



**HAL**  
open science

# Développement du système d'analyse des données recueillies par les capteurs et choix du groupement de capteurs optimal pour le suivi de la cuisson des aliments dans un four

Thomas Monrousseau

► **To cite this version:**

Thomas Monrousseau. Développement du système d'analyse des données recueillies par les capteurs et choix du groupement de capteurs optimal pour le suivi de la cuisson des aliments dans un four. Automatique / Robotique. INSA de Toulouse, 2016. Français. NNT : 2016ISAT0054 . tel-03510246

**HAL Id: tel-03510246**

**<https://theses.hal.science/tel-03510246>**

Submitted on 4 Jan 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Fédérale



Toulouse Midi-Pyrénées

# THÈSE

En vue de l'obtention du

**DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE**

Délivré par :

*l'Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse)*

---

---

Présentée et soutenue le 22/11/2016 par :

**THOMAS MONROUSSEAU**

Développement du système d'analyse des données recueillies par les capteurs et choix du groupement de capteurs optimal pour le suivi de la cuisson des aliments dans un four

---

---

## JURY

SYLVIE CHARBONNIER	Maitre de Conférences	Rapporteur
MARIE-VÉRONIQUE LE LANN	Professeur d'Université	Directeur de thèse
PHILIPPE THOMAS	Maitre de Conférences	Rapporteur
LOUISE TRAVÉ-MASSUYÈS	Directrice de Recherche	Présidente du Jury
CRISTIAN TRELEA	Professeur d'Université	Membre du Jury
SÉBASTIEN VOLATIER	Chef de projet innovation, groupe SEB	Membre du Jury

---

École doctorale et spécialité :

*EDSYS : Automatique 4200046*

Unité de Recherche :

*Laboratoire d'analyse et d'architecture des systèmes*

Directeur de Thèse :

*Marie-Véronique LE LANN*

Rapporteurs :

*Sylvie CHARBONNIER et Philippe THOMAS*



## Remerciements

Pour commencer, je tiens à remercier ma directrice de thèse, Marie-Véronique Le Lann, qui a su me soutenir pendant les quatre années qu'a duré cette thèse, et plus particulièrement sur la fin, lorsque tous les problèmes s'accumulaient. Je voudrais également remercier l'ensemble des partenaires du projet Opticook avec lesquels j'ai eu l'honneur de travailler, notamment Sébastien Volatier et Jean-Marc Brun qui ont mené ce projet malgré la complexité et les challenges qu'il impliquait. Je remercie aussi Sandrine Isz, avec qui j'ai eu l'honneur de partager de grands moments dans les infrastructures de la SNCF.

Je remercie les différents directeurs du LAAS-CNRS, Jean Arlat et Liviu Nicu, qui m'ont accueilli durant ma thèse.

Merci à tous les membres de l'équipe DISCO du LAAS, qu'ils soient stagiaires, doctorants ou permanents avec lesquels j'ai partagé du temps, des repas, des soirées et bien d'autres moments. Je remercie tout particulièrement Lisa, avec qui j'ai pu partager mon bureau pendant un temps et me rappeler que Willy Denzey existait, ainsi que Quentin et Nathalie qui m'ont aidé à vaincre le côté obscur de Latex.

Merci à Sylvie Charbonnier et Philippe Thomas pour avoir accepté de prendre de leur temps pour rapporter sur cette thèse.

Merci à l'équipe pédagogique du département Génie Electrique et Informatique de l'INSA de Toulouse avec qui j'ai eu l'occasion de travailler en tant que vacataire et en tant qu'ATER. Je remercie spécialement Elodie Chanthery, Germain Garcia, Gwendoline Le Corre, Vincent Mahout et Audine Subias qui ont cru en moi en me confiant leurs étudiants.

Enfin, je remercie ma famille et mes amis pour m'avoir aidé et supporté durant cette thèse. Merci à mes parents, Pascale et François, qui s'inquiètent beaucoup pour moi. Merci à Claire, qui partage ma vie, d'avoir supporté mes quasi-nuits blanches en cette fin de rédaction.

Merci.



# Table des figures

1.1	Représentation schématique du réseau de neurones artificiels présenté dans le brevet de la société Matsushita Electric Industrial Co . . . . .	6
2.1	Limites de classes des données Iris pour un algorithme LDA . . . . .	12
2.2	Limites de classes des données Iris pour un algorithme QDA . . . . .	12
2.3	Limites de classes des données Iris pour un arbre de décisions . . . . .	13
2.4	Limites de classes des données Iris pour un algorithme k-nn avec k=1 . . . . .	14
2.5	Limites de classes des données Iris pour un algorithme k-nn avec k=5 . . . . .	15
2.6	Limites de classes des données Iris pour un algorithme de type SVM linéaire . . . . .	16
2.7	Limites de classes des données Iris pour un algorithme de type SVM quadratique . . . . .	17
2.8	Représentation schématique d'un perceptron . . . . .	18
2.9	Limites de classes pour un réseau de neurones avec dix neurones dans sa couche cachée . . . . .	19
2.10	Limites de classes pour l'algorithme LAMDA utilisant la fonction binomiale . . . . .	23
2.11	Limites de classes pour l'algorithme LAMDA utilisant la fonction binomiale centrée . . . . .	24
2.12	Représentation graphique d'un réseau de neurones artificiels avec trois neurones sur la couche d'entrée, cinq neurones sur la couche cachée et deux neurones sur la couche de sortie . . . . .	30
3.1	Photographie d'un four servant à réaliser les cuissons pour la sélection d'attributs . . . . .	34
3.2	Valeurs moyennes des différentes classes des spectres par fluorescence du cabillaud . . . . .	46
3.3	Valeurs moyennes des différentes classes des spectres par fluorescence du cabillaud avec suppression des fonds continus . . . . .	47
3.4	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du cabillaud après excitation par lumière fluorescente . . . . .	48
3.5	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du saumon après excitation par lumière fluorescente . . . . .	48
3.6	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour de la truite après excitation par lumière fluorescente . . . . .	49
3.7	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rosbif après excitation par lumière fluorescente . . . . .	50
3.8	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de porc après excitation par lumière fluorescente . . . . .	50
3.9	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de veau après excitation par lumière fluorescente . . . . .	51

3.10	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du blanc de poulet après excitation par lumière fluorescente . . .	52
3.11	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour des cuisses de poulet après excitation par lumière fluorescente .	53
3.12	Ratio entre les longueurs d'onde 580 nm et 544 nm pour du rosbif .	54
3.13	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du cabillaud sous lumière blanche . . . . .	55
3.14	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du saumon sous lumière blanche . . . . .	56
3.15	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour de la truite sous lumière blanche . . . . .	56
3.16	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rosbif sous lumière blanche . . . . .	57
3.17	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de porc sous lumière blanche . . . . .	58
3.18	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de veau sous lumière blanche . . . . .	58
3.19	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du blanc de poulet sous lumière blanche . . . . .	59
3.20	Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour des cuisses de poulet sous lumière blanche . . . . .	60
4.1	Fonctions servant de base aux tests des fonctions d'appartenance par intervalles . . . . .	64
4.2	Exemple d'approximation d'une gaussienne par un intervalle triangulaire . . . . .	66
4.3	Schémas des différents cas possibles d'intersection d'intervalles triangulaire . . . . .	67
5.1	Les prototypes de four : à gauche le v2 et à droite le v1 . . . . .	73
5.2	Présentation graphique du modèle à base de couches pour la résolution de l'équation de la chaleur discrétisée . . . . .	75
5.3	Exemple de température recalculée à l'aide de l'équation de la chaleur et d'un r optimal . . . . .	76
5.4	Visualisation du paramètre r en fonction de plusieurs données physiques de rosbif . . . . .	78
5.5	Valeurs des poids de l'algorithme MEMBAS pour la classification de la taille d'un rosbif . . . . .	79
5.6	Valeurs des poids des séparateurs de l'algorithme SVM pour la classification de la taille d'un rosbif . . . . .	80
5.7	Matrices de confusion pour une classification de la taille des aliments avec l'algorithme LAMDA et par SVM linéaire (3200 attributs) . . .	81
5.8	Matrices de confusion pour une classification de la taille des aliments avec l'algorithme LAMDA et par SVM linéaire (3 attributs) . . . . .	81

