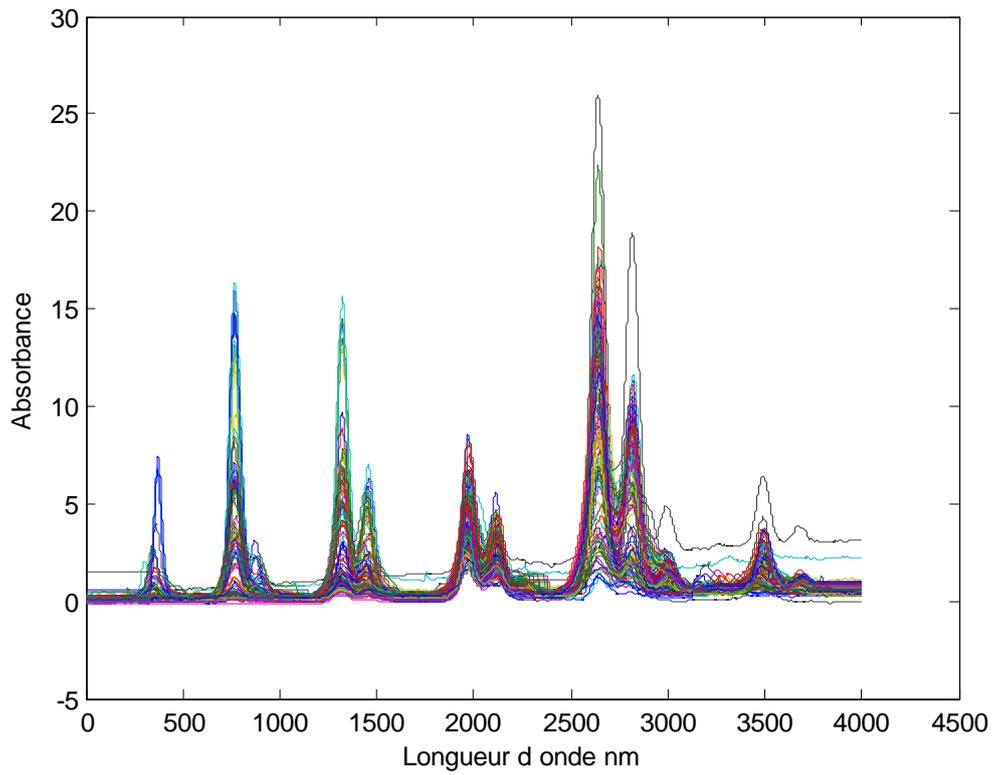


Figure II.3: La collection des spectres de la base de données Huile d'olive

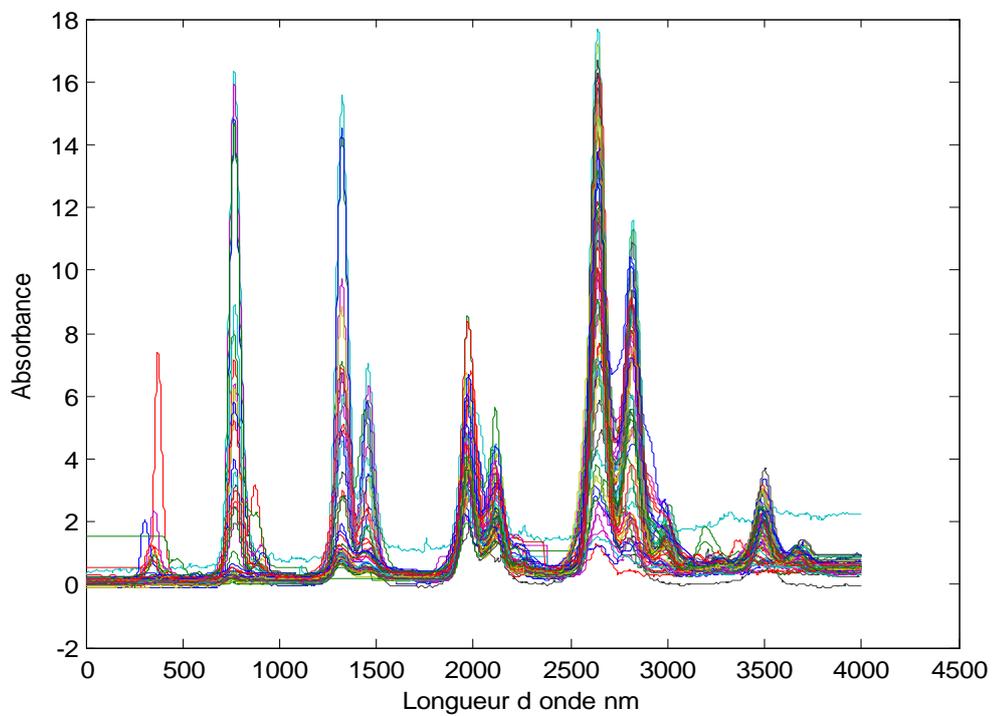


Figure II.4: Les spectres de l'ensemble d'apprentissage de la base de données Huile d'olive

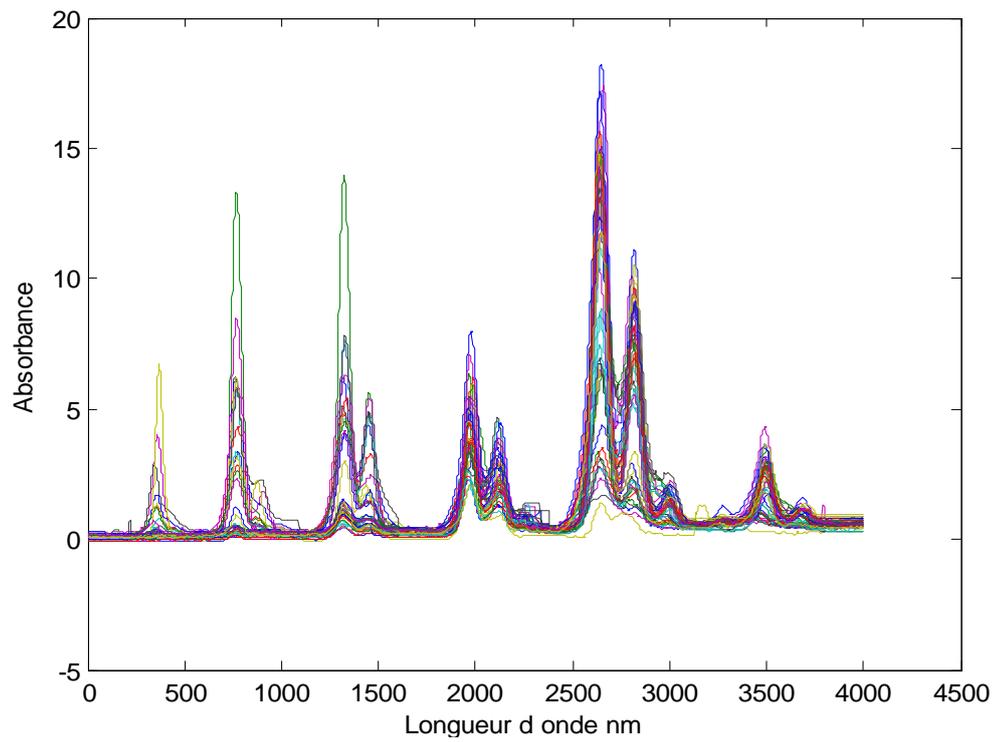


Figure II.5: spectres de l'ensemble de test de la base de données Huile d'olive

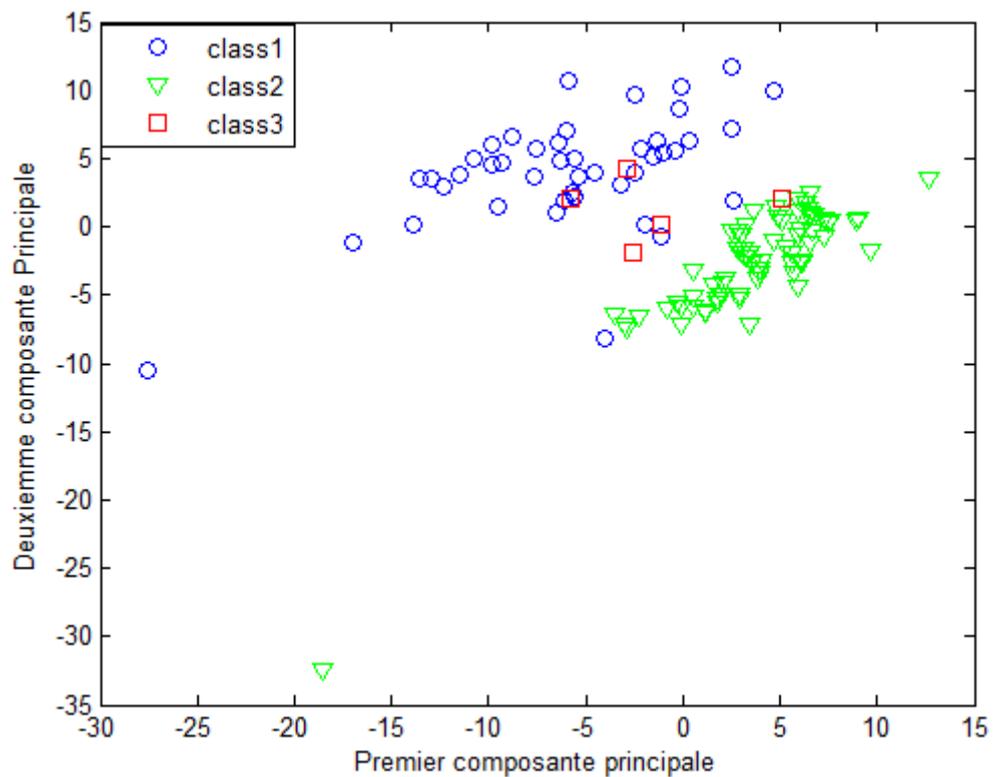


Figure II.6: Répartition des 3 classes pour la base de données huile d'olive

II.3.2. Images multispectrales (bases de données Fraise et Poivrons):

Les images multispectrales utilisés dans cette étude se réfèrent aux fruits et légumes authentique et contrefait. Le problème consiste à la discrimination entre les produits contrefaits et authentiques.

Chaque image multispectrale d'un ensemble de fraises ou de poivron a pour dimension 512x512 pixels, et chaque pixel est caractérisé par 31 canaux spectraux correspondant à des longueurs d'onde allant de 400 à 700 nm avec un intervalle de 10 nm, ce qui donne un total de taille réelle $[512 \times 512 \times 31]$ par image. La version RVB (Rouge, Vert et Bleu) correspondante (3 canaux spectraux par pixel) des deux séries d'images disponibles est présenté dans la figure II.7. Pour les deux séries d'images (Fraise et Poivron), on sélectionne les pixels utilisés dans cette base de données comme montre la figure II.8, le nombre de pixels sélectionnés pour la base de données de Fraise est 19212 pixels et pour la base de données de Poivron est 54541 pixels. A partir des pixels obtenus de cette manière, un sous-ensemble de 100 pixels sélectionnés de manière aléatoire est utilisé comme un ensemble d'apprentissage, tandis que l'ensemble de test sera déterminé par les pixels restants.

Les figures II.9 et II.10 présentent les spectres de l'ensemble d'apprentissage et la répartition des deux classes de la base de données Fraise dans l'espace de l'ACP respectivement, et les figures II.11 et II.12 présentent les spectres de l'ensemble d'apprentissage et la répartition des deux classes de la base de données Poivron dans l'espace de l'ACP respectivement.

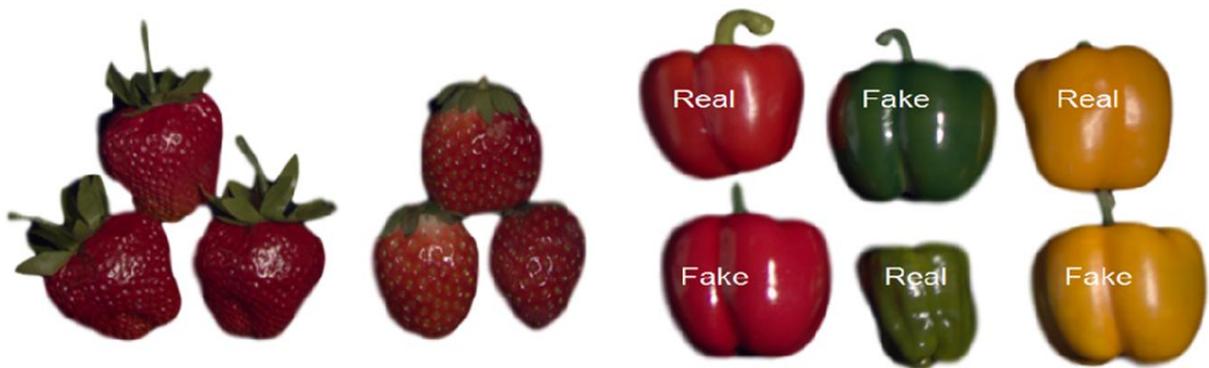


Figure II.7: Image RVB du Fraise et Poivron

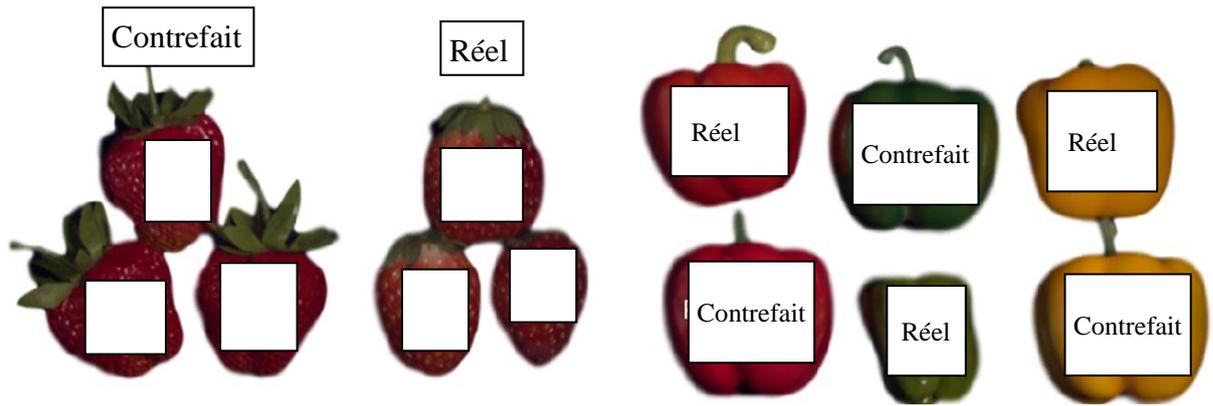


Figure II.8: Image RVB du Fraise et Poivron montre les pixels utilisés

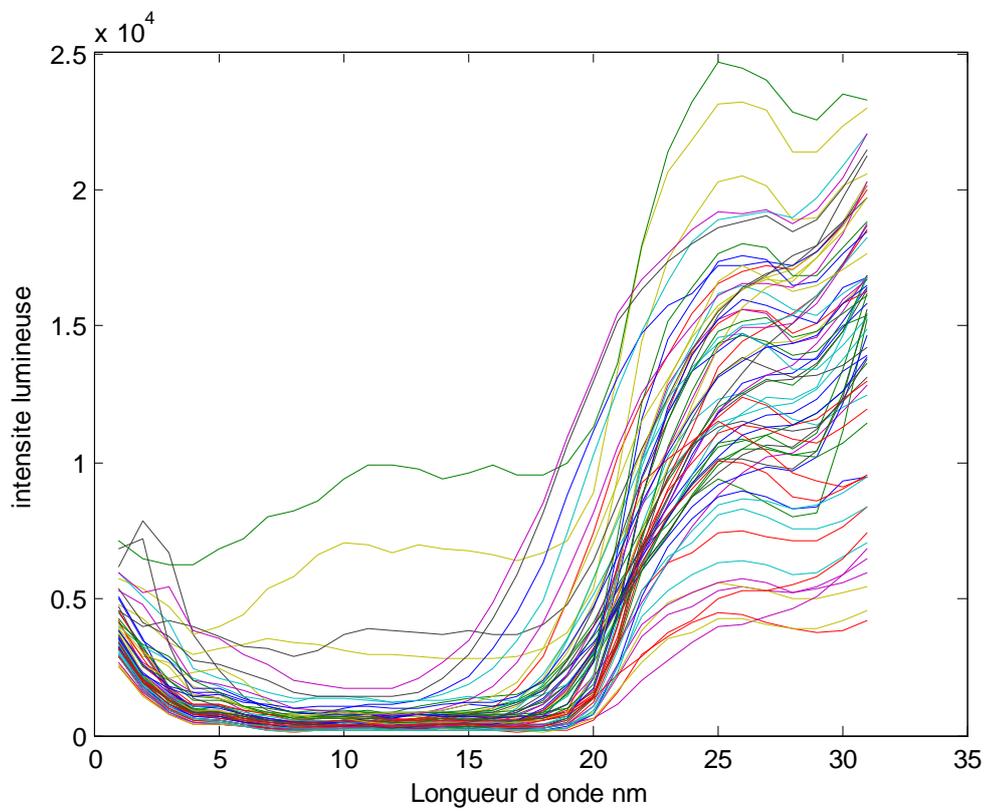


Figure II.9: Les spectres de l'ensemble d'apprentissage de la base de données Fraise