5.9	Comparaison entre les paramètres $r$ optimaux et ceux calculés par défuzzification . . . . .	82
5.10	Comparaison entre les paramètres $r$ optimaux et ceux calculés par un réseau de neurones à trois entrées . . . . .	83
5.11	Comparaison entre les paramètres $r$ optimaux et ceux calculés par un réseau de neurones à soixante entrées . . . . .	83
5.12	Exemple de mesures réalisées dans le premier prototype v1 . . . . .	85
5.13	Exemple de mesures du capteur olfactif normalisées . . . . .	86
5.14	Exemple de mesures du capteur olfactif normalisées avec une référence dynamique . . . . .	87
5.15	Poids MEMBAS pour les trois types de données olfactives sur le prototype v1 . . . . .	87
5.16	Poids MEMBAS pour l'ensemble des données du prototype v1 . . . . .	88
5.17	Résultats de la classification par LAMDA des données de la cuisson des rosbif . . . . .	89
5.18	Matrice de confusion de la classification avec LAMDA des données rosbif du prototype v1 . . . . .	89
5.19	Matrice de confusion de la classification avec des SVM des données rosbif du prototype v1 . . . . .	90
5.20	Poids de l'algorithme MEMBAS pour la sélection de capteur sur le prototype v1 avec les viandes . . . . .	91
5.21	Poids de l'algorithme MEMBAS pour la sélection de capteur sur le prototype v1 avec les poissons . . . . .	92
5.22	Exemple de données olfactives recueillies sur le prototype v2 . . . . .	93
5.23	Poids MEMBAS des différentes mesures dans le prototype v2 pour du rosbif . . . . .	94
5.24	Poids MEMBAS des différentes mesures dans le prototype v2 pour du saumon . . . . .	94
5.25	Résumé d'informations sur la classification dans le prototype v2 pour du rosbif . . . . .	96
5.26	Résumé d'informations sur la classification dans le prototype v2 pour du saumon . . . . .	97
5.27	Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole . . . . .	98
5.28	Deuxième résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole . . . . .	99
5.29	Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif plus long que celui décrit par le protocole . . . . .	99
5.30	Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole et en présence d'aromates . . . . .	100



5.31 Deuxième résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif plus long et plus épais que celui décrit par le protocole . . . . . 100

C.1 Exemple de résolution de l'équation de la chaleur pour du rosbif avec un paramètre d'équation optimal . . . . . 109

C.2 Exemple de résolution de l'équation de la chaleur pour du saumon avec un paramètre d'équation optimal . . . . . 109

# Liste des tableaux

3.1	Tableau récapitulatif des températures internes des différents aliments pour chaque état de cuisson . . . . .	32
3.2	Masses et dimensions des ingrédients utilisés . . . . .	33
3.3	Nombres d'expériences et de descripteurs pour les analyses de composés organiques volatils . . . . .	36
3.4	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du cabillaud avec la méthode d'extraction de l'ANSES . .	37
3.5	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du cabillaud avec la méthode d'extraction d'Alpha MOS .	37
3.6	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du saumon avec la méthode d'extraction de l'ANSES . . .	38
3.7	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du saumon avec la méthode d'extraction d'Alpha MOS . .	38
3.8	Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la truite avec la méthode d'extraction de l'ANSES . .	39
3.9	Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la truite avec la méthode d'extraction d'Alpha MOS .	39
3.10	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rosbif avec la méthode d'extraction de l'ANSES . . . .	40
3.11	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rosbif avec la méthode d'extraction d'Alpha MOS . . .	40
3.12	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de porc avec la méthode d'extraction de l'ANSES	41
3.13	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de porc avec la méthode d'extraction d'Alpha MOS	41
3.14	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de veau avec la méthode d'extraction de l'ANSES	42
3.15	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de veau avec la méthode d'extraction d'Alpha MOS	42
3.16	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du blanc de poulet avec la méthode d'extraction de l'ANSES	43
3.17	Résultats de la sélection des composés organiques volatils par MEM-BAS pour du blanc de poulet avec la méthode d'extraction d'Alpha MOS . . . . .	43
3.18	Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la cuisse de poulet avec la méthode d'extraction de l'ANSES . . . . .	44
3.19	Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la cuisse de poulet avec la méthode d'extraction d'Alpha MOS . . . . .	44

3.20	Taux de classification en utilisant les composés retenus pour les poissons	45
3.21	Taux de classification en utilisant les composés retenus pour les viandes	45
3.22	Tableau récapitulatif des longueurs d'onde sélectionnées pour l'analyse par fluorescence . . . . .	53
3.23	Taux de classification pour l'analyse par fluorescence avec les cinq longueurs d'onde retenues . . . . .	54
3.24	Tableau récapitulatif des longueurs d'onde sélectionnées pour l'analyse des spectres dans le visible et le proche infrarouge . . . . .	60
3.25	Taux de classification pour l'analyse des spectres dans le visible et le proche infrarouge avec les neuf longueurs d'onde retenues . . . . .	61
3.26	Bilan global de la sélection d'attributs pour le suivi de la cuisson d'aliments . . . . .	62
4.1	Table des résultats de classification pour des données bruitées avec une loi uniforme . . . . .	65
4.2	Table des résultats de classification pour des données bruitées avec une loi normale . . . . .	68
5.1	Tableau récapitulatif des données physiques des expériences de rosbif	77
D.1	Tableau des températures à cœur du rosbif qui délimitent les classes pour la classification dans le prototype v2 . . . . .	111
D.2	Tableau des températures à cœur du saumon qui délimitent les classes pour la classification dans le prototype v2 . . . . .	112

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Mise en contexte : le suivi de cuisson d'aliments à l'aide d'outils numériques</b>	<b>3</b>
1.1 La cuisson d'aliments à base de modèles physiques . . . . .	3
1.2 La cuisson d'aliments à l'aide du traitement de données . . . . .	5
1.3 Présentation du projet Open Food System . . . . .	6
<b>2 Introduction au traitement des données</b>	<b>9</b>
2.1 Algorithmes de classification . . . . .	9
2.1.1 Introduction . . . . .	9
2.1.2 Analyse discriminante linéaire et quadratique . . . . .	11
2.1.3 Arbres de décision . . . . .	11
2.1.4 k-plus proches voisins (k nearest neighbors) . . . . .	13
2.1.5 Les machines à vecteurs de support (support vector machines)	14
2.1.6 Les réseaux de neurones artificiels . . . . .	17
2.1.7 LAMDA (Learning Algorithm for Multivariate Data Analysis)	19
2.2 La sélection d'attributs . . . . .	24
2.2.1 Introduction . . . . .	24
2.2.2 Les méthodes filtrantes . . . . .	25
2.2.3 Les méthodes d'encapsulation . . . . .	26
2.2.4 Les méthodes hybrides . . . . .	27
2.2.5 Les méthodes embarquées . . . . .	27
2.3 Les techniques de régression . . . . .	27
2.3.1 Introduction . . . . .	27
2.3.2 Régression linéaire simple . . . . .	28
2.3.3 Régression linéaire multiple . . . . .	28
2.3.4 Régression polynomiale . . . . .	29
2.3.5 Réseaux de neurones . . . . .	29
<b>3 Sélection d'attributs pour le suivi de cuisson d'aliments à partir de mesures expérimentales</b>	<b>31</b>
3.1 Introduction . . . . .	32
3.2 Matériel et méthodes . . . . .	33
3.3 Études sur les composés organiques volatils . . . . .	34
3.3.1 Cabillaud . . . . .	37
3.3.2 Saumon . . . . .	38
3.3.3 Truite . . . . .	39
3.3.4 Rosbif . . . . .	39
3.3.5 Rôti de porc . . . . .	40

3.3.6	Rôti de veau . . . . .	42
3.3.7	Blanc de poulet . . . . .	42
3.3.8	Cuisse de poulet . . . . .	43
3.3.9	Bilan sur les composés organiques volatils . . . . .	45
3.4	Mesures spectrales suite à une excitation par lumière fluorescente . .	46
3.4.1	Cabillaud . . . . .	47
3.4.2	Saumon . . . . .	48
3.4.3	Truite . . . . .	49
3.4.4	Rosbif . . . . .	49
3.4.5	Rôti de porc . . . . .	49
3.4.6	Rôti de veau . . . . .	51
3.4.7	Blanc de poulet . . . . .	51
3.4.8	Cuisse de poulet . . . . .	52
3.4.9	Bilan sur l'analyse par fluorescence . . . . .	52
3.5	Mesures de réflexion dans le visible et le proche infrarouge . . . . .	54
3.5.1	Cabillaud . . . . .	55
3.5.2	Saumon . . . . .	55
3.5.3	Truite . . . . .	55
3.5.4	Rosbif . . . . .	57
3.5.5	Rôti de porc . . . . .	57
3.5.6	Rôti de veau . . . . .	57
3.5.7	Blanc de poulet . . . . .	59
3.5.8	Cuisse de poulet . . . . .	59
3.5.9	Bilan sur les spectres dans le visible et le proche infrarouge .	59
3.6	Bilan de la sélection d'attributs pour le suivi de l'état de cuisson d'aliments . . . . .	61
<b>4</b>	<b>Modélisation des incertitudes pour la classification par l'algo- rithme LAMDA</b>	<b>63</b>
4.1	Modélisation d'incertitudes à l'aide d'intervalles . . . . .	63
4.1.1	Utilisation d'intervalles non-fous . . . . .	64
4.1.2	Intervalles fous pour la modélisation de lois normales . . . . .	65
4.2	Utilisation d'agrégation pondérée dans la méthode LAMDA . . . . .	69
4.3	Bilan sur la prise en compte d'incertitudes . . . . .	69
<b>5</b>	<b>Reconstitution de l'état au cœur d'aliments pendant une cuisson</b>	<b>71</b>
5.1	Introduction . . . . .	71
5.2	Présentation des prototypes et des protocoles . . . . .	72
5.2.1	Matériel . . . . .	72
5.2.2	Paramètres pour la classification . . . . .	73
5.3	Étude par l'équation de la chaleur . . . . .	74
5.3.1	Présentation du modèle . . . . .	74
5.3.2	Recherche de r à postériori . . . . .	75
5.3.3	Recherche de r à priori . . . . .	77

5.3.4	Conclusion sur la méthode par calcul de l'équation de la chaleur	82
5.4	Techniques de classification pour le suivi de cuisson . . . . .	84
5.4.1	Présentation . . . . .	84
5.4.2	Analyses effectuées sur le prototype v1 première génération .	84
5.4.3	Prototype v1 seconde génération . . . . .	91
5.4.4	Analyses sur le prototype v2 . . . . .	92
5.5	Reconstitution de la température à cœur à l'aide de réseaux de neurones artificiels . . . . .	98
<b>Conclusion</b>		<b>103</b>
<b>A Formulation de l'algorithme MEMBAS</b>		<b>105</b>
<b>B Explications sur la lecture des matrices de confusion</b>		<b>107</b>
<b>C Exemples de résolutions de l'équation de la chaleur avec paramètre optimal</b>		<b>109</b>
<b>D Limites de classes pour la classification du rosbif et du saumon sur le prototype v2</b>		<b>111</b>
<b>Bibliographie</b>		<b>113</b>



# Introduction

La classification se retrouve dans de nombreux domaines, qu'ils soient scientifiques, industriels ou même dans la vie courante. Il est en effet souvent plus simple de chercher à regrouper des choses dans des cases, des ensembles plutôt que de chercher à exprimer tous les éléments rencontrés de manière indépendante. C'est pour cela que les scientifiques ont cherché, par exemple, à regrouper les différentes espèces de la vie terrestre dans des groupes et sous-groupes.

Peut-on unir classification, c'est à dire la répartition en groupes, en catégories, d'objets concrets ou abstraits, de notions, de données, ayant des traits communs, et cuisine? C'est l'objectif de cette thèse et plus globalement du projet industriel et scientifique dans lequel elle s'inscrit : le projet « Opticook ». Ce dernier a pour but de permettre la mise au point de fours domestiques et industriels capables de déterminer automatiquement l'état de cuisson d'un aliment, sans intervention humaine et sans capteur au contact des aliments. Cela passe par plusieurs phases. Notamment des mesures réalisées sur aliment avec des capteurs de laboratoire pour identifier ce qui différencie un aliment peu cuit d'un autre trop cuit, la construction de capteurs embarqués spécifiques et la création d'un logiciel capable d'utiliser les informations issues des capteurs pour suivre l'état de cuisson des aliments.

Durant le projet, le laboratoire auquel nous appartenons, le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) était chargé de deux missions. La première était de trouver dans les mesures effectuées en laboratoire quelles caractéristiques permettent le plus facilement de faire la différence entre deux états de cuisson. La seconde était de mettre en place un logiciel embarqué dans les prototypes des fours qui soit capable de reconstituer l'état de cuisson des aliments.

Un des principaux problèmes dans le suivi de cuisson est que l'état des aliments est caractérisé par la température au cœur de ce dernier. Cette information est mesurée historiquement sur les fours professionnels ou hauts de gamme à l'aide d'une sonde de température plantée dans le produit. Ceci induit des problèmes tels que l'hygiène et la dénaturation de l'aliment. Pour ces raisons, l'objectif est de retrouver ce qui a lieu au cœur de son repas sans mesure par contact.

Cette thèse est décomposée en cinq chapitres. Dans les deux premiers, il est abordé les différentes méthodes pour le suivi de cuissons dans des fours qui existent dans la littérature, puis il est fait une présentation des différentes méthodes de traitement des données qui seront utilisées au cours de cette thèse. Le troisième chapitre aborde les résultats obtenus pour la détermination des éléments physiques et chimiques à suivre pour reconstituer l'état au cœur des aliments. Dans le quatrième chapitre, il est présenté des améliorations pour étendre un algorithme de classification créé au sein du LAAS afin de permettre la prise en compte d'incertitudes



sur les données utilisées. Enfin, le cinquième et dernier chapitre décrit les méthodes utilisées lors d'expérimentations effectuées en temps réel sur les deux prototypes de four et les résultats associés. Ceci passe notamment par l'application de l'équation de la chaleur, d'algorithmes de classification supervisée et de la régression par réseaux de neurones.

# Mise en contexte : le suivi de cuisson d'aliments à l'aide d'outils numériques

---

## Sommaire

<b>1.1</b>	<b>La cuisson d'aliments à base de modèles physiques . . . . .</b>	<b>3</b>
<b>1.2</b>	<b>La cuisson d'aliments à l'aide du traitement de données . .</b>	<b>5</b>
<b>1.3</b>	<b>Présentation du projet Open Food System . . . . .</b>	<b>6</b>

---

Cela fait plusieurs années qu'un certain nombre de domaines de la science s'intéressent à la cuisine. Aujourd'hui on retrouve à la disposition du public des fourchettes intelligentes pour suivre les habitudes culinaires de l'utilisateur, des réfrigérateurs intelligents qui commandent la nourriture automatiquement ou encore des outils pour mesurer la fraîcheur des œufs. Dans ce chapitre, il sera introduit un état de l'art sur le suivi d'état de cuisson à l'aide de modèles physiques et à l'aide de modèles à base de données. Finalement, le projet qui sert de contexte à la présente thèse sera présenté dans une dernière partie.

## 1.1 La cuisson d'aliments à base de modèles physiques

Des études pour quantifier les valeurs de cuisson d'aliments existent depuis plusieurs décennies. En ce qui concerne les fours, un des premiers modèles est apparu en 1986, introduit par [Holtz 1986]. Dans cet article, Eva Holtz et Christina Skjöldebrand présentent un modèle de simulation de l'évolution de la température à cœur d'un pain de viande de bœuf, avec des ingrédients, dans un four à convection. Les équations du modèle, trois équations différentielles, sont composées de calculs de convection, pour déterminer le flux de température transféré en surface de la viande, et des équations de conduction pour évaluer la température au cœur de l'aliment. Le profil de température au sein de la viande est évalué par l'intégration des équations de conduction de Fourier à l'aide de différences finies. Les différentes propriétés physiques telles que la conductivité thermique, la masse volumique ou la capacité calorifique sont fonction de la température de la viande, du pourcentage d'humidité et du pourcentage de graisse et sont déterminées à l'aide de mesures issues de travaux précédents. Les résultats expérimentaux pour prédire la tempé-

rature en surface, celle à 8 mm de la surface et au centre montrent de bonnes prédictions en ce qui concerne la surface et la température à 8 mm sous la croûte. Toutefois, les résultats au cœur de la viande ne sont pas précis, avec des écarts de près de 20°C. Les auteurs imputent cette différence au modèle d'évolution de la conductivité au cours de la cuisson et de l'absence de prise en compte de certaines propriétés. De plus, les auteurs mentionnent la nécessité d'inclure des équations de bilan de matière modélisant en particulier l'évaporation.

Un modèle prenant en compte l'évaporation a été introduit dans un premier article écrit par Sandro M. Goñi et Viviana O. Salvadori en 2010 [Goñi 2010]. Cet article introduit une loi mathématique qui traduit la capacité de la viande à céder de l'eau. Cette loi suit la forme d'une fonction de type sigmoïde utilisant comme variable d'entrée la température de la viande, dont les paramètres sont déterminés à l'aide d'identifications paramétriques basées sur des expériences réalisées spécifiquement à cet effet. Le modèle inclut également un terme de variation de la température en surface par radiation et donc inclut un coefficient d'émissivité de la viande de bœuf. Les résultats sur la prédiction du temps de cuisson pour atteindre une température à cœur donnée sont relativement précis, avec moins de 8 % d'erreur. Néanmoins, ce modèle nécessite une résolution complexe à l'aide d'un modèle temporel en trois dimensions. Un modèle plus complexe incluant de manière explicite une équation d'évolution de la composition en eau d'un point à l'intérieur d'un rôti a été proposé par Hilal Isleroglu et Figen Kaymak-Ertekin en 2016 [Isleroglu 2016]. Dans cet article, la cuisson se fait par convection avec la vapeur. Il permet de calculer le profil d'humidité dans le rôti en fonction du temps. Le calcul est fait par résolution d'éléments finis type MESH 3D avec un découpage plus fin dans les zones en pourtour du rôti. La conductivité thermique est une fonction du pourcentage d'eau contenue dans le rôti initial et de la température. Les résultats obtenus sont meilleurs que ceux de l'article proposé par Eva Holtz et Christina Skjöldebrand, mais il subsiste encore des différences importantes. En particulier, dans le cas de températures de four basses (moins de 200°C), ces résultats sont même moins bons que ceux obtenus par Sandro M. Goñi et Viviana O. Salvadori en 2010 avec un modèle approché pour décrire la disparition de l'eau dans la matrice.