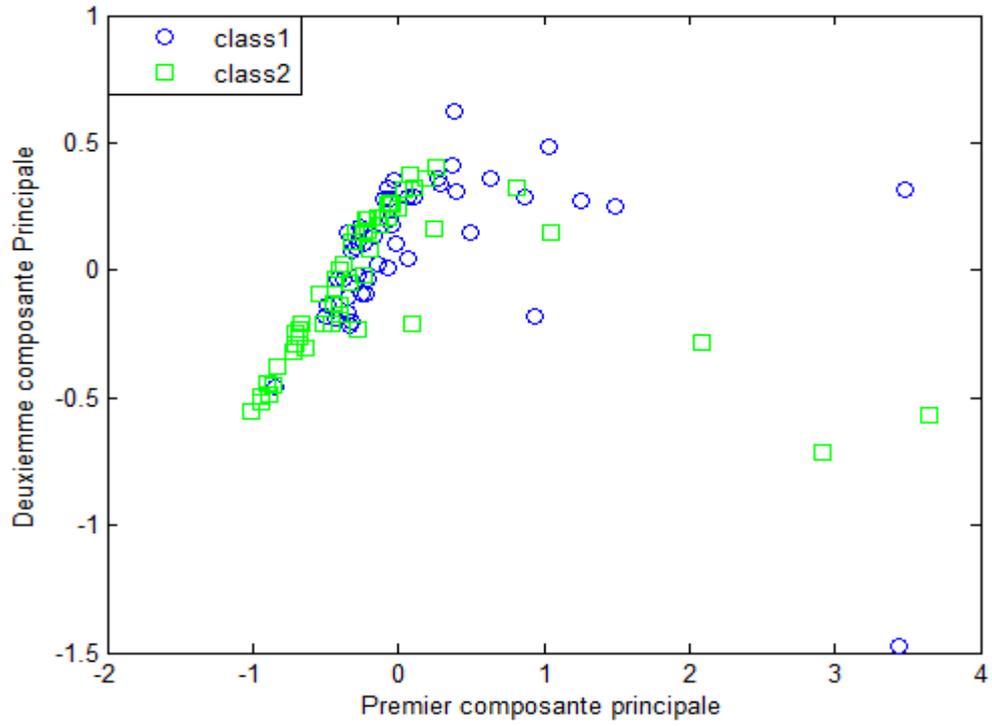


Figure II.10: Répartition des deux classes de la base de données Fraise

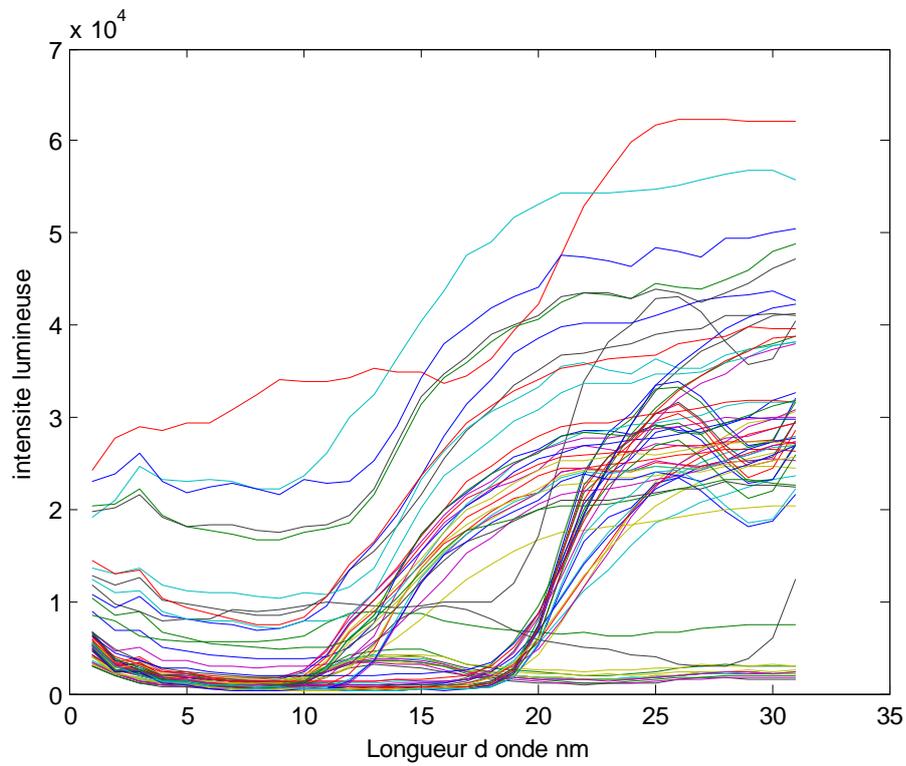


Figure II.11: Les spectres de l'ensemble d'apprentissage de la base de données Poivrons

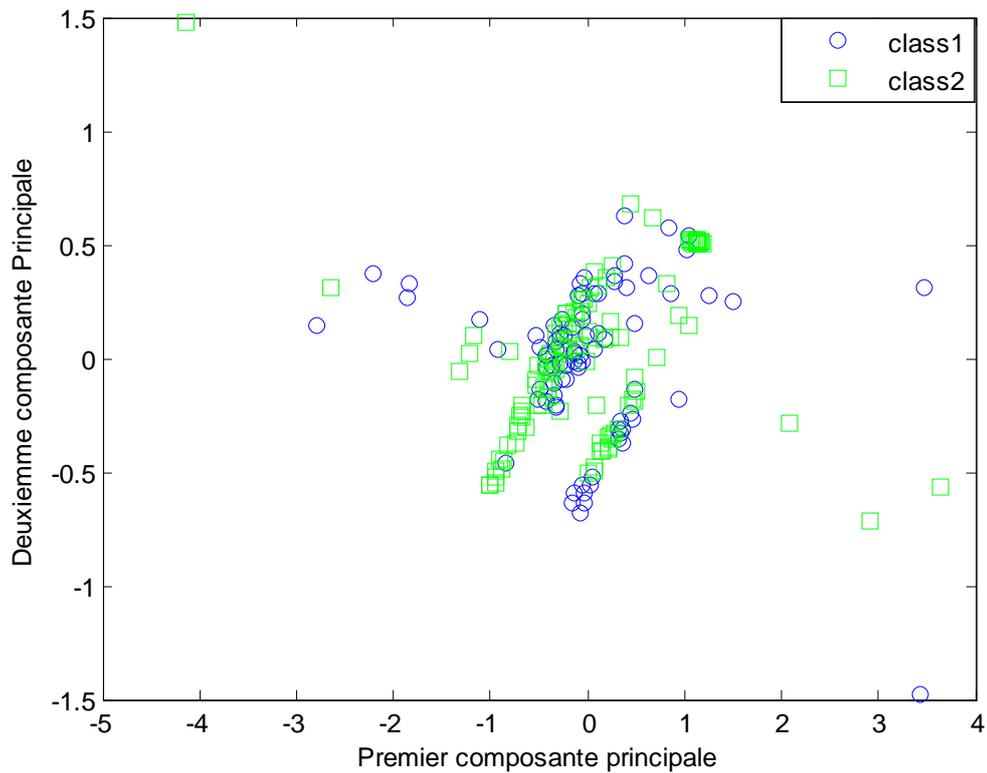


Figure II.12 Répartition des deux classes de la base de données Poivron

II.3.3. Base de données poisson

La base de données poissons consiste en 158 poissons de 7 espèces différentes. Tous les poissons sont capturés du même lac en Finlande. Pour Chaque observation on a 6 variables. Le problème dans cette base de données consiste à connaître à quelle espèce appartient le poisson capturé en se basant dépend sur les mesures prises sur le corps du poisson.

La Figure II.13 montre les images de 7 espèces de poisson, et la figure II.14 illustre les mesures prises sur le corps du poisson.

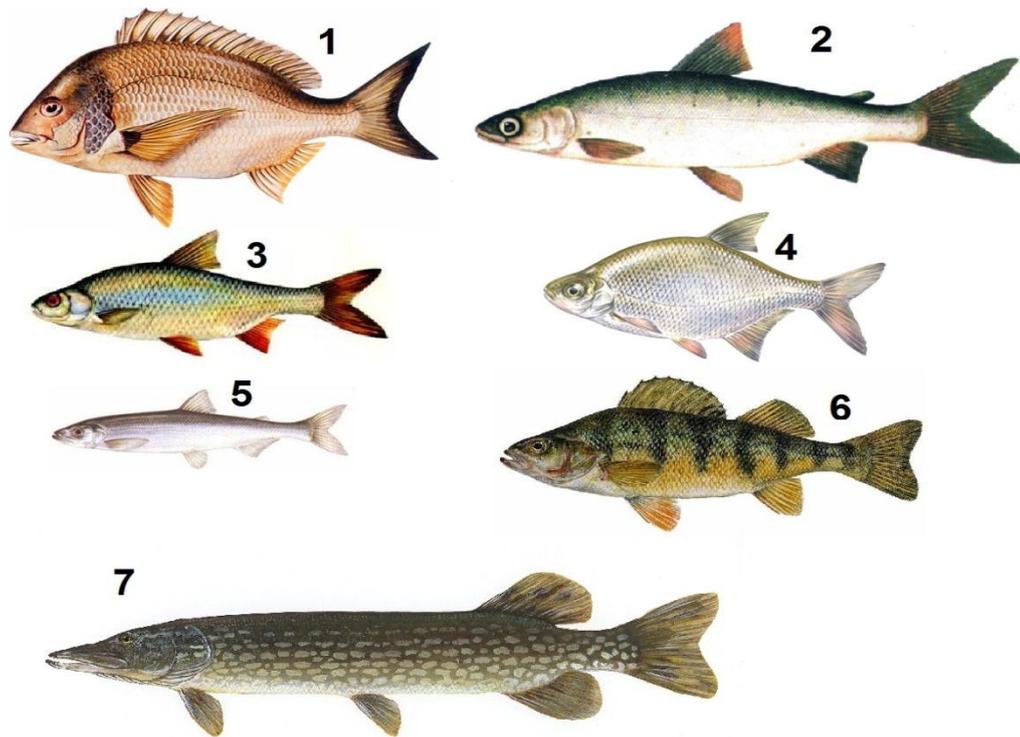


Figure II.13: images des poissons de la base de données Poisson

La figure II.13 montre les sept espèces de poisson utilisés dans la base de données Poisson tel que: 1- Bream, 2- Siika, 3- Roach, 4- Bjoerknan, 5- Smelt, 6- Perch; 7- Pike.

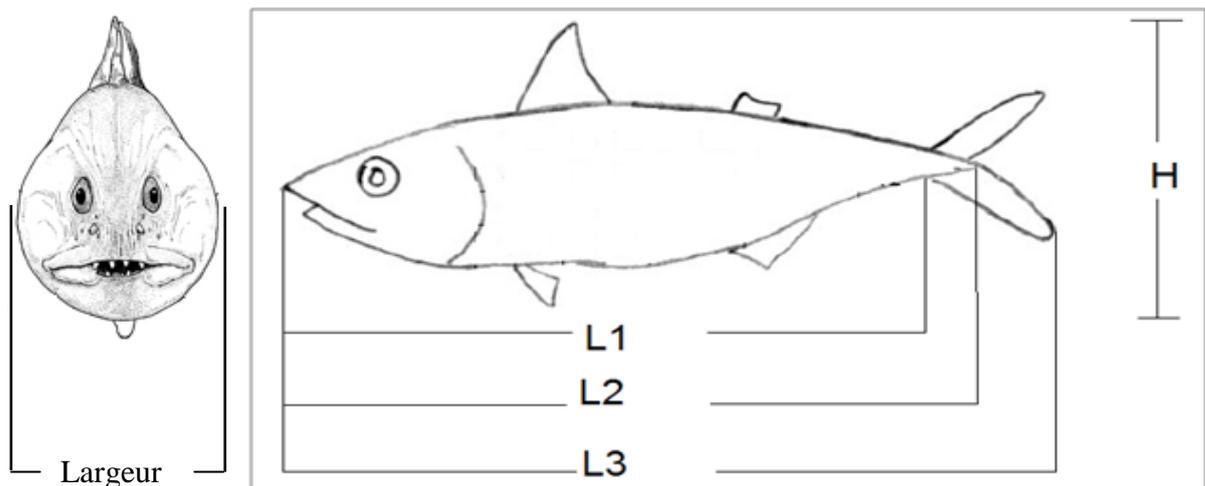


Figure II.14: Différentes caractéristiques mesures prises sur le corps de poisson

Description des variables:

1. Poids : poids du poisson (en grammes).
2. Longueur 1 : longueur de nez jusqu'au début de la queue (en cm).
3. Longueur 2 : longueur de nez jusqu'à l'entaille de la queue (en cm).
4. Longueur 3 : longueur de nez jusqu'à la fin de la queue (en cm).
5. Hauteur : hauteur maximum du poisson.
6. Largeur : largeur maximum du poisson.

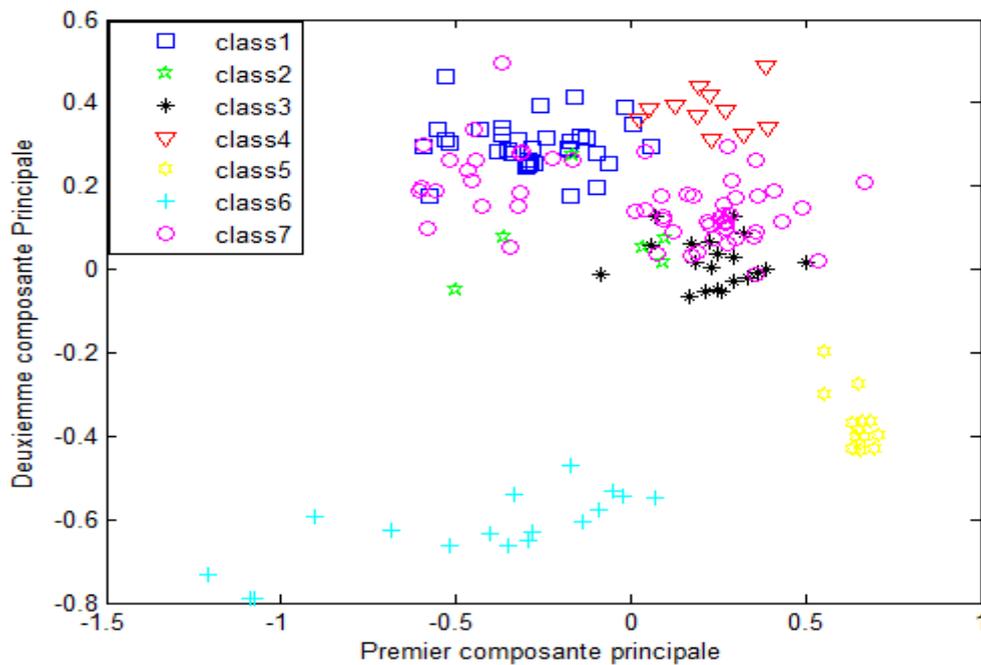


Figure II.15 Répartition des sept classes de la base de données Poisson

II.4. Conclusion:

Dans ce chapitre nous avons présenté la spectroscopie en infrarouge pour les produits alimentaires, et une introduction sur les images multispectrale. Par la suite nous avons présenté les bases de données utilisées dans ce travail. La visualisation est l'objectif primordial de l'analyse des données, une représentation bidimensionnelle est souvent préférée pour voir la répartition des données dans cette espace. Cette représentation fournit des informations intéressantes sur la distribution des données et d'autres tels que les valeurs aberrantes. Pour la visualisation des données nous avons utilisés l'analyse en composante principale.

L'objectif du chapitre suivant est la description et l'utilisation des méthodes de classification pour prédire la classe de produit alimentaire.

Chapitre III

Etude expérimentale

III.1. Introduction

Ce chapitre présente l'étude expérimentale des résultats de simulation obtenus par l'application des classificateurs (RBFN, SVM et KRR) aux quatre bases de données (Huile d'olive, Fraise, Poivron et Poisson).

Dans la première partie de ce chapitre nous évaluerons les performances en classification des modèles RBFN, SVM et KRR sur les quatre bases de données, et dans la deuxième partie nous avons étudié l'influence de la réduction de la dimension des données (nombre de variables d'entrée pour les classificateurs) sur les classificateurs, la réduction se fait par l'application de l'ACP.

III.2. Les résultats expérimentaux :

III.2.1. Évaluation des performances :

III.2.1.1. Taux de classification:

Dans le domaine de la classification, nous avons besoin d'un critère de précision. Dans notre cas nous avons choisi de calculer le taux de classification, il est donné par la formule suivante:

$$Taux(\%) = \frac{P}{N} \times 100 \quad \text{III.1}$$

P : Echantillons bien classées.

N : Nombre d'échantillons des classes cibles.

III.2.1.2. Matrice de confusion

La matrice de confusion, dans la terminologie de l'apprentissage supervisé, est un outil servant à mesurer la qualité d'un système de classification. Chaque colonne de la matrice représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence).

Un des intérêts de la matrice de confusion est qu'elle montre rapidement si le système parvient à classifier correctement.

Un modèle sans erreurs aura ses résultats concentrés sur la diagonale de sa matrice de confusion.

III.2.2. Optimisation des paramètres des modèles RBFN, SVM et KRR

Dans cette partie du travail une série d'expériences a été menée pour essayer de trouver les meilleurs modèles en utilisant l'ensemble d'apprentissage et l'ensemble de validation (détermination des paramètres optimaux) des trois classificateurs (RBFN, SVM et KRR) pour

chaque base de données. Après la détermination du meilleur modèle on l'applique sur l'ensemble de test.

La technique de validation utilisée dans ce travail est la K-fold validation. Nous avons choisi cette technique pour sa certitude que chaque échantillon sera utilisé par l'ensemble d'apprentissage et l'ensemble de validation.

Pour la détermination du meilleur modèle, il est nécessaire de déterminer la valeur du paramètre de régularisation C le paramètre σ pour le modèle SVM, et la valeur de σ le nombre de neurones pour le modèle RBFN, et la valeur du paramètre de régularisation λ ainsi que la valeur de σ pour le modèle KRR.

De manière générale, pour estimer les bons paramètres de la fonction noyau utilisée il n'existe pas de critères universels permettant de choisir ces paramètres. Ce choix dépend de l'application. Ici, l'objectif est de trouver les paramètres qui donnent des meilleurs taux de classification. Sur la base de cette étude expérimentale, une grille d'optimisation est utilisée. La grille est constituée de 20×20 point dont les valeurs minimale et maximale de C sont respectivement, égales à 0.1 et 10^4 et celle de σ valent 10^{-4} et 10 pour le SVM. Dans le cas de RBFN la grille est constituée de 20×30 point dont les valeurs minimale et maximale de σ sont respectivement, égales à 10^{-4} et 10, et le nombre de neurones varie de 2 à 60 avec un pas de 2. Et pour le KRR on construit une grille de 20×20 point dont les valeurs minimale et maximale de λ sont respectivement 10^{-7} et 10^{-2} et celle de σ valent 10^{-4} et 10. Les valeurs de C , σ et λ testées varient de façon logarithmique.

Dans le cas multiclasse nous avons utilisé la stratégie un contre un, donc pour k classe nous avons $k \times (k - 1) / 2$ classificateurs biclasses avec les mêmes paramètres pour chaque hyperplan séparateur.

III.2.3. Les résultats d'apprentissage:

Dans cette section ne sont utilisé que l'ensemble d'apprentissage et l'ensemble de validation pour l'évaluation du meilleur modèle de classification.

Les figures ci-dessous montrent l'évolution du taux de classification de validation en fonction des paramètres des classificateurs RBFN (σ et nombre de neurones), des paramètres de SVM (σ et C) et des paramètres de KRR (σ et λ) pour les quatre bases de données dans la phase d'apprentissage.

Les figures III.1, III.2, III.3 et III.4 montrent l'évolution du taux de classification de validation en fonction du nombre de neurones dans la couche cachée du classificateur RBFN sur les

quatre bases de données Huile d'olive, Fraise, Poivrons et Poisson respectivement, cependant la Figure III.1 montre que le nombre de neurones optimal pour la base huile d'olive est égale à 10 neurones, la Figure III.2 montre que le nombre de neurones optimal pour la base Fraise est égale à 60 neurones, la Figure III.3 montre que le nombre de neurones optimale pour la base Poivrons est égale à 54 neurones, la Figure III.4 montre que le nombre de neurones optimale pour la base Poisson est égale à 8 neurones.

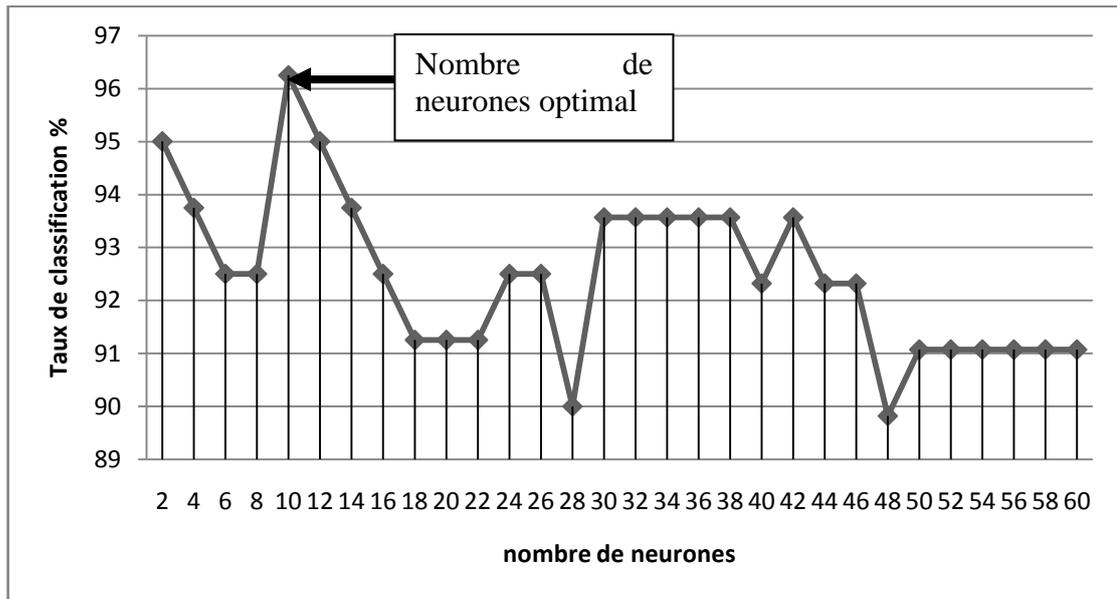


Figure III.1 : Evolution du taux de classification de validation en fonction du nombre de neurones dans la couche cachée du classificateur RBFN sur la base huile d'olive

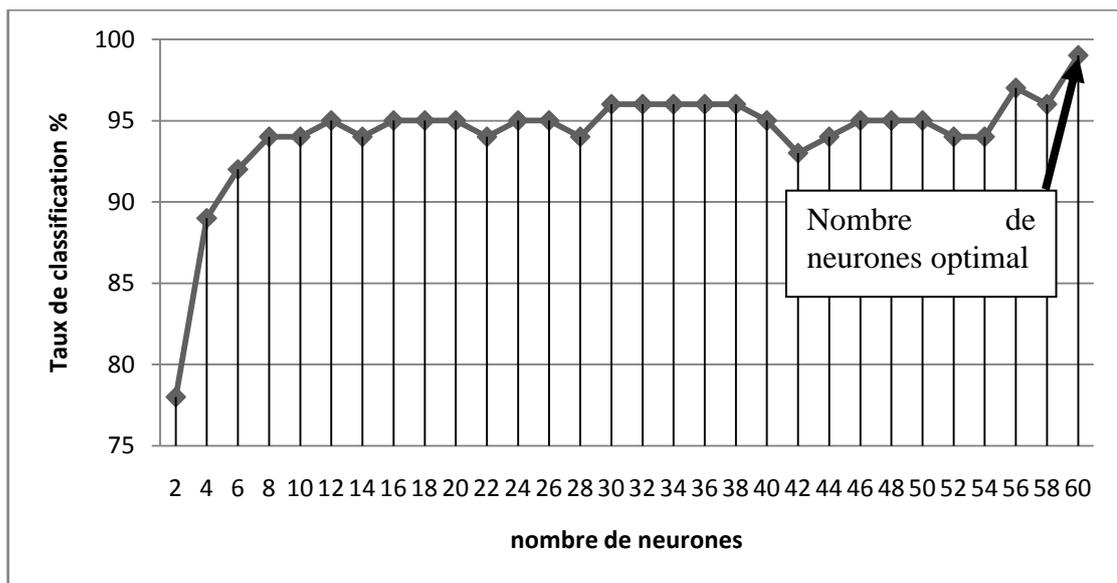


Figure III.2 : Evolution du taux de classification de validation en fonction du nombre de neurones dans la couche cachée du classificateur RBFN sur la base fraise

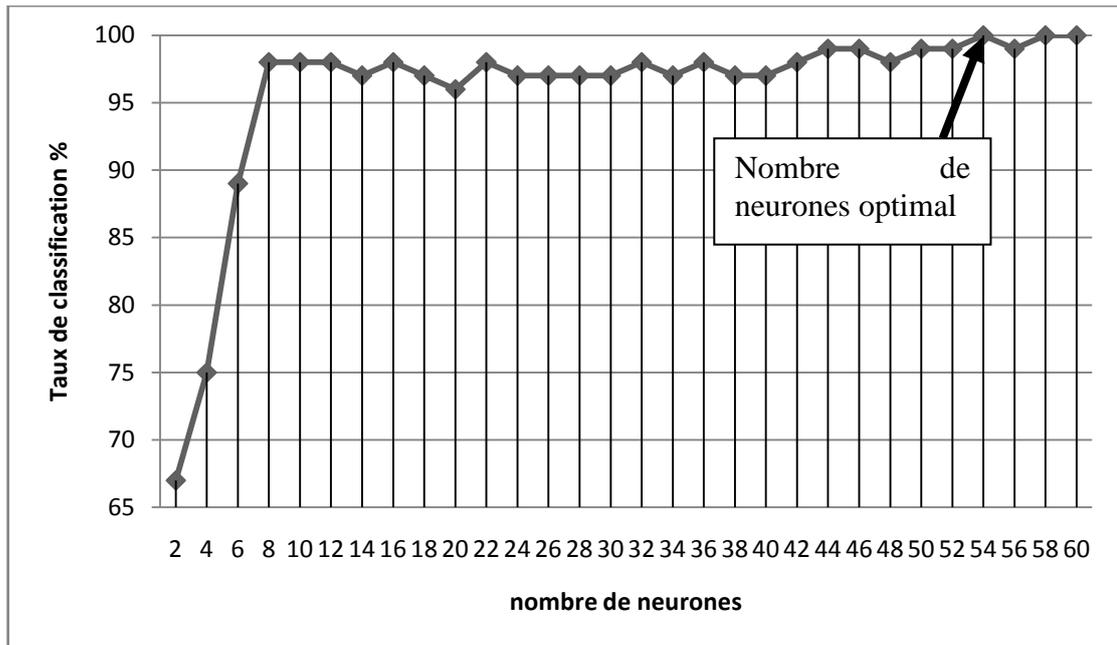


Figure III.3: Evolution du taux de classification de validation en fonction du nombre de neurones dans la couche cachée du classificateur RBFN sur la base poivron

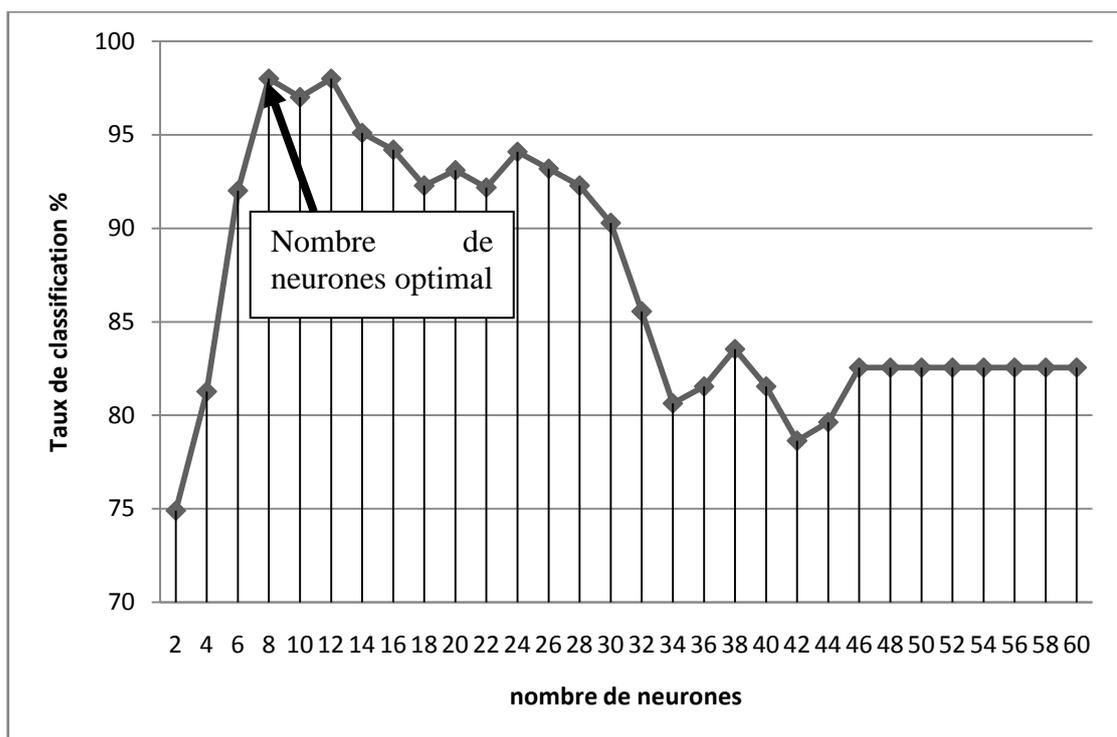


Figure III.4: Evolution du taux de classification de validation en fonction du nombre de neurones dans la couche cachée du classificateur RBFN sur la base poisson.

Les Figure III.5, III.6, III.7 et III.8 montrent l'évolution du taux de classification de validation en fonction de sigma du classificateur RBFN sur les quatre bases de données Huile d'olive, Fraise, Poivrons, Poisson respectivement. La Figure III.5 montre que le sigma optimal pour la base Huile d'olive est égale à 10, la Figure III.6 montre que le sigma optimal pour la base Fraise est égale à 1.62, la Figure III.7 montre que le sigma optimal pour la base Poivrons est égale à 5.45, et enfin la Figure III.8 montre que le sigma optimal pour la base Poisson est égale à 5.45.

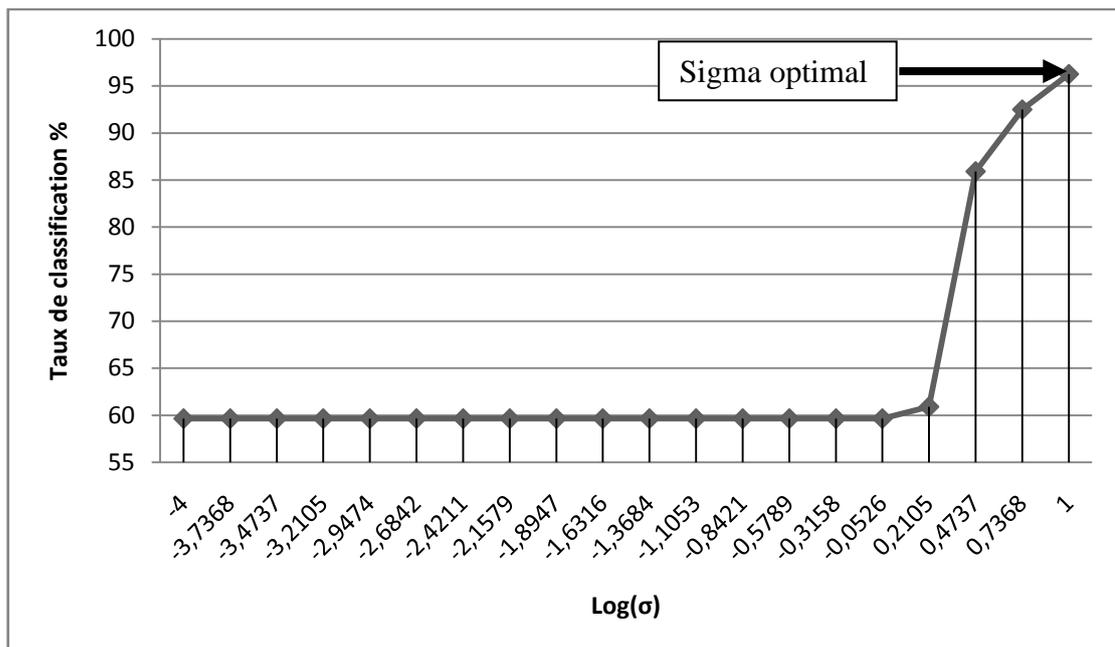


Figure III.5 : Evolution du taux de classification de validation en fonction de sigma du classificateur RBFN sur la base huile d'olive

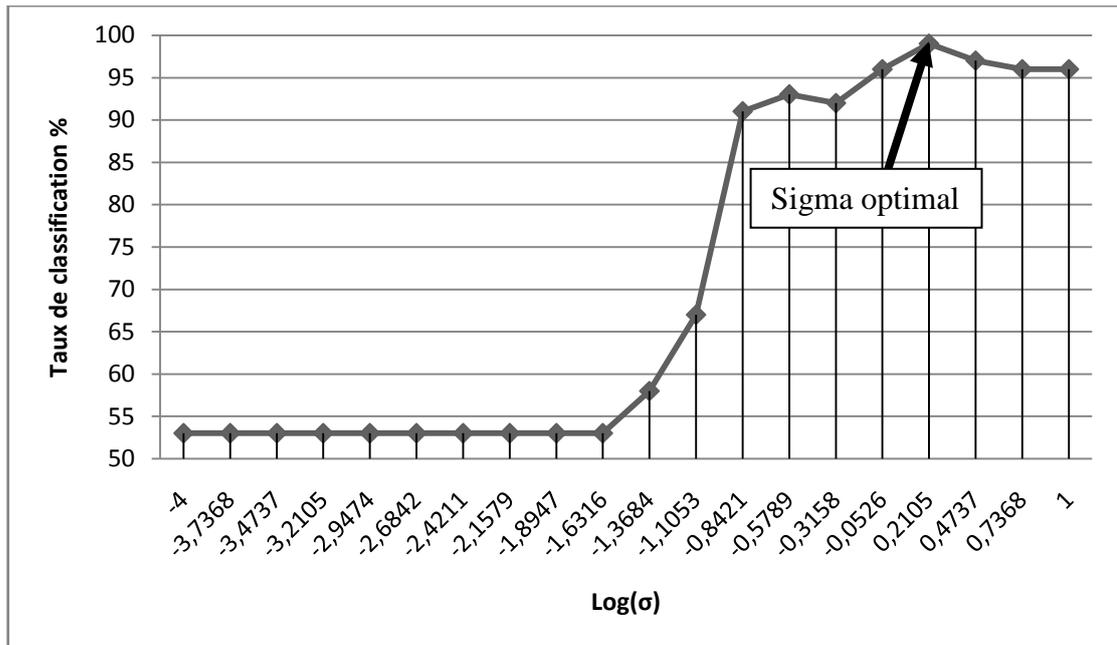


Figure III.6: Evolution du taux de classification de validation en fonction de sigma du classificateur RBFN sur la base fraise

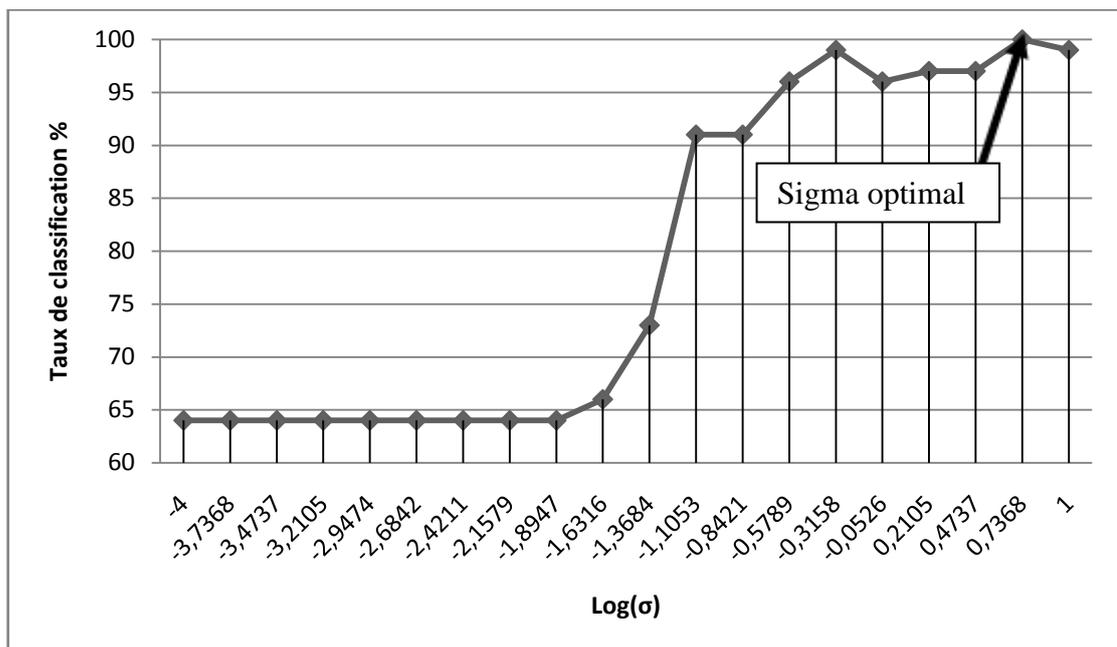


Figure III.7: Evolution du taux de classification de validation en fonction de sigma du classificateur RBFN sur la base poivron

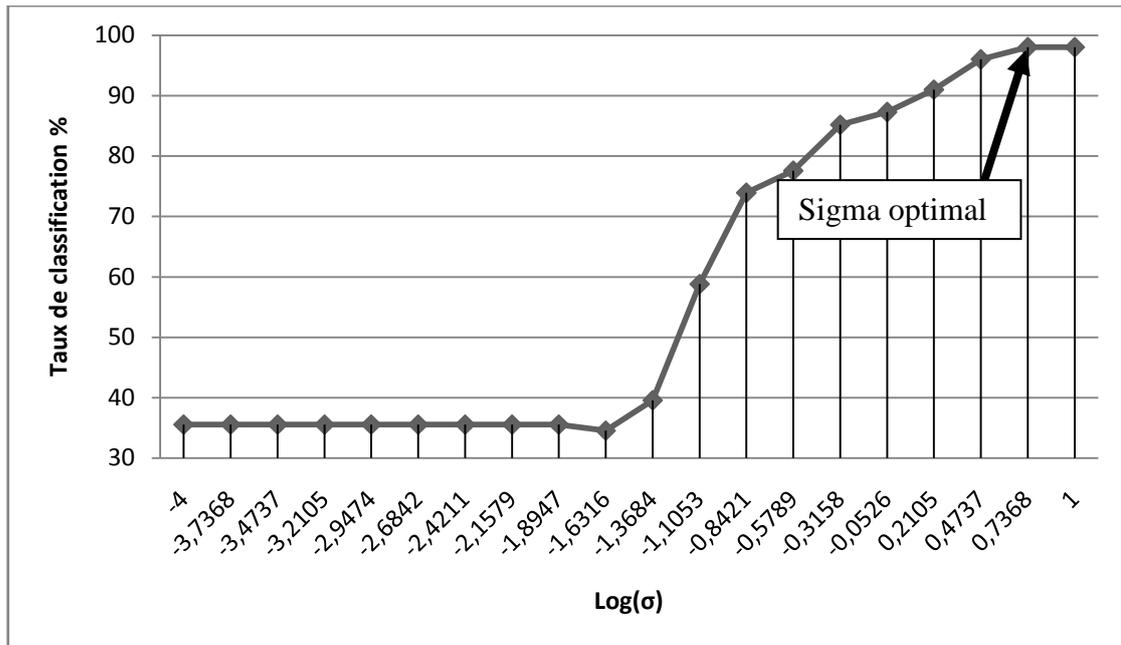


Figure III.8: Evolution du taux de classification de validation en fonction de sigma du classificateur RBFN sur la base poisson.