Concernant la recherche de paramètres physiques d'aliments pour la mise en place de modèles, PS Sheridan et NC.C. Shilton ont publié en 2002 [Sheridan 2002] un article décrivant un moyen de déterminer le coefficient de diffusivité thermique d'un morceau de bœuf de type steak haché lors d'une cuisson par radiation dans le lointain infra-rouge. Le modèle utilisé est basé sur la résolution de l'équation de conduction de la chaleur dans une seule direction, de la surface vers le cœur, en supposant que la température est identique en tout point de la surface et que le matériau est homogène. La diffusivité thermique  $d$  est alors calculée de la manière suivante :  $d = \frac{\Delta T}{\Delta t} / \frac{\Delta T}{\Delta x}$ , avec  $\Delta T$  l'écart en température entre deux points séparés par une distance  $\Delta x$  et  $t$  représentant le temps à un instant. En remarquant que l'évolution de la mesure de température en surface et de la température à cœur en fonction du temps pouvaient être modélisées par des fonctions logarithmiques par morceaux, ils ont pu déterminer la valeur de la diffusivité thermique pour 5 points

entre 6 et 18 minutes, régulièrement espacés de 2 minutes, le tout étant répété pour plusieurs taux de graisse dans la viande. Les résultats montrent que la diffusivité évolue avec le temps et que le type d'évolution dépend du pourcentage de graisse. Un second article fait suite, toujours en 2002, dans lequel les auteurs ajoutent le fait que la diffusivité thermique est fonction du pourcentage de graisse au sein de la viande [Shilton 2002].

Au niveau industriel, la société Matsushita Electric Industrial Co pose un brevet en avril 1999 sur un four à micro-ondes [Tomohiro 1999]. Les auteurs proposent une méthode pour calculer la température au cœur d'un aliment, ce dernier étant considéré comme un parallélépipède rectangle dont toutes les faces sont homogènes en température. Le modèle prend la forme :

$$\Delta T = K_f E e^{-\alpha x} \quad (1.1)$$

avec :  $\Delta T$ , la différence de température entre la surface et un point à une distance  $x$  de la surface,  $K_f$  un coefficient spécifique au type d'aliment pris dans une table de valeurs,  $E$  une constante proportionnelle à l'énergie fournie en surface et  $\alpha$  une constante similaire à la conductivité thermique.

L'équation est appliquée aux six faces du pavé et une fois l'ensemble des résultats mixés, l'algorithme permet de connaître l'incrément de température en un point donné. Ceci permet de connaître l'énergie à fournir pour obtenir une température cible en un point. Il est toutefois difficile de juger la qualité des résultats ainsi obtenus car aucun n'est fourni dans le document.

## 1.2 La cuisson d'aliments à l'aide du traitement de données

En plus de méthodes basées sur la physique, des techniques utilisant des données expérimentales ont été introduites dans le passé. Avant de publier le brevet [Tomohiro 1999] présenté dans la section précédente, la société Matsushita Electric Industrial Co présente un brevet en février 1995 pour le contrôle des cuissons dans un four [Nishii 1995]. Le modèle utilisé est un réseau de neurones artificiels (voir sous-section 2.1.6) capable de calculer la température en surface et au cœur des aliments. Les variables d'entrée du réseau sont la température dans le four à l'instant du calcul et une minute avant, le temps écoulé depuis le début de la cuisson et le type d'aliment, représenté par un nombre entier (figure 1.1). Les aliments gérés par ce brevet sont principalement des poissons sous forme de filets. Certains résultats sont présentés mais il n'est pas possible de déterminer les conditions de réalisation de ces expériences. On note toutefois que la technique des réseaux de neurones a été abandonnée dans le brevet qui a suivi.

Plus récemment, un four nommé « June » a été pré-commercialisé [June]. Ce four se présente comme un produit haute technologie équipé d'une caméra haute définition pour transmettre l'image du produit dans le four sur un téléphone. Toutefois, pour réaliser le contrôle de la température à cœur, une sonde est utilisée pour

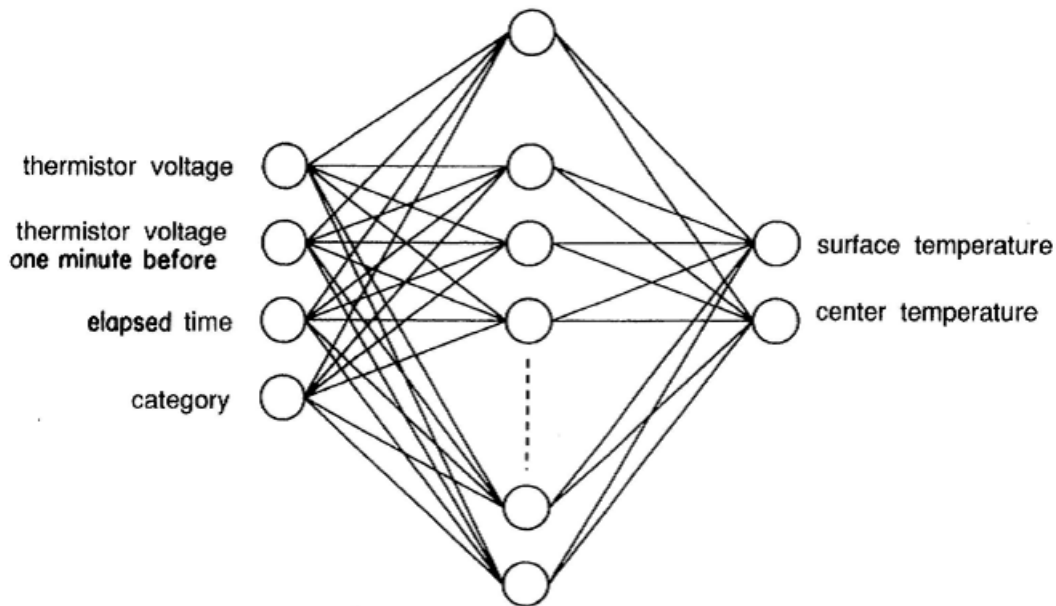


FIGURE 1.1 – Représentation schématique du réseau de neurones artificiels présenté dans le brevet de la société Matsushita Electric Industrial Co

venir mesurer directement la température.

Le projet dans lequel se base la présente thèse a pour objectif de réaliser un suivi d'état à cœur sans avoir recours à une sonde de température.

### 1.3 Présentation du projet Open Food System

Cette thèse se place dans le cadre d'un projet industriel du nom d' « Open Food System » (OFS). L'objectif global de ce projet est de créer un environnement numérique pour le cuisinier, en mettant à disposition des recettes numériques, des services ou par la création d'appareils de cuisine connectés et intelligents. Ce projet est porté par le groupe SEB autour duquel gravitent vingt-cinq partenaires privés et publics, dont deux entreprises, six PME technologiques, quinze laboratoires de recherche et une association. Deux axes de travail majeurs sont présents dans ce cadre, le programme « Nos recettes » et le programme « Opticook ».

« Nos recettes » comporte différents travaux sur l'intégration du numérique dans la cuisine quotidienne. Ceci inclut la définition d'un ensemble de recettes numériques, la compréhension par des algorithmes du contenu des recettes et l'intégration des points précédents dans un environnement complet avec plusieurs appareils capable de communiquer. Un des grands développements de cette partie porte sur la création d'un système de conseil personnalisé de recettes relié à une base de données et capable d'interpréter des mots clés pour agir dynamiquement sur la recette.