Les figures III.9, III.10, III.11 et III.12 montrent l'évolution du taux de classification de validation en fonction des paramètre de régularisation "C" du classificateur SVM sur les quatre bases de données Huile d'olive, Fraise, Poivron, Poisson respectivement, la Figure III.9 montre que le paramètre de régularisation "C" optimale pour la base Huile d'olive est égale à 78.47, la Figure III.10 montre que le paramètre de régularisation "C" optimal pour la base Fraise est égale à 10^4 , la Figure III.11 montre que le paramètre de régularisation "C" optimal pour la base Poivron est égale à 10^4 , la Figure III.12 montre que le paramètre de régularisation "C" optimal pour la base Poisson est égale à 3792.

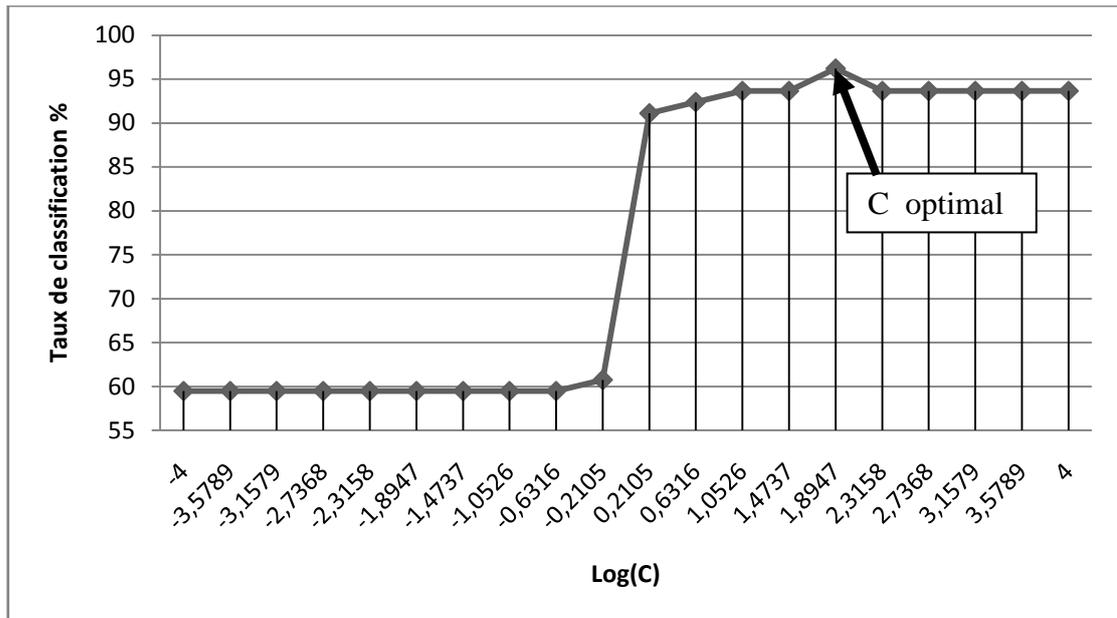


Figure III.9: Evolution du taux de classification de validation en fonction de C du classificateur SVM sur la base huile d'olive

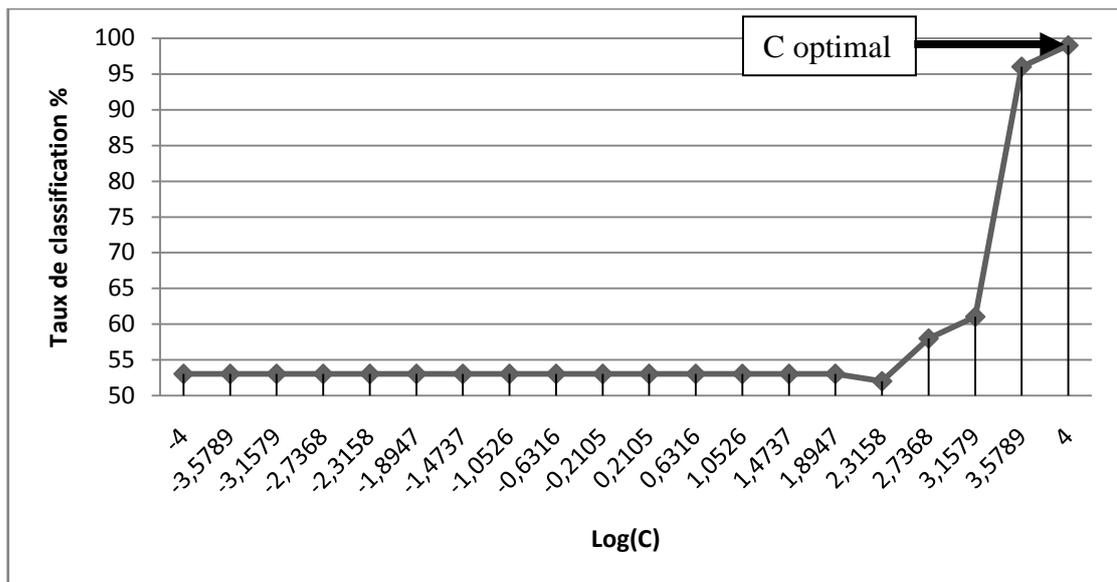


Figure III.10: Evolution du taux de classification de validation en fonction de C du classificateur SVM sur la base fraise

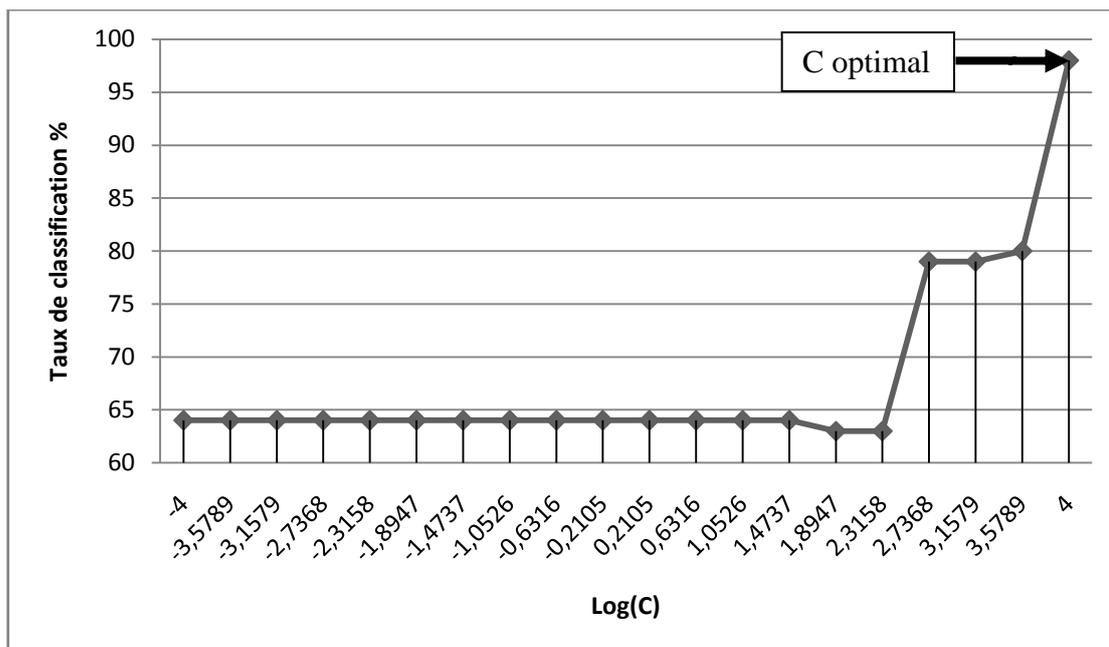


Figure III.11: Evolution du taux de classification de validation en fonction de C du classificateur SVM sur la base poivron

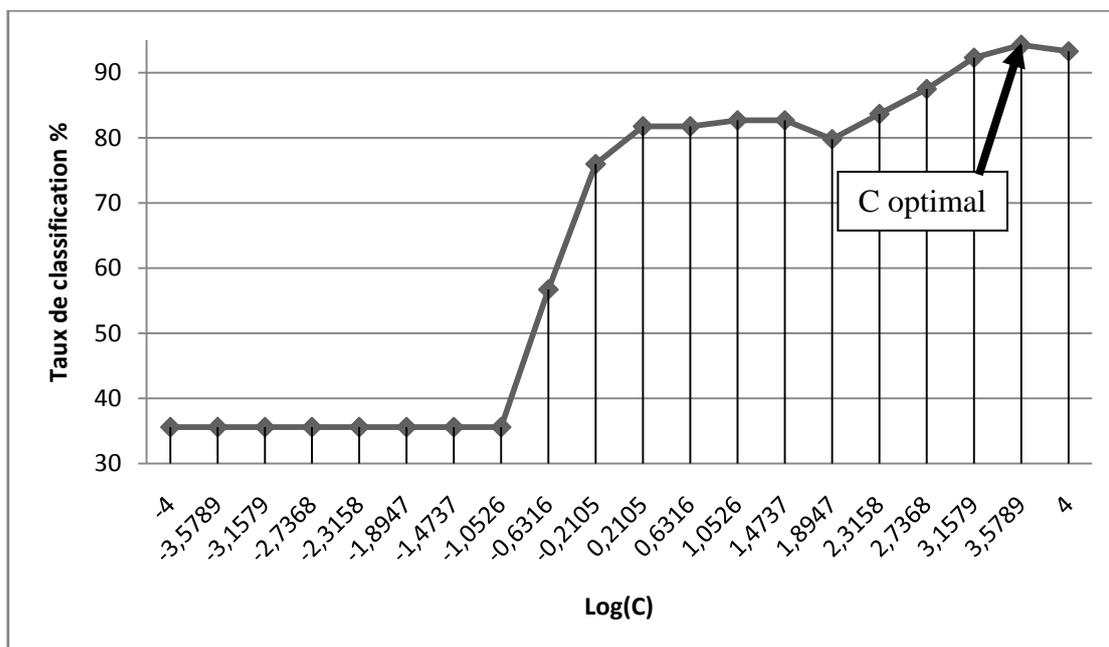


Figure III.12: Evolution du taux de classification de validation en fonction de C du classificateur SVM sur la base Poisson

Les figures III.13, III.14, III.15 et III.16 montrent l'évolution du taux de classification de validation en fonction de sigma du classificateur SVM sur les quatre bases de données Huile d'olive, Fraise, Poivrons, Poisson respectivement, la Figure III.13 montre que le sigma

optimal pour la base Huile d'olive est égale à 2×10^{-4} , la Figure III.14 montre que le sigma optimal pour la base Fraise est égale à 10^{-4} , la Figure III.15 montre que le sigma optimal pour la base Poivrons est égale à 1.1×10^{-3} , la Figure III.16 montre que le sigma optimal pour la base Poisson est égale à 0.48.

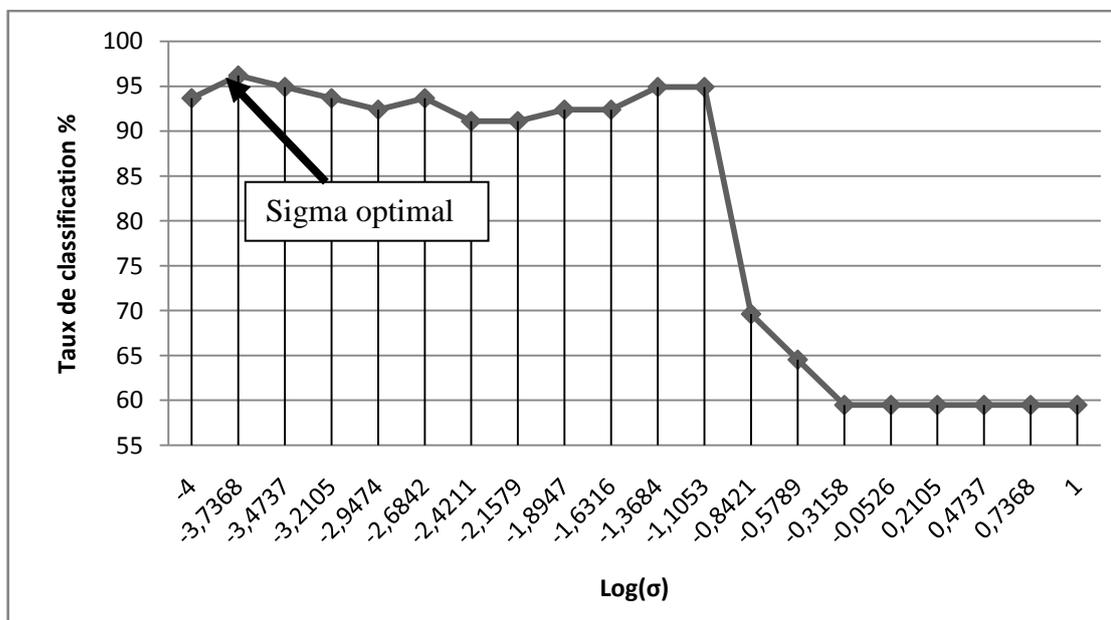


Figure III.13: Evolution du taux de classification de validation en fonction de sigma du classificateur SVM sur la base huile d'olive

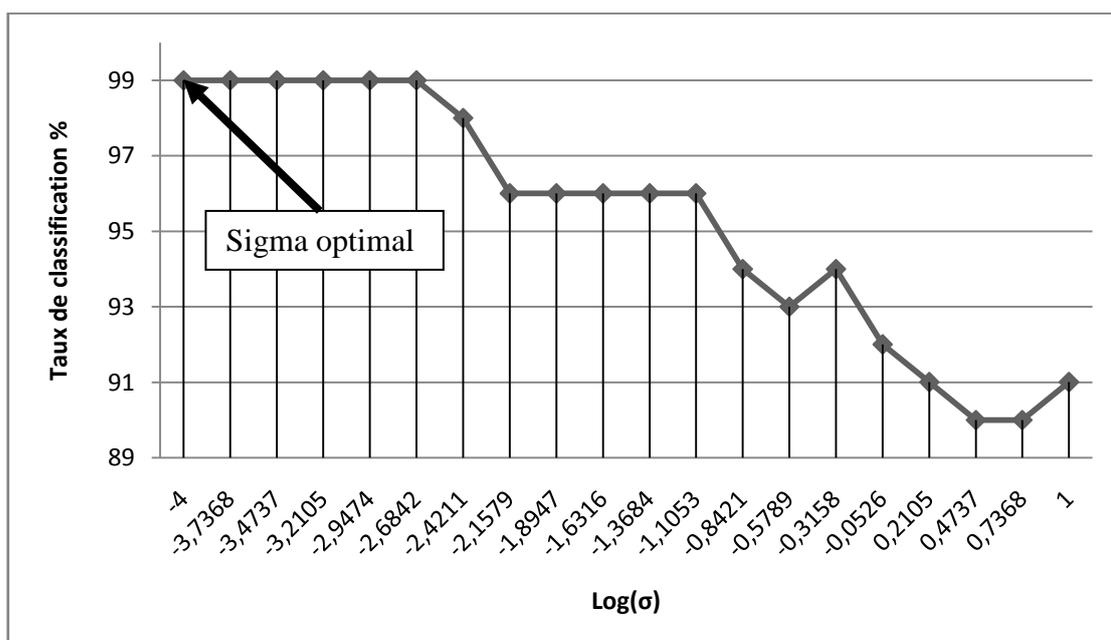


Figure III.14: Evolution du taux de classification de validation en fonction de sigma du classificateur SVM sur la base fraise.

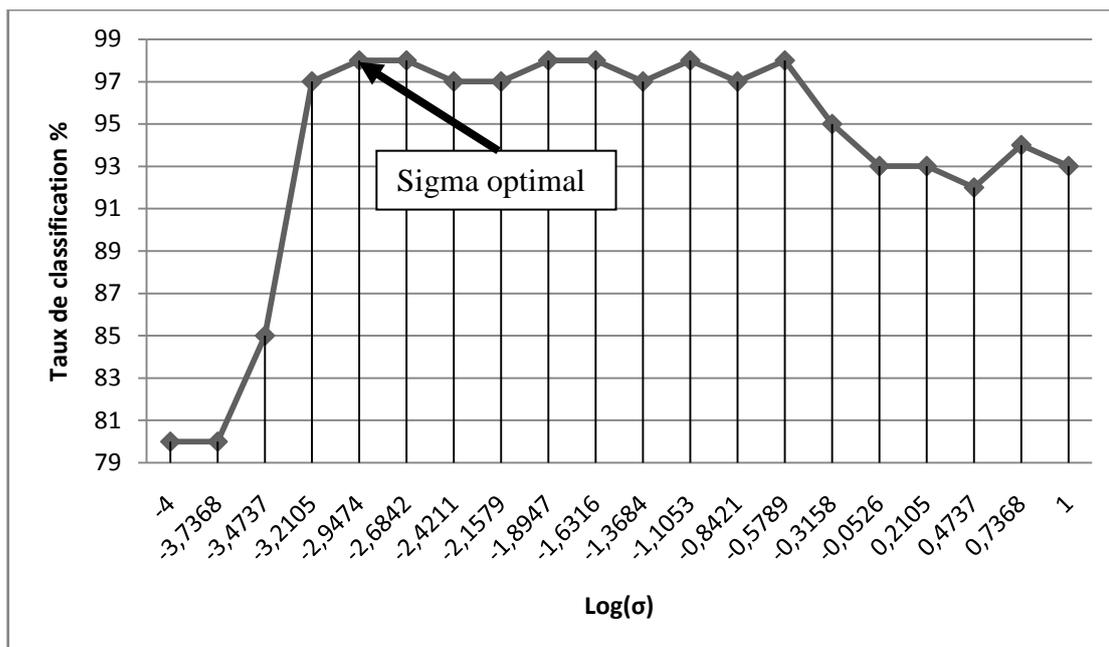


Figure III.15: Evolution du taux de classification de validation en fonction de sigma du classificateur SVM sur la base poivron

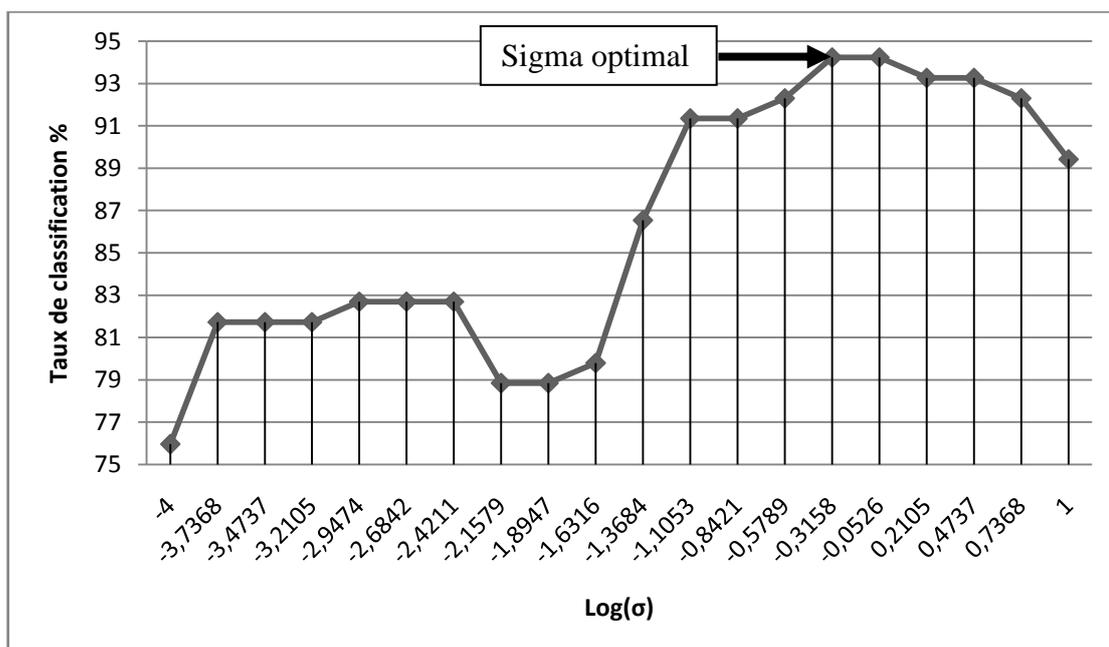


Figure III.16: Evolution du taux de classification de validation en fonction de sigma du classificateur SVM sur la base Poisson

Les figures III.17, III.18, III.19 et III.20 montrent l'évolution du taux de classification de validation en fonction du paramètre de régularisation " λ " du classificateur KRR sur les quatre bases de données Huile d'olive, Fraise, Poivron, Poisson respectivement, la Figure III.17 montre que le paramètre de régularisation " λ " optimal pour la base Huile d'olive est

égale à 4.28×10^{-5} , la Figure III.18 montre que le paramètre de régularisation " λ " optimal pour la base Fraise est égale à 1.43×10^{-4} , la Figure III.19 montre que le paramètre de régularisation " λ " optimal pour la base Poivron est égale à 1.12×10^{-6} , la Figure III.20 montre que le paramètre de régularisation " λ " optimal pour la base Poisson est égale à 10^{-7} .

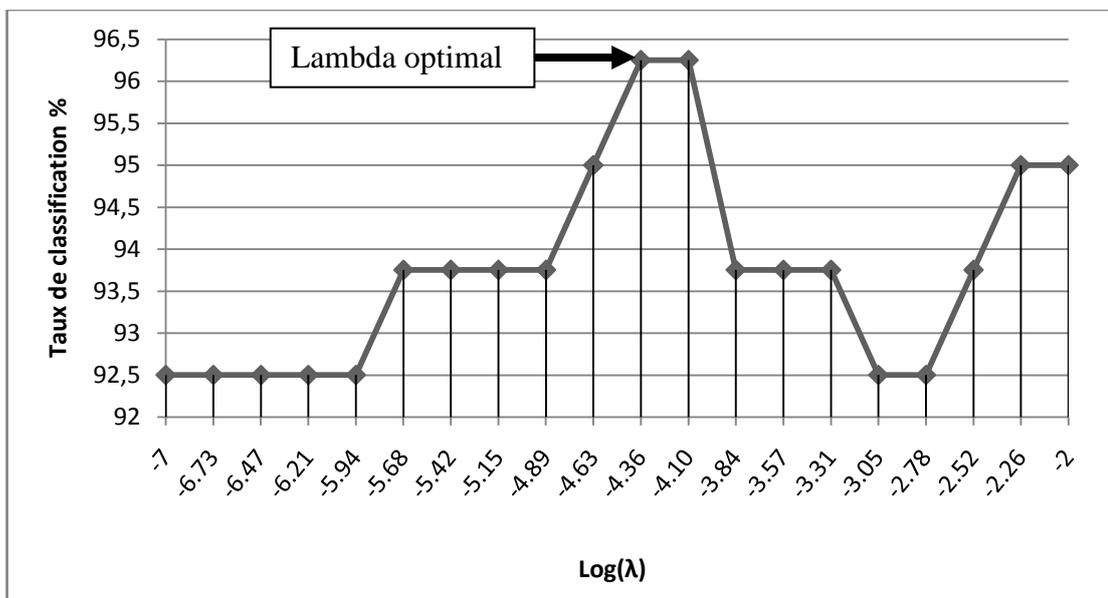


Figure III.17 : Evolution du taux de classification de validation en fonction de lambda du KRR sur la base Huile d'olive

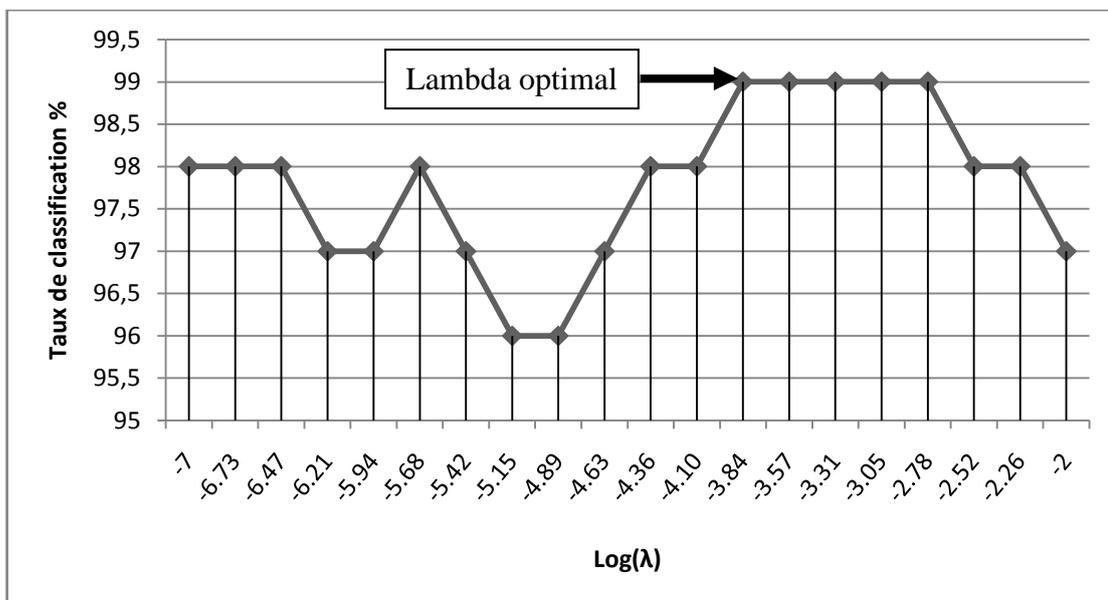


Figure III.18 : Evolution du taux de classification de validation en fonction de lambda du KRR sur la base Fraise

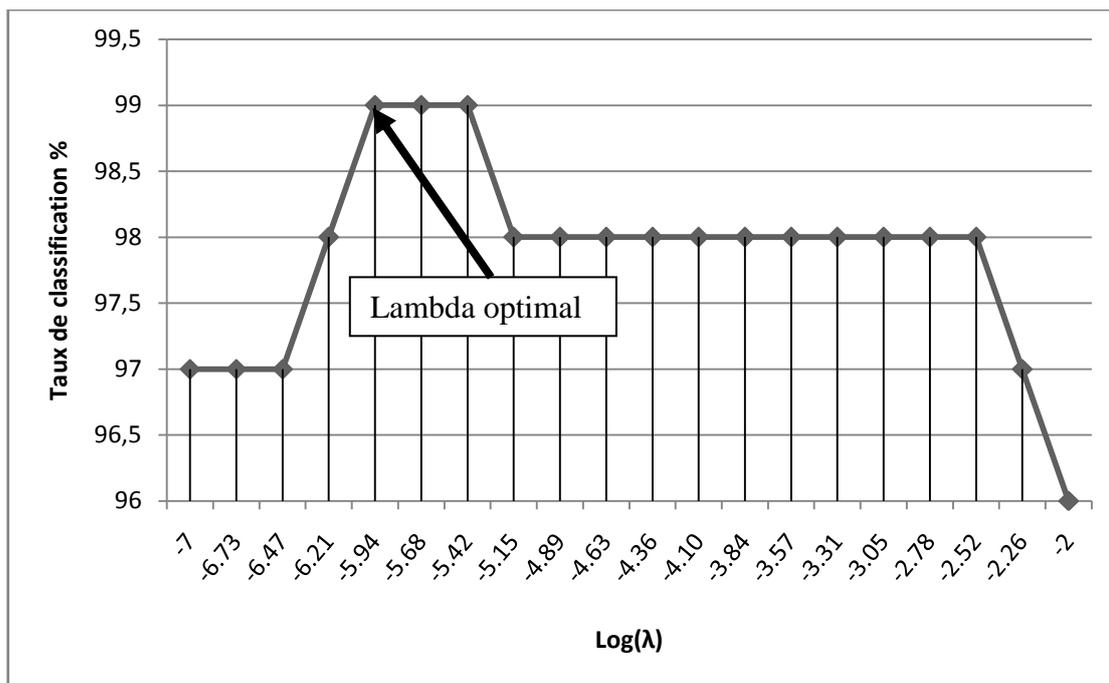


Figure III.19 : Evolution du taux de classification de validation en fonction de lambda du KRR sur la base Poivron

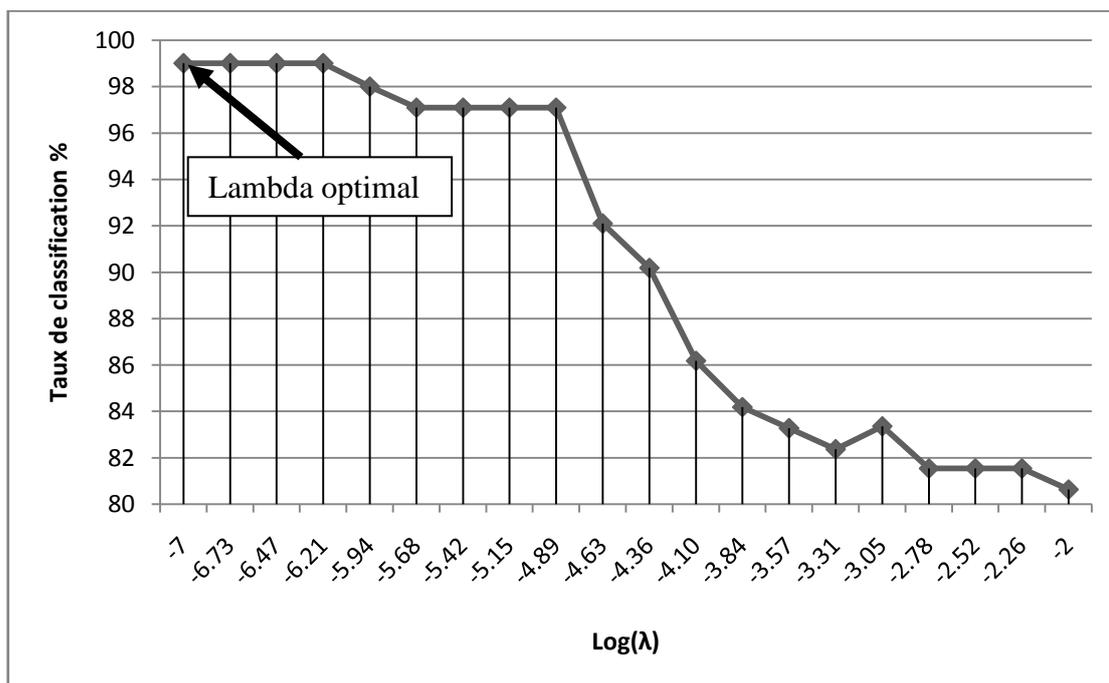


Figure III.20 : Evolution du taux de classification de validation en fonction de lambda du KRR sur la base Poissons

Les figures III.21, III.22, III.23 et III.24 montrent l'évolution du taux de classification de validation en fonction de sigma du classificateur KRR sur les quatre bases de données Huile d'olive, Fraise, Poivrons, Poisson respectivement, la Figure III.21 montre que le sigma

optimal pour la base Huile d'olive est égale à 2×10^{-4} , la Figure III.22 montre que le sigma optimal pour la base Fraise est égale à 1×10^{-4} , la Figure III.23 montre que le sigma optimal pour la base Poivrons est égale à 1.1×10^{-3} , la Figure III.24 montre que le sigma optimal pour la base Poisson est égale à 0.48.

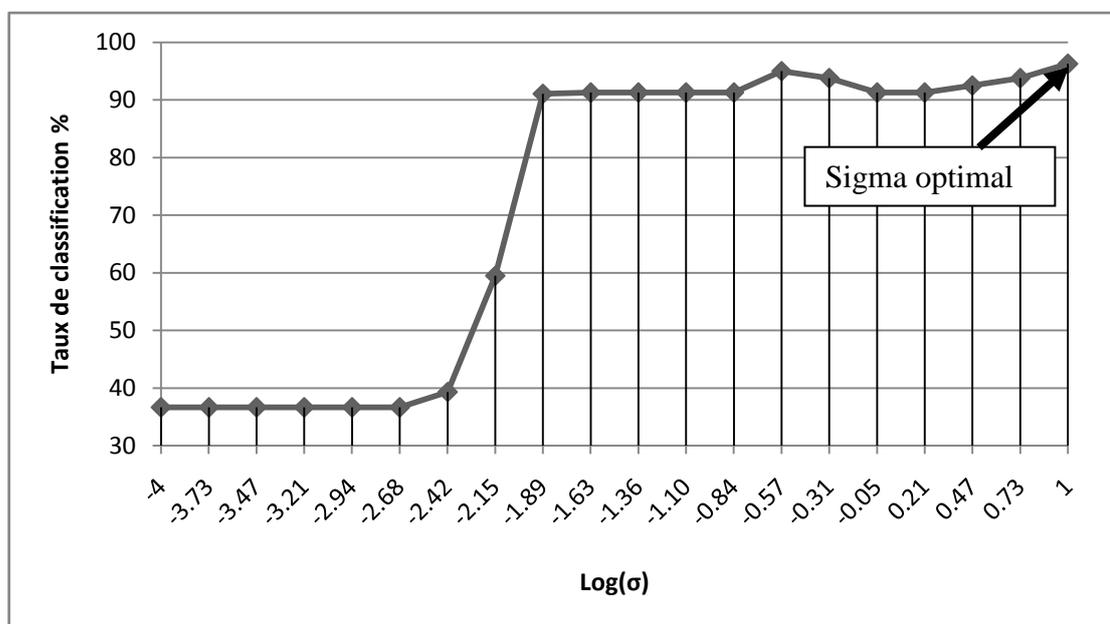


Figure III.21 : Evolution du taux de classification de validation en fonction de sigma du KRR sur la base Huile d'olive

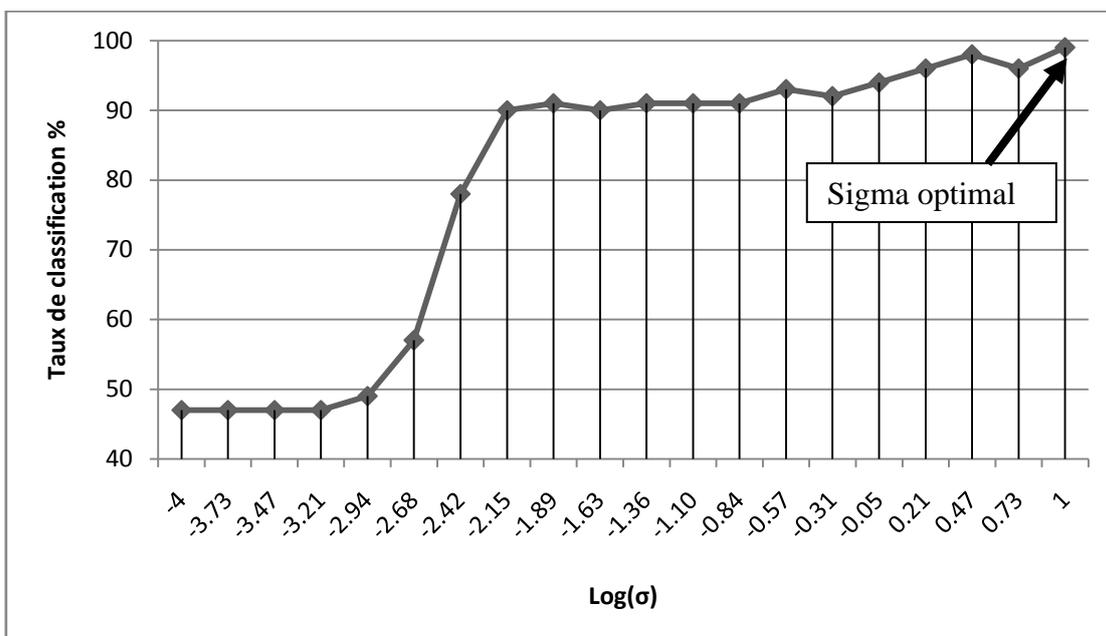


Figure III.22 : Evolution du taux de classification de validation en fonction de sigma du KRR sur la base Fraise