« Opticook » est un projet de développement d'appareils de cuisson capables de suivre de manière automatique la cuisson d'un aliment, sans capteur au contact et sans intervention humaine. C'est au sein de ce projet que s'est déroulée cette thèse. À ce projet ont participé : Alpha MOS, une entreprise Toulousaine spécialisée dans la conception de capteurs olfactifs, l'Agence Nationale de Sécurité Sanitaire de l'Alimentation, de l'Environnement et du Travail (ANSES), Bonnet Thirode Grande Cuisine (BTGC), fabricant de matériel de grande cuisine pour les professionnels, l'institut Franche-Comté Electronique, Mécanique, Thermique et Optique - Sciences et Technologies (FEMTO-ST), l'Institut d'Electronique, Micro-électronique et de Nanotechnologie et le Laboratoire d'Automatique, Génie Informatique et Signal (IEMN-LAGIS), spécialisés dans le traitement d'informations thermiques, le Laboratoire Informatique, Electronique et Vision (LE2I), responsables des capteurs visuels, l'unité de recherche Procédés Alimentaires et Microbiologiques (PAM), rattachée à l'étude des réactions physico-chimiques durant la cuisson, le groupe SEB, chef de file du projet et enfin le Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), chargé du traitement des données issues des expériences.

Aujourd'hui, la cuisson des aliments se fait principalement de manière empirique. Le cuisinier utilise l'expérience pour déterminer l'état d'un plat à partir de trois paramètres principaux : l'odeur, la texture et la couleur. Les professionnels ont néanmoins depuis plusieurs années des outils qui permettent de suivre la cuisson à l'aide de mesures chiffrées, comme notamment la sonde de cuisson à cœur qui permet de mesurer la température au cœur des aliments. L'objectif est donc de fournir aux utilisateurs un moyen de contrôler la cuisson des viandes et des poissons sans capteur au contact ni intervention humaine, les capteurs au contact des produits pouvant

causer des problèmes d'hygiène ou de déstructuration du produit. Au moment où ce projet fut lancé, il n'existait aucun appareil dans le commerce capable de remplir ces objectifs.

Ces objectifs passent par quatre grandes étapes :

- Identifier les paramètres de mesures à suivre durant la cuisson pour identifier différents niveaux de qualité des aliments
- Définir des nouveaux barèmes de cuisson en fonction des attentes des consommateurs et des utilisateurs
- Développer des capteurs capables de suivre les paramètres identifiés dans la première étape avec des contraintes de coût et pouvant être intégrés dans un four (contraintes de taille et de températures)
- Produire des appareils de cuisson automatisés

# Introduction au traitement des données

---

## Sommaire

---

<b>2.1 Algorithmes de classification . . . . .</b>	<b>9</b>
2.1.1 Introduction . . . . .	9
2.1.2 Analyse discriminante linéaire et quadratique . . . . .	11
2.1.3 Arbres de décision . . . . .	11
2.1.4 k-plus proches voisins (k nearest neighbors) . . . . .	13
2.1.5 Les machines à vecteurs de support (support vector machines)	14
2.1.6 Les réseaux de neurones artificiels . . . . .	17
2.1.7 LAMDA (Learning Algorithm for Multivariate Data Analysis)	19
<b>2.2 La sélection d'attributs . . . . .</b>	<b>24</b>
2.2.1 Introduction . . . . .	24
2.2.2 Les méthodes filtrantes . . . . .	25
2.2.3 Les méthodes d'encapsulation . . . . .	26
2.2.4 Les méthodes hybrides . . . . .	27
2.2.5 Les méthodes embarquées . . . . .	27
<b>2.3 Les techniques de régression . . . . .</b>	<b>27</b>
2.3.1 Introduction . . . . .	27
2.3.2 Régression linéaire simple . . . . .	28
2.3.3 Régression linéaire multiple . . . . .	28
2.3.4 Régression polynomiale . . . . .	29
2.3.5 Réseaux de neurones . . . . .	29

---

## 2.1 Algorithmes de classification

### 2.1.1 Introduction

Les algorithmes de classification supervisée sont un ensemble de méthodes qui ont pour objectif d'attribuer à un ensemble de données (très souvent sous la forme d'un vecteur de données), un sous-ensemble de l'espace appelé classe. Au départ, l'algorithme utilise une base d'apprentissage contenant des données labélisées pour construire un modèle propre à chaque algorithme. Une fois l'apprentissage réalisé, l'algorithme reçoit de nouvelles données mais dont la classe est inconnue et il doit



alors déterminer à quelle classe appartient cette nouvelle donnée, c'est l'étape de reconnaissance.

Avec l'arrivée massive de l'informatique et des systèmes d'information, ce type d'algorithme est de plus en plus utilisé et ce, parmi un vaste panel d'applications. Ces techniques sont par exemple employées en médecine pour résoudre notamment des problèmes de diagnostic, en finance, en agriculture ou encore en contrôle de l'habitat. Ces méthodes sont également très utiles depuis l'avènement du « Big Data » [Rawat 2014], qui consiste à analyser de très grandes quantités de données par des méthodes de classification, de partitionnement ou encore de sélection d'attributs.

Lors de la phase d'apprentissage, il est généralement nécessaire de réaliser plusieurs opérations. Si la base d'apprentissage comporte des données numériques, elles doivent être normalisées pour éviter les problèmes d'échelles. L'ensemble d'arrivée dépend de l'algorithme utilisé, généralement les données sont placées entre 0 et 1, mais ce n'est pas une règle absolue. Il convient également de séparer les données entre celles qui seront utilisées pour réaliser l'apprentissage et celles qui serviront à tester l'algorithme. L'idéal est d'avoir plusieurs jeux de données à disposition mais ce n'est pas toujours possible. Dans le cas où l'utilisateur ne dispose que d'un seul ensemble de données, il existe des techniques pour permettre cette séparation. Les deux plus utilisées sont les suivantes :

- La validation croisée : l'échantillon est divisé en  $k$  sous-échantillons de tailles homogènes choisis aléatoirement, puis l'apprentissage se fait sur  $k - 1$  sous-échantillons, le dernier étant utilisé pour tester l'algorithme. L'opération est répétée  $k$  fois afin que chaque sous-échantillon soit utilisé en phase de test.
- La mise à l'écart d'un échantillon ou « Leave one out » : il s'agit d'un cas particulier de la validation croisée utilisé dans le cas où l'utilisateur dispose d'un nombre d'échantillons  $N$  petit. Il s'agit alors de réaliser l'entraînement avec  $N - 1$  échantillons et de tester avec celui qui n'a pas été utilisé. L'opération est réalisée  $N$  fois.

Une autre chose à prendre en compte lors de l'apprentissage est la répartition des données dans chaque classe. Pour être optimal, il doit y avoir un nombre équivalent d'échantillons dans toutes les classes, si ce n'est pas le cas, il est conseillé de réduire le nombre d'échantillons des classes qui en possèdent le plus.

Dans cette section, les algorithmes de classification les plus courants sont présentés. Pour chaque cas, un exemple basé sur le jeu de données des Iris [Fisher 1936] illustrera la manière dont l'algorithme sépare les classes. La base de données des Iris est une base très utilisée depuis des années pour tester les algorithmes de classification. Elle comporte un nombre de 150 échantillons repartis en trois classes (50 par classe) ainsi que quatre attributs : la longueur des sépales, la largeur des sépales, la longueur des pétales et la largeur des pétales. Ici pour permettre une représentation graphique, seules les informations sur les pétales seront conservées. Sur chaque figure, les Iris setosa (parfois appelés classe 1) sont en bleu, les Iris versicolor (parfois appelés classe 2) en rouge et les Iris virginica (parfois appelés classe 3) en jaune.

### 2.1.2 Analyse discriminante linéaire et quadratique

L'analyse discriminante linéaire (LDA, aussi appelée LDA de Fisher) est une méthode de base en classification qui consiste à utiliser un hyperplan pour séparer les données qui appartiennent à des classes différentes. Dans le cas où le nombre de classes est égal à deux, un seul hyperplan est utilisé et le résultat de la classification est donné par le côté de l'hyperplan dans lequel se situe la donnée à classer. Cette méthode fait l'hypothèse que les données sont distribuées selon une loi normale et que la covariance de chacune des classes est identique. L'hyperplan est alors construit de manière à maximiser la distance entre l'hyperplan et la moyenne des classes tout en minimisant la variance à l'intérieur des classes.

Le cas de l'analyse discriminante quadratique (QDA) reprend l'hypothèse de distribution normale présente dans la LDA mais ne suppose pas que la covariance est identique dans chaque classe. Le cas quadratique propose une séparation des classes non plus à l'aide d'hyperplan mais avec une surface plus complexe définie comme il suit où  $x$  est le vecteur de données à classer et  $A$ ,  $B$  et  $C$  sont les matrices des coefficients qui définissent la surface de séparation :

$$x^T Ax + Bx + C = 0 \quad (2.1)$$

En pratique il existe deux manières de procéder pour construire une QDA [Carraro 2007], la première est de calculer les coefficients directement par un calcul complet des termes dans la version quadratique, la seconde en augmentant la dimension de l'espace des attributs, en y ajoutant notamment le carré des variables puis de procéder à une LDA. Ces deux méthodes donnent des résultats proches en termes de performance. Nous donnons en figure 2.1 et 2.2 les résultats de séparations entre les classes obtenus respectivement par ces deux méthodes dans le cas de la base de données Iris.