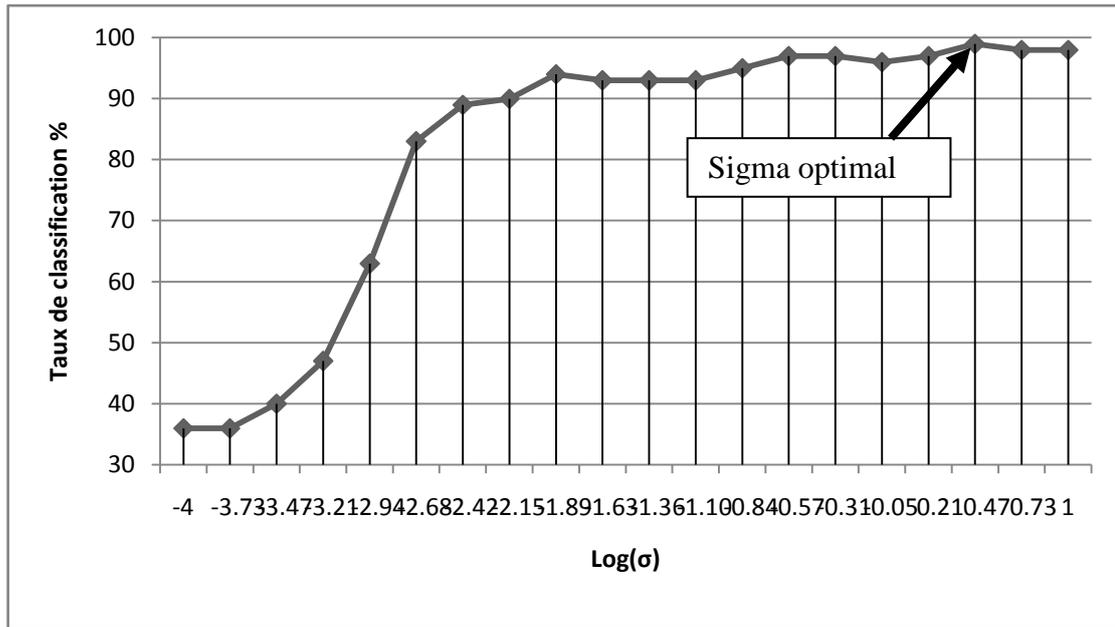


Figure III.23 : Evolution du taux de classification de validation en fonction de sigma du KRR sur la base Poisson

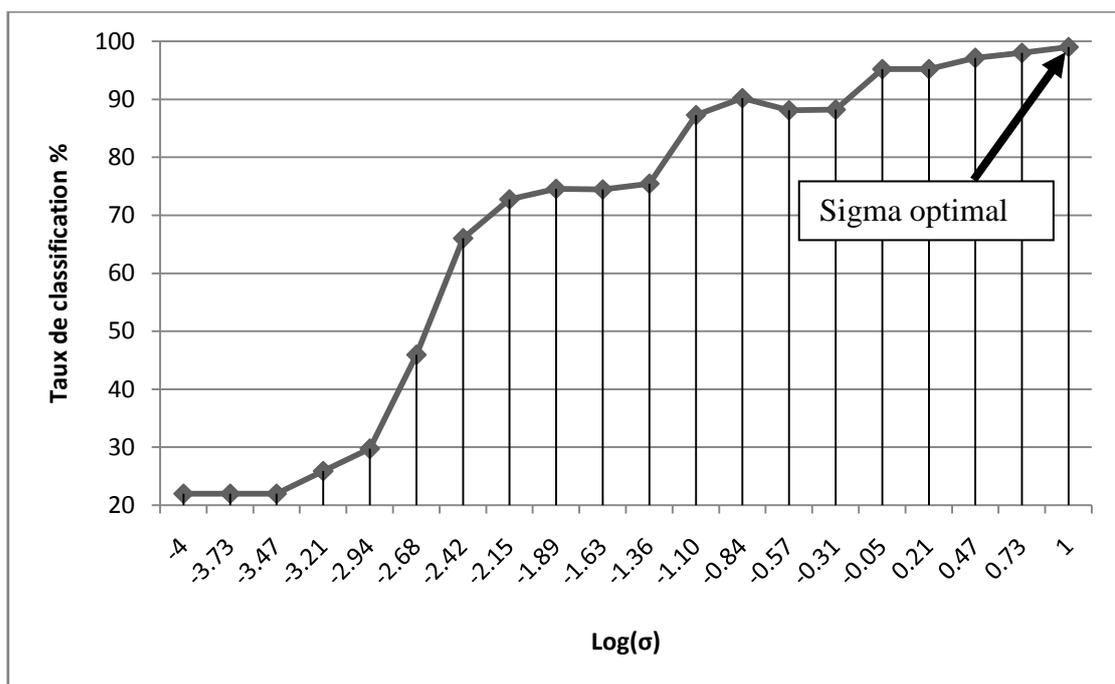


Figure III.24 : Evolution du taux de classification de validation en fonction de sigma du KRR sur la base Poissons

Les paramètres optimaux des classificateurs RBFN, SVM et KRR qui donnent les meilleurs taux de classification pour les quatre bases de données dans la phase d'apprentissage sont répertoriés dans le tableau III.1 pour le classificateur RBFN, le tableau III.2 pour le classificateur SVM et le tableau III.3 pour le classificateur KRR.

Base de données / Paramètre	Huile d'olive	Fraise	Poivron	Poisson
Sigma(σ)	10	1.62	5.45	5.45
Nombre de neurones	10	60	54	8

Tableau III.1: Les paramètres optimaux du classificateur RBFN

Base de données / Paramètre	Huile d'olive	Fraise	Poivron	Poisson
Sigma (σ)	2×10^{-4}	10^{-4}	1.1×10^{-3}	0.48
C	10	10^4	10^4	3792

Tableau III.2: Les paramètres optimaux du classificateur SVM

Base de données / Paramètre	Huile d'olive	Fraise	Poivron	Poisson
Sigma (σ)	10	10	2.97	10
Lambda (λ)	10^{-2}	1.43×10^{-4}	1.12×10^{-6}	10^{-7}

Tableau III.3: Les paramètres optimaux du classificateur KRR

III.2.4. Les résultats de test:

Dans cette section nous allons appliquer les meilleurs modèles obtenus qui correspondent aux meilleurs taux de classification dans la section des résultats d'apprentissage, sur l'ensemble de test. Les résultats sont présentés dans les tableaux ci-dessous.

III.2.4.1. Résultats de la base de donnée Huile d'olive

III.2.4.1.1. Application RBFN

L'application du modèle RBFN obtenu avec les paramètres optimaux $\sigma = 10$ et $ncc = 10$ neurones, sur l'ensemble de test de la base de données huile d'olive donne le taux de classification égale à 95.12 %. Le taux de classification pour chaque classe est illustré dans le tableau III.4 (Matrice de confusion)

Classes	1 (Estimée)	2 (Estimée)	3 (Estimée)	Nombre/classe	taux
1 (Réelle)	14	1	0	15	93.33 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	0	1	1	2	50 %

Tableau III.4: Matrice de confusion obtenue par l'application du modèle RBFN sur la base de données Huile d'olive.

III.2.4.1.2. Application SVM

L'application du modèle SVM obtenu avec les paramètres optimaux $\sigma = 2 \times 10^{-4}$ et $C = 78.47$, sur l'ensemble de test de la base de données huile d'olive donne le taux de classification égale à 92.68 %. Le taux de classification pour chaque classe est illustré dans le tableau III.5 (Matrice de confusion)

Classes	1 (Estimée)	2 (Estimée)	3 (Estimée)	Nombre/classe	taux
1 (Réelle)	14	1	0	15	93.33 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	1	1	0	2	0 %

Tableau III.5: Matrice de confusion obtenue par l'application du modèle SVM sur la base de données Huile d'olive.

III.2.4.1.3. Application KRR

L'application du modèle KRR obtenu avec les paramètres optimaux $\lambda = 10^{-2}$ et $\sigma = 10$, sur l'ensemble de test de la base de données huile d'olive donne le taux de classification égale à 92.68 %. Le taux de classification pour chaque classe est illustré dans le tableau III.6 (Matrice de confusion)

Classes	1 (Estimée)	2 (Estimée)	3 (Estimée)	Nombre/classe	taux
1 (Réelle)	13	1	1	15	86.66 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	1	0	1	2	50 %

Tableau III.6: Matrice de confusion obtenue par l'application du modèle KRR sur la base de données Huile d'olive

Dans les tableaux III.4, III.5, et III.6 on constate que les taux de classification de la classe "1" et la classe "2" est élevé (93.33% pour la classe "1" et 100% pour la classe "2"), par contre pour la troisième classe le taux de classification ne dépasse pas le 50%, et ceci expliqué par le petit nombre d'échantillons d'apprentissage et la nature de cette classe (mélange des deux premières classes).

III.2.4.2. Résultats de la base de donnée Fraise

III.2.4.2.1. Application RBFN

L'application du modèle RBFN obtenu avec les paramètres optimaux $\sigma = 1.62$ et $ncc = 60$ neurones, sur l'ensemble de test de la base de données Fraise donne un taux de classification égale à 97.97 %. Le taux de classification pour chaque classe est illustré dans le tableau III.7 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	8949	242	9236	97.37 %
2 (Réelle)	148	9828	9976	98.51 %

Tableau III.7 : Matrice de confusion obtenue par l'application du modèle RBFN sur la base de données Fraise.

III.2.4.2.2. Application SVM

L'application du modèle SVM obtenu avec les paramètres optimaux $\sigma = 10^{-4}$ et $C = 10^4$, sur l'ensemble de test de la base de données Fraise donne un taux de classification égale à 99.12%. Le taux de classification pour chaque classe est illustré dans le tableau III.8 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	9132	104	9236	98.87 %
2 (Réelle)	65	9911	9976	99.34 %

Tableau III.8 : Matrice de confusion obtenue par l'application du modèle SVM sur la base de données Fraise.

III.2.4.2.3. Application KRR

L'application du modèle KRR obtenu avec les paramètres optimaux $\lambda = 1.43 \times 10^{-4}$ et $\sigma = 10$, sur l'ensemble de test de la base de données Fraise donne le taux de classification égale à 99.47 %. Le taux de classification pour chaque classe est illustré dans le tableau III.9 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	9172	64	9236	99.30 %
2 (Réelle)	37	9939	9976	99.62 %

Tableau III.9 : Matrice de confusion obtenue par l'application du modèle KRR sur la base de données Fraise

III.2.4.3. Résultats de la base de donnée Poivron

III.2.4.3.1. Application RBFN

L'application du modèle RBFN obtenu avec les paramètres optimaux $\sigma = 5.45$ et $ncc = 54$ neurones, sur l'ensemble de test de la base de données Poivron donne un taux de classification égale à 98.53 %. Le taux de classification pour chaque classe est illustré dans le tableau III.10 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	23460	226	23686	99.04 %
2 (Réelle)	573	30282	30855	98.14 %

Tableau III.10 : Matrice de confusion obtenue par l'application du modèle RBFN sur la base de données Poivron.

III.2.4.3.2. Application SVM

L'application du modèle SVM obtenu avec les paramètres optimaux $\sigma = 1.1 \times 10^{-3}$ et $C = 10^4$, sur l'ensemble de test de la base de données Poivron donne un taux de classification égale à 98.69 %. Le taux de classification pour chaque classe est illustré dans le tableau III.11 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	23389	297	23686	98.74 %
2 (Réelle)	416	30439	30855	98.65 %

Tableau III.11 : Matrice de confusion obtenue par l'application du modèle SVM sur base de données Poivrons

III.2.4.3.3. Application KRR

L'application du modèle KRR obtenu avec les paramètres optimaux $\lambda = 1.12 \times 10^{-6}$ et $\sigma = 2.97$, sur l'ensemble de test de la base de données Poivrons donne le taux de classification égale à **98.94 %**. Le taux de classification pour chaque classe est illustré dans le tableau III.12 (Matrice de confusion).

Classes	1 (Estimée)	2 (Estimée)	Nombre/classe	taux
1 (Réelle)	23577	129	23686	99.45 %
2 (Réelle)	447	30408	30855	98.55 %

Tableau III.12 : Matrice de confusion obtenue par l'application du modèle KRR sur la base de données Poivrons.

III.2.4.4. Résultats de la base de donnée Poisson

III.2.4.4.1. Application RBFN

L'application du modèle RBFN obtenu avec les paramètres optimaux $\sigma = 5.45$ et $ncc = 8$ neurones, sur l'ensemble de test de la base de données Poisson donne un taux de classification égale à **100 %**. Le taux de classification pour chaque classe est illustré dans le tableau III.13 (Matrice de confusion).

Classes	1(Estimée)	2(Estimée)	3(Estimée)	4(Estimée)	5(Estimée)	6(Estimée)	7(Estimée)	Nombre/classe	taux
1(Réelle)	11	0	0	0	0	0	0	11	100%
2(Réelle)	0	2	0	0	0	0	0	2	100%
3(Réelle)	0	0	7	0	0	0	0	7	100%
4(Réelle)	0	0	0	4	0	0	0	4	100%
5(Réelle)	0	0	0	0	5	0	0	5	100%
6(Réelle)	0	0	0	0	0	6	0	6	100%
7(Réelle)	0	0	0	0	0	0	19	19	100%

Tableau III.13: Matrice de confusion obtenue par l'application du modèle RBFN sur la base de données Poisson.

III.2.4.4.2. Application SVM

L'application du modèle SVM obtenu avec les paramètres optimaux $\sigma = 0.4833$ et $C = 3792$ sur l'ensemble de test de la base de données Poisson donne un taux de classification égale à **92.59 %**. Le taux de classification pour chaque classe est illustré dans le tableau III.14 (Matrice de confusion).

Classes	1(Estimée)	2(Estimée)	3(Estimée)	4(Estimée)	5(Estimée)	6(Estimée)	7(Estimée)	Nombre/classe	taux
1(Réelle)	10	0	0	1	0	0	0	11	90.9 %
2(Réelle)	0	1	1	0	0	0	0	2	50 %
3(Réelle)	0	2	5	0	0	0	0	7	71.42%
4(Réelle)	0	0	0	4	0	0	0	4	100 %
5(Réelle)	0	0	0	0	5	0	0	5	100 %
6(Réelle)	0	0	0	0	0	6	0	6	100 %
7(Réelle)	0	0	0	0	0	0	19	19	100 %

Tableau III.14: Matrice de confusion obtenue par l'application du modèle SVM sur la base de données Poisson.

III.2.4.4.3. Application KRR

L'application du modèle KRR obtenu avec les paramètres optimaux $\lambda = 10^{-7}$ et $\sigma = 10$, sur l'ensemble de test de la base de données Poisson donne un taux de classification égale à **98.14%**. Le taux de classification pour chaque classe est illustré dans le tableau III.15 (Matrice de confusion).

Classes	1(Estimée)	2(Estimée)	3(Estimée)	4(Estimée)	5(Estimée)	6(Estimée)	7(Estimée)	Nombre/classe	taux
1(Réelle)	11	0	0	0	0	0	0	11	100 %
2(Réelle)	0	2	0	0	0	0	0	2	100 %
3(Réelle)	0	1	6	0	0	0	0	7	85.71%
4(Réelle)	0	0	0	4	0	0	0	4	100 %
5(Réelle)	0	0	0	0	5	0	0	5	100 %
6(Réelle)	0	0	0	0	0	6	0	6	100 %
7(Réelle)	0	0	0	0	0	0	19	19	100 %

Tableau III.15: Matrice de confusion obtenue par l'application du modèle KRR sur la base de données Poisson.

Les tableaux III.13, III.14 et III.15 montrent que les meilleurs taux de classification obtenus avec le RBFN et KRR. Le SVM étaient moins performant par rapport aux deux premiers (RBFN, KRR).

III.2.5. Etude comparative

Pour valider les résultats obtenus sur les deux bases de données Fraise et Poivron, nous avons comparé ces résultats avec ceux obtenus dans la littérature en l'occurrence les travaux de Matteo ottavian et al. publié en 2012 dans une revue cotée dans le domaine de la chimie [53].

Les taux de classification pour les bases de données Fraise et Poivron obtenus avec le classificateur RBFN, SVM, KRR et les résultats obtenus avec la méthode des facteurs similaires de la référence [53] sont illustrés dans le tableau III.16 et tableau III.17. Le tableau III.16 montre que le KRR donne les meilleurs résultats par rapport aux autres classificateurs (RBFN, SVM et les facteurs similaires) pour la base de données Fraise.

Stratégie Classe	Taux de classification RBFN	Taux de classification SVM	Taux de classification KRR	Référence [53]
Contrefait 1	99.15 %	99.77 %	100 %	94.6 %
Contrefait 2	97.89 %	99.71 %	99.55 %	90.9 %
Contrefait 3	98.59 %	98.56 %	99.37 %	88.9 %
Réelle 1	98.74 %	99.94 %	100 %	97.7 %
Réelle 2	95.31 %	97.89 %	98.22 %	95.5 %
Réelle 3	97.97 %	98.56 %	99.64 %	95.4 %

Tableau III.16: Comparaison des résultats obtenus pour la base de données Fraise

Stratégie Classe	Taux de classification RBFN	Taux de classification SVM	Taux de classification KRR	Référence [53]
Contrefait rouge	97.16 %	98.32 %	97.60 %	99.10 %
Contrefait vert	97.44 %	98.56 %	98.18 %	98.20 %
Contrefait jaune	100 %	99.11 %	100 %	100 %
Réelle rouge	99.20 %	98.75 %	99.48 %	93.90 %
Réelle vert	97.13 %	97.09 %	98.34 %	99.40 %
Réelle jaune	99.88 %	99.60 %	100 %	100 %

Tableau III.17 : Comparaison des résultats obtenus pour la base de données Poivron.

III.2.6. Réduction du nombre de variables

L'augmentation du nombre de variables dans une base de données implique l'amélioration des performances en classification [25]. Mais ce n'est pas toujours vrai dans la pratique, car ce ne sont pas toutes les variables présentes dans les données de grande dimension qui aident dans la prédiction de classes. De nombreuses variables peuvent être non pertinentes et éventuellement nuisible pour la classification. En outre, la redondance entre les variables est fréquente [54], [26]. La présence de variables inutiles et redondante ralentit l'algorithme d'apprentissage, et le confond en faisant le surapprentissage [54]. En d'autres termes, en éliminant les variables non pertinentes et redondantes on rend la conception du classificateur simple, par conséquent on améliore les performances de sa prévision et son efficacité de calcul [55], [56].

Les données de grande dimension, peuvent être transformées pour réduire les dimensions sans perdre trop d'informations sur les classes [57]. "Malédiction de la dimension" un terme inventé par Bellman [58] en 1961 à décrire les problèmes rencontrés dans l'analyse des données en grande dimension. Il a prouvé que pour estimer efficacement la fonction de prédiction jusqu'à un degré donné de précision, une augmentation de la dimension des données conduit à une croissance exponentielle du nombre d'échantillons dans la base de données. Un phénomène lié à la malédiction de la dimensionnalité était observé par Ruadys et Jain [59] et a été nommé comme le "Pic phénomène." Ils ont constaté que la précision d'un classificateur augmente d'abord avec l'augmentation du nombre de caractéristiques et se rapproche de la valeur optimale, puis commence à diminuer.

Les Figures III.25 et III.26 montrent les spectres des échantillons de l'huile d'olive à deux longueurs d'onde obtenus à partir des 120 spectres de la base de données. Un exemple du phénomène de colinéarité est montré dans la figure III.25 pour deux longueurs d'onde consécutives dans le spectre (12 et 14 nm), les points représentant les échantillons se placent presque exactement sur une droite. Un exemple de deux longueurs d'ondes qui ne sont pas colinéaires est illustré dans la Figure III.26. Pour un écart de 1100 nm entre les longueurs d'onde (1400 et 2500 nm), les points ne sont pas alignés.

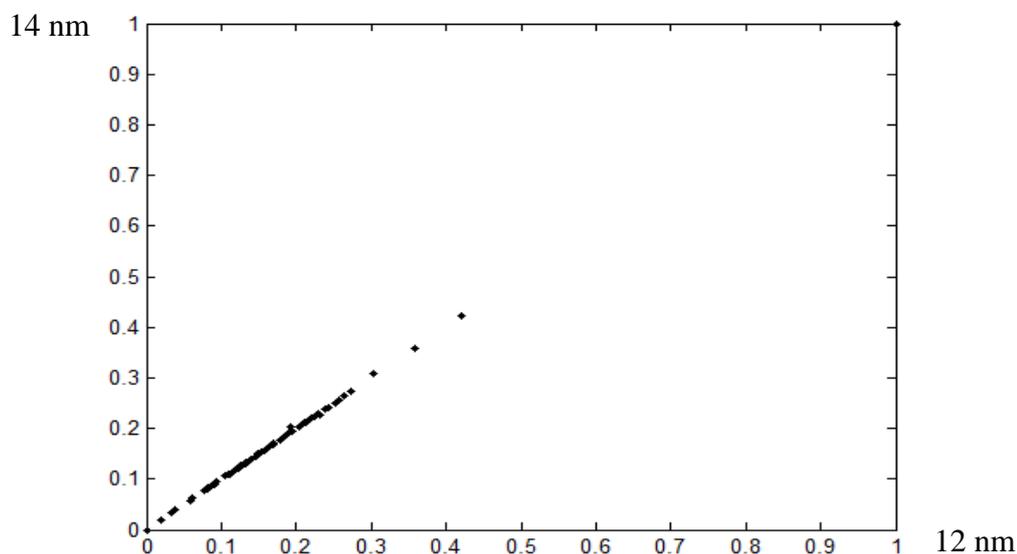


Figure III.25: Exemple des deux longueurs d'onde consécutives 12 et 14 nm qui sont très colinéaires

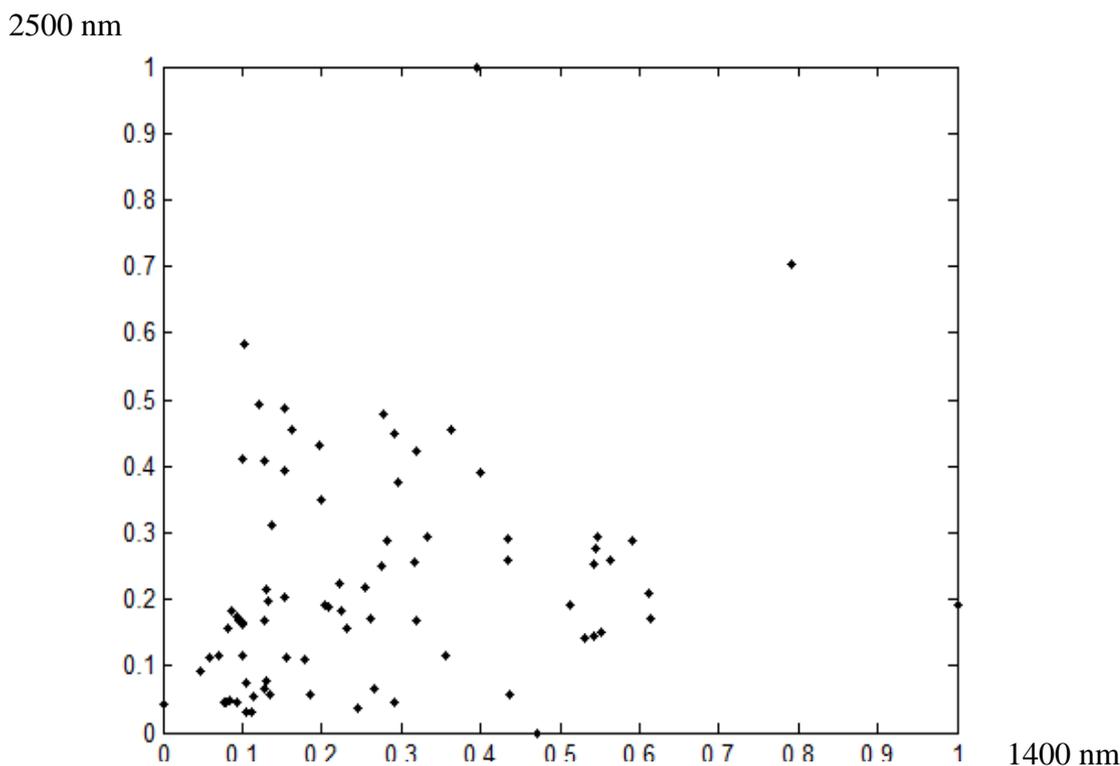


Figure III.26: Exemple des deux longueurs d'onde 1400 et 2500 nm qui ne sont pas colinéaires.

III.2.6.1. Réduction de dimension de la base de données huile d'olive

Dans cette section, nous nous attacherons à l'étude du comportement de chaque classificateur avec le changement de la dimension de l'espace d'entrée. Dans ce travail seulement la base de données huile d'olive est concerné par la réduction de dimension, pour ce faire, nous appliquons l'ACP (Analyse en Composantes Principales) sur l'ensemble d'apprentissage pour l'obtention d'une matrice U de vecteurs propres, on multiplie la matrice de données initiale par la matrice de vecteur propre pour l'obtention d'une nouvelle matrice Y telque chaque colonne de cette matrice représente une composante principale. Nous considérons la matrice Y comme une matrice de données en utilisant seulement deux composantes principales et on fait varier ce nombre jusqu'à la 50^{ème} composante principale et voir par la suite l'évolution du taux de classification.

Les figures III.27, III.28, III.29 montrent le taux de classification en fonction du nombre de composantes principales, et ces figures montrent que le nombre de composantes principales optimal pour les classificateurs RBFN, SVM et KRR est égal à 3.

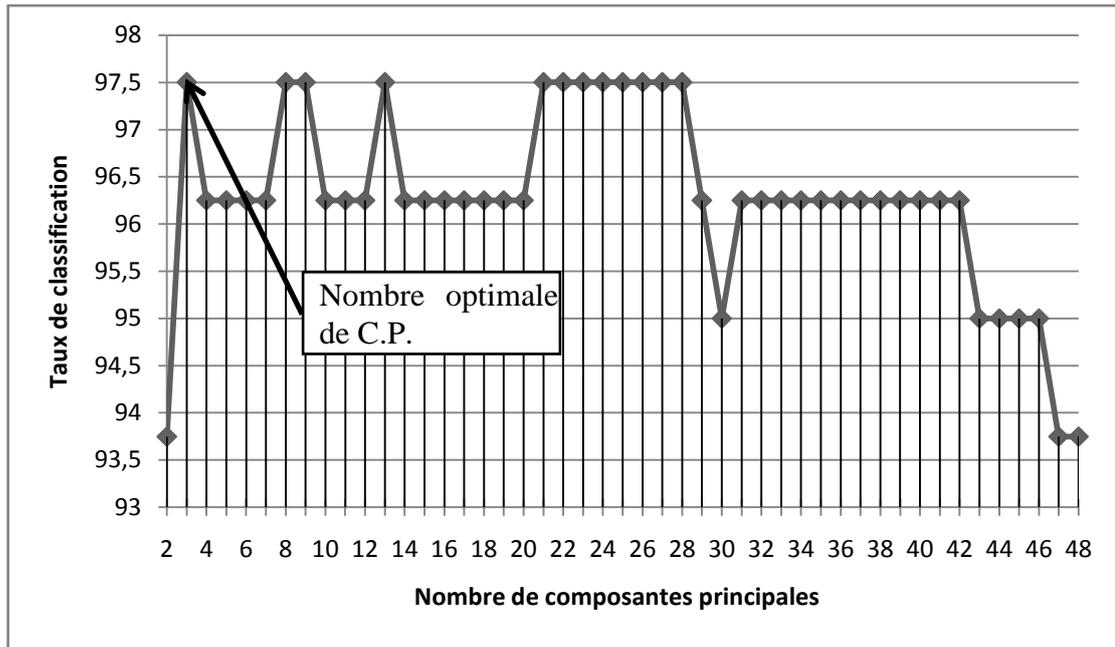


Figure III.27 : Évolution du taux de classification en fonction de nombre de composants principales du RBFN sur la base de données Huile d'olive

L'application du modèle RBFN obtenu avec les paramètres optimaux $\sigma = 0.88$ et $ncc = 16$ neurones, sur l'ensemble de test de la base de données Huile d'olive avec une dimension réduite à 3 donne le taux de classification égale à 95.12 %.

Classes	1 (estimée)	2 (estimée)	3 (estimée)	Nombre/classe	Taux
1 (Réelle)	14	1	0	15	93.33 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	1	0	1	2	50 %

Tableau III.18 : Matrice de confusion obtenue par l'application RBFN sur base de données Huile d'olive réduite.

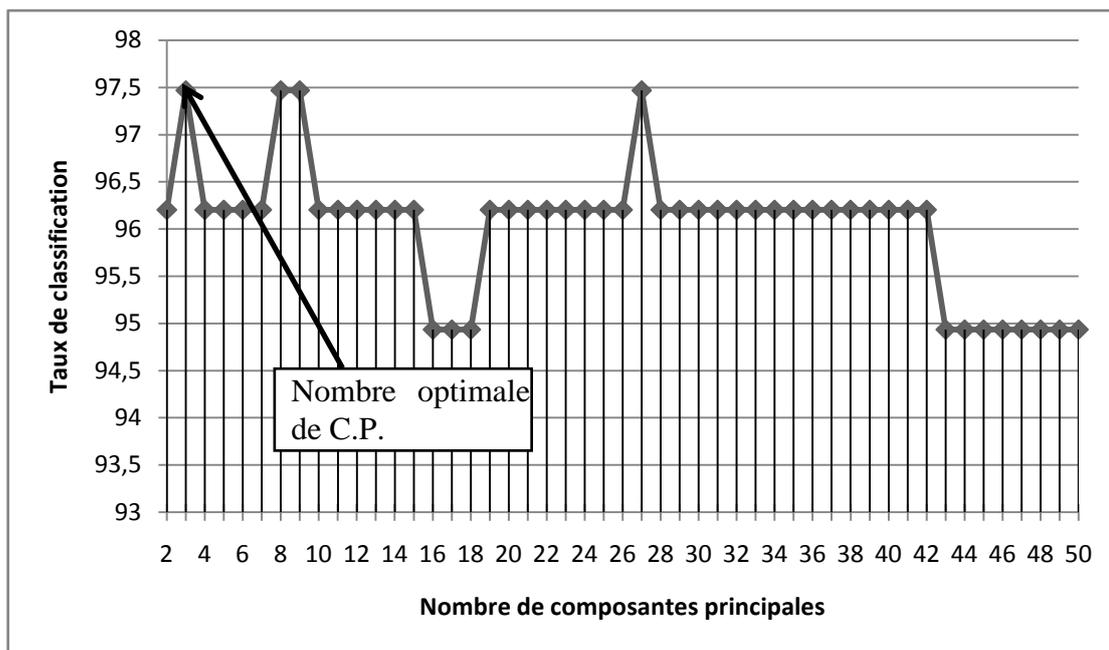


Figure III.28 : Évolution du taux de classification en fonction de nombre de composants principales du SVM sur la base de données Huile d'olive.

L'application du modèle SVM obtenu avec les paramètres optimaux $\sigma = 2.97$ et $C = 29.76$, sur l'ensemble de test de la base de données Huile d'olive avec un dimension réduite à 3 donne le taux de classification égale à 95.12 %.

Classes	1 (estimée)	2 (estimée)	3 (estimée)	Nombre/classe	Taux
1 (Réelle)	14	1	0	15	93.33 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	1	0	1	2	50 %

Tableau III.19 : Matrice de confusion obtenue par l'application SVM sur base de données Huile d'olive réduite.

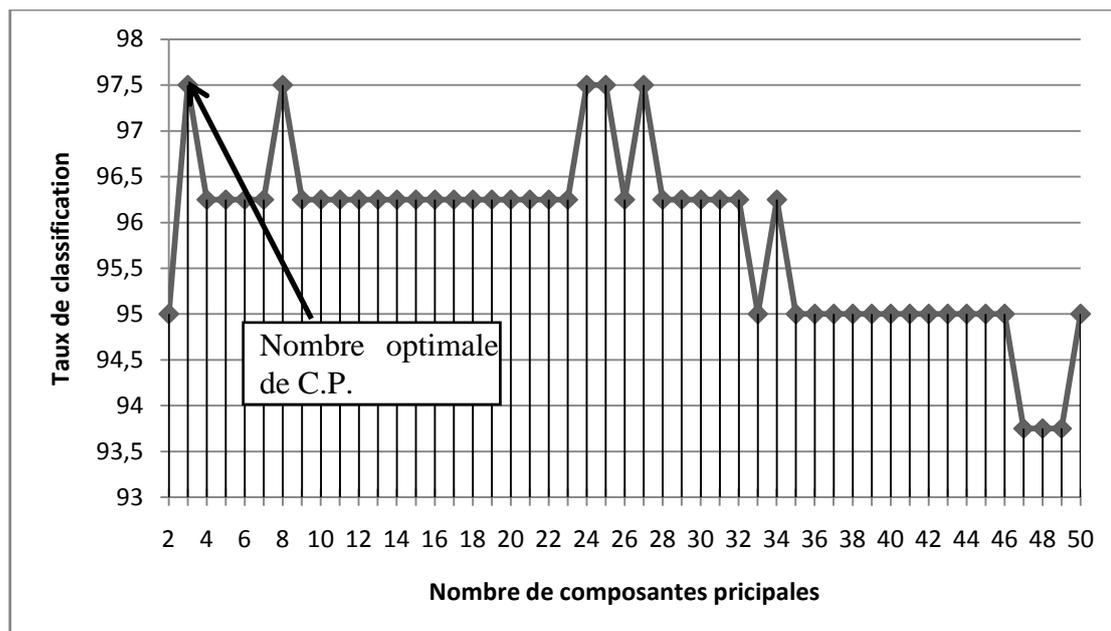


Tableau III.29 : Évolution du taux de classification en fonction de nombre de composantes principales du KRR sur la base de données Huile d'olive.

L'application du modèle KRR obtenu avec les paramètres optimaux $\sigma = 0.48$ et $\lambda = 5.5 \times 10^{-3}$, sur l'ensemble de test de la base de données Huile d'olive avec une dimension réduite à 3 donne un taux de classification égale à 95.12 %

Classes	1 (Estimée)	2 (Estimée)	3 (Estimée)	Nombre/classe	Taux
1 (Réelle)	14	1	0	15	93.33 %
2 (Réelle)	0	24	0	24	100 %
3 (Réelle)	1	0	1	2	50 %

Tableau III.20 : Matrice de confusion obtenue par l'application KRR sur base de données Huile d'olive à dimension réduite

Le taux de classification de la base de données huile d'olive est amélioré après La réduction de la dimension pour les deux classificateur SVM et KRR de 92.68% à 95.12% c'est à dire une amélioration d'un échantillon bien classé par rapport avant l'application de l'ACP, le tableau III.18 montre que l'amélioration du SVM était par un échantillon dans la troisième classe, et pour le KRR l'amélioration était aussi par un échantillon mais cette fois dans la première classe comme le montre le tableau III.19. Par contre le tableau III.17 montre qu'il n'y a aucune amélioration pour le classificateur RBFN.

Stratégie \ Base de données	Huile d'olive	Huile d'olive ACP	Fraise	Poivrons	Poisson
RBFN	95.12 %	95.12 %	97.97 %	98.53 %	100 %
SVM	92.68 %	95.12 %	99.12 %	98.69 %	92.59 %
KRR	92.68 %	95.12 %	99.47 %	98.94 %	98.14 %

Tableau III.21 : résultats de classification de quatre bases de données obtenus avec les trois classificateurs.