### 2.1.3 Arbres de décision

Les arbres de décision sont une méthode de classification qui partitionne les données de manière hiérarchique sous forme d'un arbre dont les feuilles sont les différentes classes [Lee 2014]. A chaque niveau, un critère de séparation est utilisé pour déterminer dans quelle branche va être placée la donnée en cours de classification. Il est important lors de la construction de l'arbre de bien choisir les critères de séparation afin de rendre le traitement de données plus efficace et de diminuer le nombre de feuilles. Ces critères peuvent par ailleurs aussi bien concerner un seul attribut de la donnée à classer que plusieurs de ces attributs simultanément.

Afin de séparer au mieux les données à chaque nœud, des fonctions mathématiques telles que le coefficient de Gini ou la mesure d'entropie sont utilisées pour quantifier le pouvoir de discrimination du nœud. Pour ces deux fonctions, un minimum est recherché pour garantir la meilleure séparation des données.

Les arbres de décision sont intéressants pour leur capacité à pouvoir manipuler différents types de variables, par le fait qu'ils soient facilement compréhensibles et

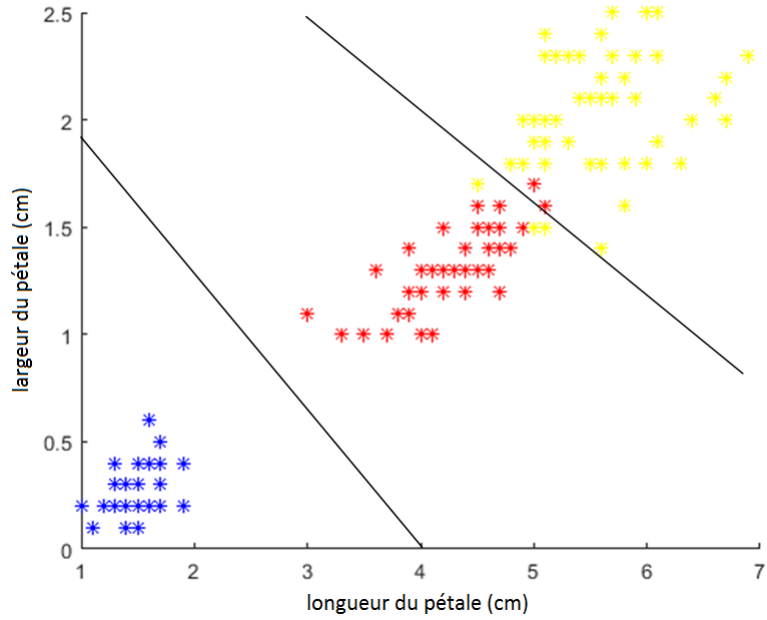


FIGURE 2.1 – Limites de classes des données Iris pour un algorithme LDA

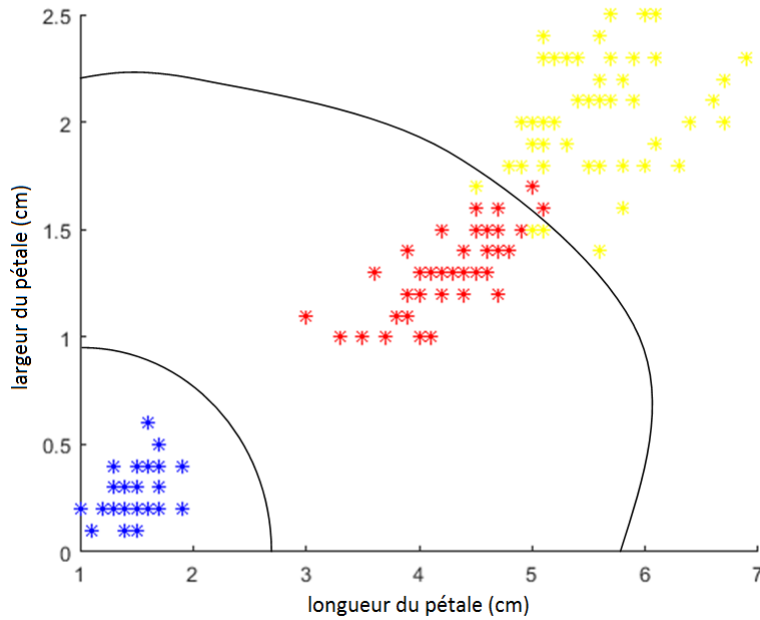


FIGURE 2.2 – Limites de classes des données Iris pour un algorithme QDA

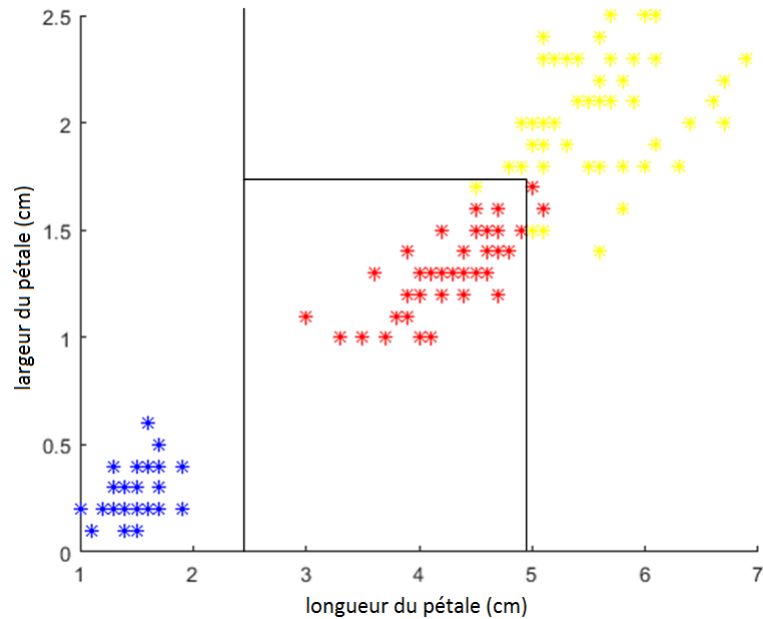


FIGURE 2.3 – Limites de classes des données Iris pour un arbre de décisions

puissent être construits à partir d’un modèle de connaissances et pas uniquement à partir de données. Leur principal défaut est la difficulté potentielle à construire l’arbre dans des cas difficiles, notamment pour des critères multivariables ou en présence de fortes incertitudes sur les données. Les séparations obtenues par l’application d’une telle méthode sont montrées en figure 2.3.

#### 2.1.4 k-plus proches voisins (k nearest neighbors)

La technique des k-plus proches voisins est une des techniques les plus classiques en classification. Lors de l’introduction d’un échantillon à tester, l’algorithme calcule la distance à chaque point de la base d’apprentissage et conserve les  $k$  plus proches [Cunningham 2007], avec  $k > 0$ . La classe attribuée à l’échantillon est généralement celle qui est majoritaire parmi les  $k$  retenus, le cas trivial étant pour  $k = 1$  où la classe choisie est alors celle du point le plus proche.

Les deux principaux paramètres de cet algorithme sont le choix de  $k$  et la fonction de mesure de la distance. Un  $k$  trop grand va introduire parmi les proches voisins des candidats qui n’appartiennent pas à la classe de l’échantillon à classer et à l’inverse un  $k$  trop petit va ignorer certains points similaires à l’échantillon. Un  $k$  optimal dépend principalement de la taille de la base d’apprentissage et de la dispersion des points dans celle-ci [Enas 1986]. Il est également conseillé de prendre un  $k$  impair pour éviter les égalités lors de l’assignation de la classe à l’échantillon. Le second paramètre à déterminer, la fonction de distance, est très généralement un cas particulier de la distance de Minkowski (équation 2.2) avec  $q = 2$  et  $w_i = 1$  pour tout  $i$  (norme euclidienne).

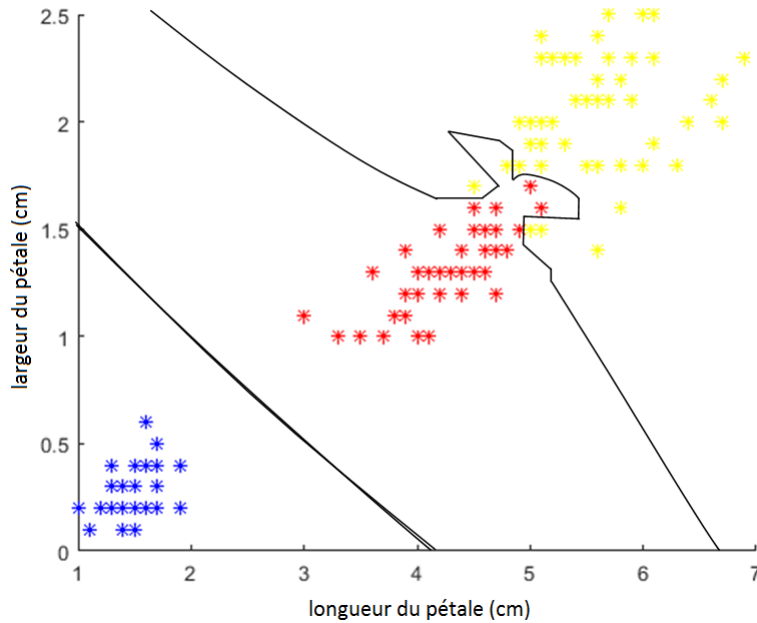


FIGURE 2.4 – Limites de classes des données Iris pour un algorithme k-nn avec k=1

$$d(X, Y) = \sqrt[q]{\sum_{i=1}^I w_i |x_i - y_i|^q} \quad (2.2)$$

avec  $X = \{x_1, \dots, x_I\}$  et  $Y = \{y_1, \dots, y_I\}$ ,  $I$  étant le nombre de descripteurs de chacun des individus à classer.

Il est néanmoins possible d'introduire d'autres mesures de distance selon les cas voire d'utiliser des fonctions pour manipuler des variables qui ne soient pas numériques. La principale force de cette méthode est qu'il n'y a pas besoin de construire un modèle à partir de la base d'apprentissage, le modèle étant la base elle-même (on parle dans ce cas de « lazy learning »). Ceci pose toutefois des problèmes de stockage en mémoire puisqu'il est nécessaire de retenir l'ensemble des points, ainsi que de temps de calcul des voisins qui peut être long sur des bases de données de taille élevée. Une réponse possible à ces problèmes est une réduction du nombre de points en agglomérant des zones de la base d'apprentissage originale. Les limites formées par les différentes classes de la base de données Iris sont données en figure 2.4 pour le cas  $k = 1$  et en figure 2.5 pour le cas  $k = 5$ .

### 2.1.5 Les machines à vecteurs de support (support vector machines)

Le principe d'un algorithme de type machine à vecteurs de support est de construire un hyperplan qui maximise la marge de séparation entre les données d'en-

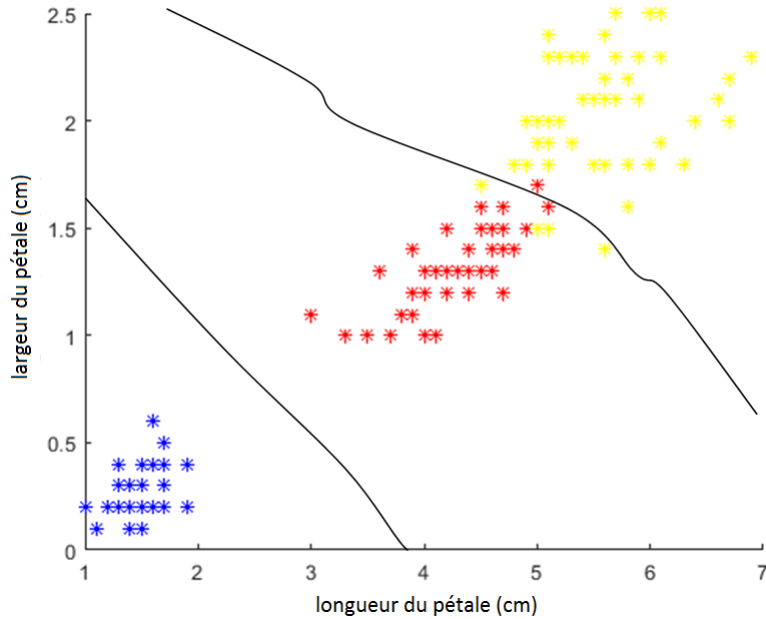


FIGURE 2.5 – Limites de classes des données Iris pour un algorithme k-nn avec k=5

trainement qui appartiennent à des classes différentes [Vapnik 1982, Burges 1998]. Il s’agit d’une généralisation des algorithmes de classification discriminants, les rendant plus précis par la garantie d’une séparation maximale des classes et qui évite aussi les problèmes de surentrainement.

Pour exprimer le principe de base, on considère dans un premier temps un problème à deux classes linéairement séparables dans un espace de dimension  $I$ . Un vecteur à classer dans cet espace est de la forme  $X = \{x_1, \dots, x_I\}$ . Une fonction qui permet de classer linéairement un élément  $X$  peut alors s’écrire :

$$f(X) = w^T X + b \tag{2.3}$$

avec  $X \in \text{classe}_1$  si  $f(X) < 0$  et  $X \in \text{classe}_2$  si  $f(X) > 0$

La fonction  $f$  représente un hyperplan qui sépare les données. Il est alors possible de définir ce plan de manière à maximiser l’écart entre l’hyperplan et les données d’entraînement. Sur un ensemble d’entraînement  $\{X_n, n \in [1, \dots, N]\}$ , la marge qui exprime l’écart entre les données et l’hyperplan s’écrit :

$$M = \min_n \left( \frac{|w^T X_n + b|}{\|w\|} \right) \tag{2.4}$$

Le choix du couple  $\{w, b\}$  est alors un problème d’optimisation sous contrainte dans lequel on cherche à maximiser la marge  $M$ .

L’algorithme peut être étendu aux cas où les données ne sont pas séparables en introduisant en plus dans le problème une fonction de coût pondérée par un

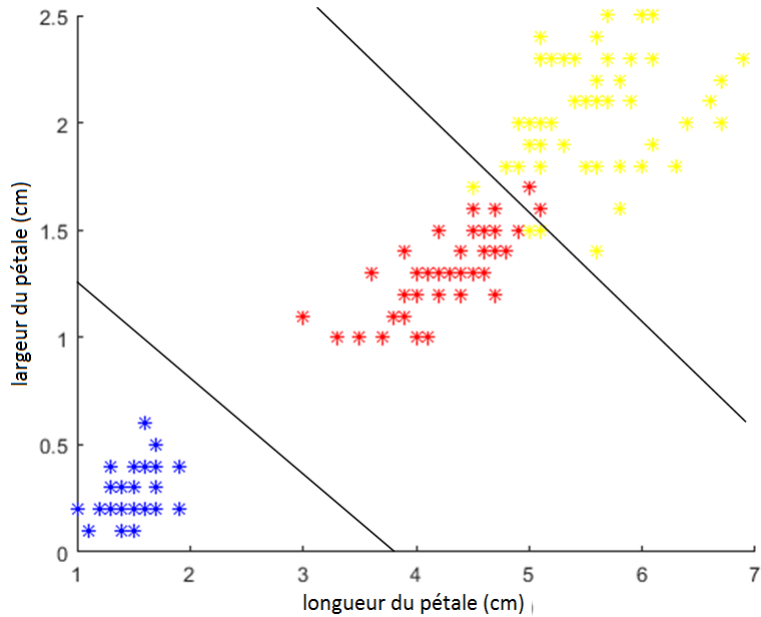


FIGURE 2.6 – Limites de classes des données Iris pour un algorithme de type SVM linéaire

paramètre selon l'importance à donner aux données mal classées. La fonction erreur peut, par exemple, être linéaire ou quadratique.

Il est également possible de classer des données non linéairement séparables en effectuant une transformation de l'espace de base vers un nouvel espace de dimension supérieure ou égale dans lequel les données seront linéairement séparables. Ceci est réalisé à l'aide d'une fonction noyau.

La méthode s'étend aussi aux problèmes multi-classes [Hsu 2002]. Une première solution consiste à entraîner l'algorithme comme un ensemble de problèmes à deux classes en isolant une classe et en l'entraînant par rapport à l'ensemble des autres classes (stratégie dite « un contre tous »). Une autre approche considère uniquement des classes deux à deux en réduisant à chaque phase d'entraînement le jeu de données d'apprentissage pour ne garder que deux classes (stratégie dite « un contre un »).

Les SVM sont des outils très puissants pour réaliser de la classification de données numériques, ils offrent des solutions pour des problèmes non-linéaires et non-convexes grâce au changement d'espace. De plus, la solution au problème d'optimisation pour la recherche de l'hyperplan est convexe et comporte une solution unique et sans minimums locaux. Néanmoins, le choix des paramètres tels que la fonction noyau a des conséquences difficiles à prévoir et peut notamment entraîner des problèmes de sur-apprentissage. Nous donnons en figure 2.6 et 2.7, le résultat obtenu grâce à cette méthode dans le cas de la base des Iris, respectivement dans les cas d'un SVM linéaire et d'un SVM quadratique.