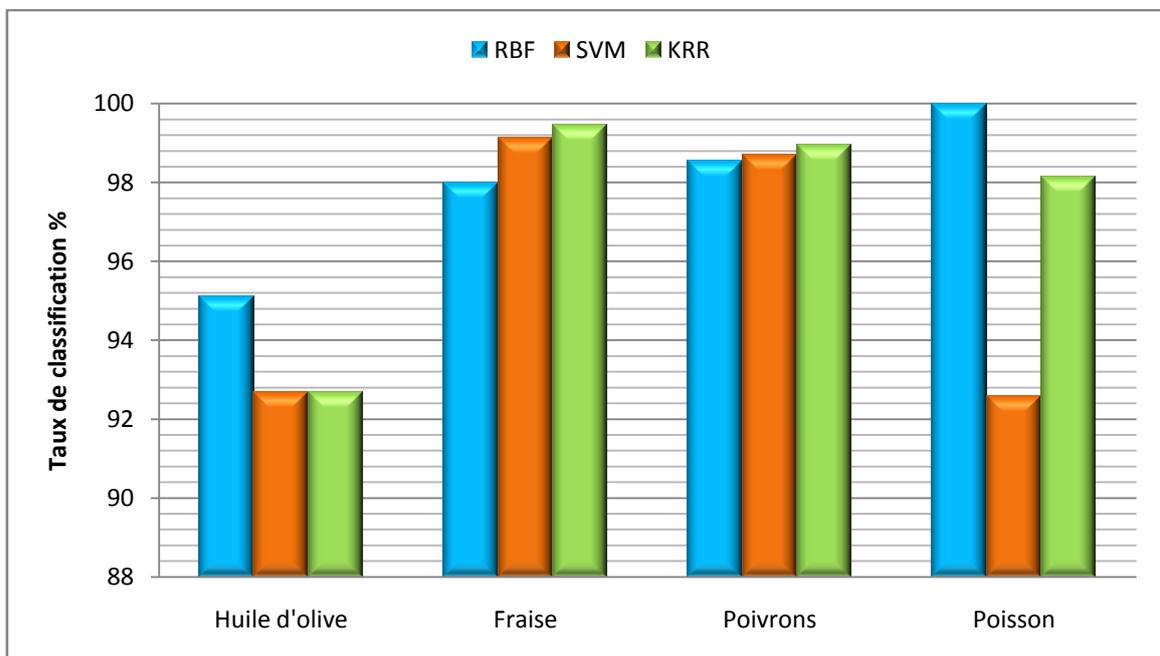


Figure III.30 : Illustration des résultats obtenus avec les trois classificateurs sur les quatre bases de données

III.3. Conclusion

Dans ce chapitre nous avons appliqué les méthodes de classification étudiées dans le chapitre 1 (RBFN, SVM et KRR) sur les bases de données présentées dans le chapitre 2 (Huile d'olive, Fraise, Poivrons et poisson). Par la suite nous avons appliqué une réduction de dimension sur la base de données huile d'olive et nous avons utilisé l'analyse en composantes principales. Les résultats obtenus dans ce chapitre montrent que le RBFN donne les meilleurs taux de classification pour les bases de données huile d'olive et poisson, alors que les meilleurs taux de classification pour les bases de données fraise et poivron sont obtenus par le classificateur KRR.

Conclusion générale

Conclusion générale

L'objectif de ce travail était l'authentification rapide et fiable de produits alimentaires en utilisant les méthodes de l'intelligence artificielle.

En premier temps nous avons appliqué les classificateurs RBFN, SVM et KRR sur les bases de données fournies. D'après les résultats obtenus de l'étude expérimentale nous avons constaté que les classificateurs donnent de meilleurs taux de classification allant de 92% à 100%. Pour les données spectroscopiques (huile d'olive) le RBFN était le meilleur modèle, tandis que le modèle KRR donne les meilleurs taux de classification avec les images multispectrales (Fraise et poivrons). Pour la quatrième base de données (mesures morphologique sur le corps du poisson) les meilleurs taux de classification sont obtenus aussi avec le modèle RBFN.

Pour résoudre le problème de la grande dimension nous avons utilisé l'analyse en composantes principales pour réduire le nombre de variables à l'entrée du classificateur. Les résultats obtenus après la réduction de la dimension sont améliorés pour le modèle KRR et le modèle SVM, par contre le résultat du modèle RBFN reste inchangé même après l'application de la réduction de la dimension. Donc, d'après les résultats le modèle RBFN présente une faible sensibilité à la malédiction de la dimensionnalité.

Les perspectives

Comme perspective à ce travail nous envisageons d'utiliser:

- ❖ Le modèle LWKRR (en anglais: Local Wiegthed Kernel Ridge Regression): la construction d'un modèle de classification local peut améliorer la précision de la classification. Les échantillons voisins qui sont similaires avec un échantillon de test sont sélectionnés pour construire un modèle local car ces derniers ont plus d'influence sur le résultat final de la classification.
- ❖ Stratégie de fusion: des échantillons de la base de données utilisés peuvent donner leurs meilleures taux de classification avec un classificateur, par contre les autres échantillons de la même base donnent leurs meilleurs résultats avec un autre classificateur, donc la fusion des différents classificateurs est indispensable.
- ❖ L'algorithme génétique: l'utilisation de l'algorithme génétique pour l'optimisation des paramètres est rapide et peut améliorer le taux classification.

Bibliographie

Bibliographie

- [1] M. Lees, “*Food Authenticity : Issues and methodologies*”, Eurofins Scientific Laboratories, Nantes, 1999.
- [2] D. Bertrand, & E. Dufour, “*La spectroscopie infrarouge et ses applications analytiques*”, TEC&DOC, Paris, 2006.
- [3] Olivier Fumière and Georges Sinnaeve and Pierre Dardenne, “*L’authentification rapide des poulets de chair sous label : distinction entre poulets issus de souches à croissance lente ou rapide par la spectrométrie dans le proche infrarouge*”, Biotechnologie Agronomie Société et Environnement, vol. 4, pp. 214-220, 2000.
- [4] Isabelle Boutrolle, “*MESURE DE L’APPRECIATION DES ALIMENTS PAR LES CONSOMMATEURS: Etat des pratiques et propositions méthodologiques*”, AgroParisTech, 2007.
- [5] S. Wold, K. Esbensen, and P. Geladi, “*Principal component analysis*”, Chemometrics and intelligent laboratory systems, vol. 2, no. 1, pp. 37–52, 1987.
- [6] L. Aarhus, “*Nonlinear empirical modeling using local pls models*”, Ph.D. dissertation, University of Oslo, 1994.
- [7] N. Darper, H. Smith. “*Applied regression analysis*”. Wiley, New York, 1981.
- [8] R. Gunst and R. Mason. “*Regression analysis and its applications*”. Marcel Dekker, New York, 1980.
- [9] N. Cristianini and J. Shawe-Taylor, “*An introduction to support vector machines and other kernel-based learning methods*”, Cambridge University Press, 2000.
- [10] E. Hoerl and R W. Kennard, “*Ridge Regression: Biased Estimation for Nonorthogonal Problems*”, Technometrics, Vol. 12, no. 1, pp. 55-67, 1970.
- [11] P. Geladi. “*Some recent trends in the calibration literature*”, Chemometrics and Intelligent Laboratory Systems, vol 60, 211-224, 2002.
- [12] N. Benoudjit, “*Variable selection and neural networks: Application in infrared spectroscopy and chemometrics*”, VDM Verlag Dr. Müller Aktiengesellschaft & Co. KG, Dudweiler Landstr. Germany, 2008.
- [13] A. Gusnanto, Y. Pawitan, J. Huang, and B. Lane, “*Variable selection in random calibration of nearinfrared instruments: ridge regression and partial least squares regression settings*”, Journal of Chemometrics, vol. 17, no. 3, pp. 174–185, 2003.

- [14] N. Benoudjit, C. Archambeau, A. Lendasse, J. A. Lee, and M. Verleysen, “*Width optimization of the gaussian kernels in radial basis function networks*”, in ESANN, vol. 2, pp. 425–432, 2002.
- [15] L. Douha, N. Benoudjit, F. Douak, and F. Melgani, “*Support vector regression in spectrophotometry: An experimental study*”, Critical Reviews in Analytical Chemistry, vol. 42, no. 3, pp. 214–219, 2012.
- [16] M. D.BUHMANN, “*Radial Basis Functions: theory and implementations*”, Cambridge University Press, Cambridge, 2003.
- [17] M. J. D.POWELL, “*Radial basis functions for multivariable interpolation : a review*”, Algorithms for approximation, New York, NY, USA, Clarendon Press, p. 143-167, 1987.
- [18] D.BROOMHEAD, D.LOWE, “*Multivariate functional interpolation and adaptative networks*”, Complex Systems, vol. 2, no 3, pp 321-355, 1988.
- [19] J.MOODY, C.DARKEN, “*Fast Learning in Networks of Locally-tuned Processing Units*”, Neural Computation, vol. 1, No. 2, p. 281-294, 1989.
- [20] Vladimir Vapnik, “*The Nature of Statistical Learning Theory*”, Springer Information Science and Statistics, 2000.
- [21] Bernhard Schölkopf, Alexander J. Smola, “*Learning With Kernels: Support Vector Machines, Regularization, Optimization and Beyond*”, MIT Press, 2002.
- [22] V. Vapnik, et S. Kotz, “*Estimation of Dependences Based on Empirical Data*”, Springer Series in Statistics, 1982.
- [23] F. Melgani and L. Bruzzone, “*Classification of hyperspectral Remote Sensing Images With Support Vector Machines*”, Geoscience and Remote Sensing, vol. 42, pp. 1778-1790, 2004.
- [24] R. Steve Gunn, “*Support Vector Machines for Classification and Regression: Technical Report*”, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, University of Southampton, 1998.
- [25] R.O. Duda, P.E. Hart, and D.G. Stork, “*Pattern Classification*”, Wiley, 2001.
- [26] L. Yu and H. Liu, “*Efficient Feature Selection via Analysis of Relevance and Redundancy*”, J. Machine Learning Research, vol. 5, pp. 1205-1224, 2004.
- [27] Corinna Cortes and V. Vapnik, “*Support-Vector Networks, Machine Learning*”, vol. 20, No. 3, pp. 273-297, 1995.
- [28] L. Hamel. “*Knowledge discovery with support vector machines*”. Wiley Edition, 2009.

- [29] A. Shigeo. “*Support Vector Machines for Pattern Classification*”. Springer-Verlag London Limited, 2005.
- [30] V.N. Vapnik. “*Statistical Learning Theory*”. Edition Wiley, 1998.
- [31] S. Knerr, L. Personnaz, J. Dreyfus, et al. “*Single-layer ll learning revisited : A stepwise procedure for building and training a neural network*”. Optimization Methods and Software, vol. 1, pp. 23–34, 1990.
- [32] Amaury Lendasse and Vincent Wertz and Michel Verleysen, “*Model Selection with Cross-Validations and Bootstraps – Application to Time Series Prediction with RBFN Models*”, ICANN/ICONIP 2003, LNCS, vol. 2714, pp. 573–580, 2003.
- [33] D. Chaillou, “ Evaluation d’une méthode analytique de discrimination et d’identification des cafés par spectrophotométrie proche infrarouge”, Spectra Analyse, N° 258, pp.31-38, 2007
- [34] R. H. Wilson, “*Spectroscopic techniques for food analysis*”, VCH, New York, 1994.
- [35] Pavia, D. L., Lampman, G. M. & Kriz, G. S. “*Introduction to spectroscopy*”, Saunders College Publishing, 1996.
- [36] Y. W. Lai, , Kemsley, & R. H. Wilson, “*Quantitative analysis of potential adulterants of extra virgin olive oil using infrared spectroscopy*”. Food Chemistry vol 53, 95-98 1995.
- [37] M. Defernez, E. K. Kemsley, & R. H. Wilson, “*Use of infrared spectroscopy and chemometrics for the authentication of fruit purées*”. J. Agric. Food Chem, vol. 43, pp. 109-113, 1995.
- [38] M. Defernez, & R. H. Wilson, “*Mid-infrared spectroscopy and chemometrics for determining the type of fruit used in jam*”. J Science of. Food and Agriculture, vol. 67, No. 4, 461-467 (1995).
- [39] R. Briandet, E. K. Kemsley & R. H. Wilson, “*Discrimination of arabica and robusta in instant coffee by Fourier transform infrared spectroscopy and chemometrics*”. J. Agric. Food Chem, vol. 44, No. 1, pp. 170-174, 1996.
- [40] E. K. Kemsley, S. Ruault & R. H. Wilson, “*Discrimination between coffea arabica and canephora variant robusta beans using infrared spectroscopy*”. Food Chem, vol. 54, No. 3, pp. 321-326, 1995.
- [41] A. Szydłowska-Czerniak, “*MIR spectroscopy and partial least-squares regression for determination of phospholipids in rapeseed oils at various stages of technological process*”, Food Chem, vol. 105, No. 3, pp. 1179-1187, 2007.
- [42] I. Murray, & I. A. Cowe, “*Advances in Near-Infrared Spectroscopy*”, VCH Publishers, pp. 61-66, 1992.

- [43] D. Bertrand, P. Robert, & W. Loisel, "Identification of some wheat varieties by near infrared reflectance spectroscopy", *J. Sci. Food. Agric.*, vol. 36, No. 11, pp. 1120-1124, 1985.
- [44] A. M. C. Davies, & W. F. McClure, "Near infrared analysis in the Fourier domain with special reference to process control", *Anal. Proc.*, vol. 22, pp. 321-322, 1985.
- [45] T. Sato, "Application of principal component analysis on near-infrared spectroscopic data of vegetable oils for their classification". *JAOCS*, vol. 71, pp. 293-298, 1994.
- [46] D. G. Evans, C. N. G. Scotter, L. Z. Day, & M. N. Hall, "Determination of the authenticity of orange juice by discriminant analysis of near infrared spectra: a study of pre-treatment and transformation of spectral data". *JNIRS*, vol. 1, No. 1, pp. 33-44 1993.
- [47] K. Y. L. Yap, S. Y. Chan, & C. S. Lim, "The reliability of traditional authentication – A case of ginseng misfit". *Food Chem*, vol. 107, No. 1, pp. 570-575, 2008.
- [48] Y. B. Che Man, Z. A. Syahariza, M. E. S. Mirghani, S. Jinap, & J. Bakar, "Analysis of potential lard adulteration in chocolate and chocolate products using Fourier transform infrared spectroscopy", *Food Chem*, vol. 90, No. 4, pp. 815-819, 2005.
- [49] <http://www.models.life.ku.dk/oliveoil>
- [50] http://www.cs.columbia.edu/CAVE/databases/multispectral/zip/strawberries_ms.zip
- [51] http://www.cs.columbia.edu/CAVE/databases/multispectral/zip/peppers_ms.zip
- [52] <https://github.com/thomas-haslwanter/statsintro/blob/master/Data/amstat/fishcatch.dat.txt>
- [53] Matteo Ottavian, Pierantonio Facco, Luca Fasolato, Massimiliano Barolo, "Multispectral data classification using similarity factors", *Chemometrics and Intelligent Laboratory Systems*, vol. 118, pp. 13-23, 2012.
- [54] D. Koller and M. Sahami, "Toward Optimal Feature Selection", *Proceedings 13th International Conference on Machine Learning*, pp. 284-292, 1996.
- [55] M. Dash and H. Liu, "Feature Selection for Classification", *Intelligent Data Analysis*, vol. 1, no. 3, pp. 131-156, 1997.
- [56] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection", *J. Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [57] L. Jimenez and D. Landgrebe, "Supervised Classification in High Dimensional Space: Geometrical, Statistical and Asymptotical Properties of Multivariate Data", *Systems, Man and Cybernetics - Part C*, vol. 28, no. 1, pp. 39-54, 1998.
- [58] R. Bellman, "Adaptive Control Processes: A Guided Tour". Princeton Univ. Press, 1961.

- [59] S. Rudys and A. Jain, “*Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for practitioners*”, *Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252-264, Mar. 1991.

Résumé

L'objectif de ce mémoire est l'authentification rapide et fiable des produits alimentaires. L'authentification de propriétés chimiques contenues dans un produit alimentaire, serait hautement souhaitable pour les fabricants et pour ceux qui impliqués dans la protection de la marque car un produit alimentaire peut être vendu avec un prix élevé alors qu'il est considéré comme produit alimentaire contrefait.

Dans ce travail nous avons appliqué trois classificateurs (RBFN, SVM et KRR) sur quatre bases de données (Huile d'olive, Fraise, Poivron et Poisson). Parmi les méthodes de validation rencontrés dans ce travail nous avons utilisé la méthode k-fold validation, et dans le cas de classification multiclasse la stratégie un contre un a été utilisé.

Pour les bases de données de grande dimension (Huile d'olive) nous avons fait une Réduction de dimension on utilisant l'analyse en composantes principales.

Mots clés :

Authenticité des produits alimentaires, régression ridge à noyau (KRR), réseaux de neurones à fonction de base radiale, machine à vecteur support (SVM), spectroscopie infrarouge.

ملخص

الغرض من هذه المذكرة هو المصادقة السريعة والموثوق بها للمنتجات الغذائية . المصادقة على الخصائص الكيميائية للمنتجات الغذائية، سيكون جد مرغوب فيه من طرف المصنعين و المسؤولين عن حماية العلامة التجارية لأنه يمكن بيع المنتجات الغذائية بسعر مرتفع بينما هي منتجات غذائية رديئة الجودة.

في هذا العمل طبقنا ثلاثة المصنفات (RBFN ، SVM و KRR) على أربعة قواعد بيانات (زيت الزيتون والفراولة والفلفل والسلك) ومن بين أساليب المصادقة التي تطرقنا لها هذا العمل استخدمنا طريقة المصادقة kfold، وفي حالة وجود عدة فئات فإننا نستعمل إستراتيجية التصنيف واحد ضد واحد.

بالنسبة لقواعد المعطيات الكبيرة (زيت الزيتون) فإننا قمنا بتقليص حجمها و ذلك باستعمال L'analyse en : (composantes principales).

الكلمات المفتاحية: المصادقة على المواد الغذائية، المصنف المستند على الانحدار اللوجستي، الشبكات العصبية ذات الدالة المحدودة، آلة المتجه الداعم، التحليل الطيفي.

Abstract

The purpose of this memory is a fast and reliable authentication of food products. The authentication of chemical properties contained in a food product, would be highly desirable for manufacturers and those involved in the protection of the mark because a food product can be sold with a high price when it is considered as fake product.

In this work, we applied three classifiers (RBFN, SVM and KRR) on four databases (olive oil, Strawberry, Pepper and Fish). Among the validation methods encountered in this work we used the k-fold validation method, and in the case of a multiclass classification the one against one strategy was used.

In high dimensional data (olive oil), we made a dimension reduction by using principal component analysis.

Keywords: Food Authenticity, kernel ridge regression (KRR), radial basis function neuronal network, support vector machine (SVM), infrared spectroscopy.