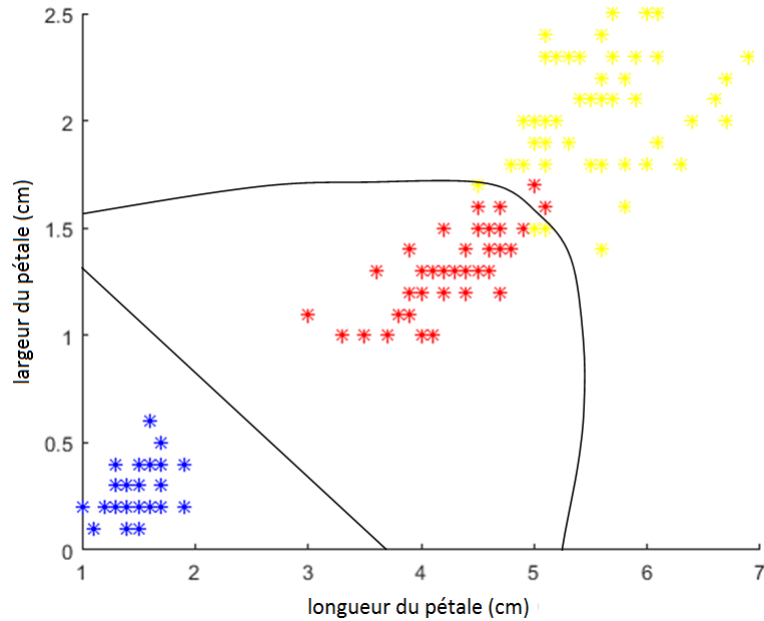


FIGURE 2.7 – Limites de classes des données Iris pour un algorithme de type SVM quadratique

### 2.1.6 Les réseaux de neurones artificiels

Un réseau de neurones artificiel est un assemblage de cellules sous forme de graphe qui imite le fonctionnement des neurones biologiques [Biem 2015]. Ces cellules sont nommées neurones et sont connectées entre elles au moyen d’arcs pondérés qui font varier l’intensité du signal entre les différentes cellules. Le réseau reçoit en entrée un vecteur  $X = \{x_1, \dots, x_I\}$  composé de  $I$  variables. Les neurones contiennent deux fonctions :

- une fonction d’agrégation qui combine les valeurs d’entrées du neurone en une seule valeur en fonction des paramètres  $W$  du neurone
- une fonction d’activation qui va déterminer l’état de sortie du neurone en fonction du résultat de la fonction d’agrégation

Un des agencements les plus simples est le perceptron (figure 2.8), composé d’une seule couche de neurones avec une fonction signe. Les paramètres sont donnés sous la forme d’un vecteur de poids  $W = \{w_0, w_1, \dots, w_I\}$ . L’agrégation se calcule dans le cas présent par :

$$a(W, X) = \sum_{i=1}^I x_i w_i + w_0 \quad (2.5)$$

Ce modèle est utilisable pour résoudre un problème de classification à deux classes, l’ensemble des sorties possible étant  $-1$  ;  $+1$ .



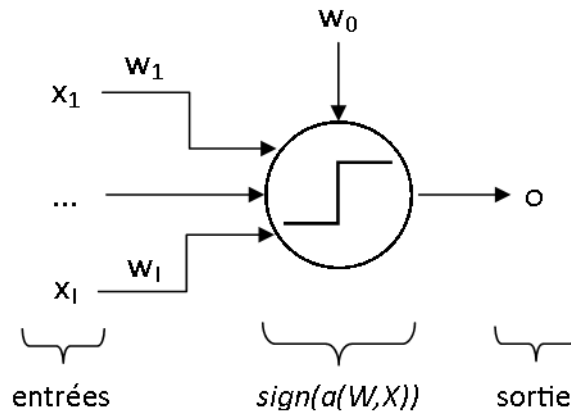


FIGURE 2.8 – Représentation schématique d'un perceptron

Historiquement, le premier modèle de réseau de neurones a été proposé en 1943 [McCulloch 1943] mais c'est en 1962 [Rosenblatt 1962] qu'apparaît une technique efficace pour régler les paramètres des perceptrons intitulée « perceptron learning rule ». Au fil des années, les réseaux de neurones ont connus de nombreuses évolutions sur leurs architectures avec notamment l'introduction de réseaux multicouches ou de nouvelles fonctions d'activation comme la sigmoïde. Mais des évolutions sont également à noter au niveau des algorithmes d'apprentissage avec l'introduction de l'algorithme de rétro propagation du gradient.

Aujourd'hui, la fonction sigmoïde est la plus couramment utilisée dans les réseaux à chaîne directe pour deux raisons : l'usage d'une fonction continue permet d'utiliser des algorithmes efficaces d'entraînement du réseau par descente de gradient, ce qui est impossible si les fonctions d'activation sont discontinues, et il a été montré [Rumelhart 1986] qu'un réseau de neurones avec des fonctions sigmoïdes et une seule couche cachée peut approximer n'importe quelle fonction continue à condition d'utiliser suffisamment de neurones.

Les réseaux de neurones peuvent également effectuer d'autres tâches que de la classification comme de l'approximation de fonction non-linéaire ou encore du clustering. Dans le cas de la classification, l'utilisation des réseaux de neurones a pour avantage d'offrir un calcul simple à mettre en œuvre une fois le réseau construit et ses paramètres identifiés et de permettre un large choix d'architectures, ouvrant notamment au « deep learning » [LeCun 2015]. Néanmoins il n'est pas toujours aisé de trouver l'ensemble des paramètres optimaux (poids, nombre de neurones, ...) pour un problème et le risque de sur-apprentissage est très élevé avec cette méthode. La figure 2.9 montre les séparations entre les classes des données Iris en utilisant un réseau de neurones artificiels avec dix neurones dans la couche cachée.

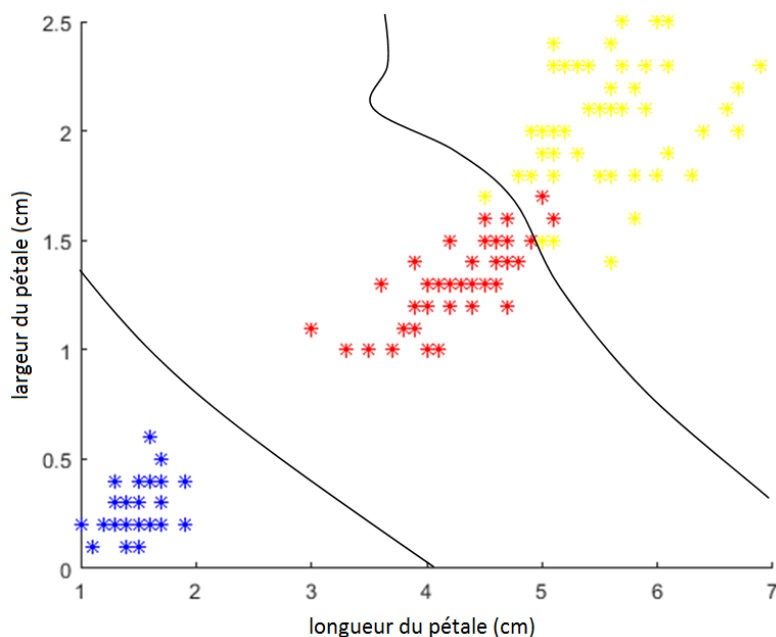


FIGURE 2.9 – Limites de classes pour un réseau de neurones avec dix neurones dans sa couche cachée

### 2.1.7 LAMDA (Learning Algorithm for Multivariate Data Analysis)

L'algorithme LAMDA a été introduit dans les années 80 par J. Aguilar-Martin [Aguilar-Martin 1982] en collaboration avec plusieurs chercheurs [Desroches 1987, Piera 1989]. C'est une méthode basée sur la logique floue [Zadeh 1965] capable d'effectuer de la classification et du partitionnement sur des grands ensembles de données.

L'apprentissage pour une classe donnée consiste à extraire de la base de données un ensemble de paramètres qui seront ensuite utilisés par des fonctions d'appartenance floue. Ces paramètres peuvent être par exemple une moyenne, un écart-type, etc. Une fois l'apprentissage réalisé, l'algorithme se déroule en trois étapes :

- le calcul de l'appartenance de chaque élément (attribut) du vecteur d'entrée à chacune des classes
- l'agrégation des degrés d'appartenances des attributs dans chaque classe
- la décision finale à partir des appartenances globales de la donnée aux différentes classes

L'algorithme est capable de traiter de manière simultanée et sans transformation préalable autre que la normalisation différents types d'attributs : des variables quantitatives (nombres réels), des variables qualitatives (liste d'étiquettes comme des couleurs, des formes, ...) et des intervalles flous ou non-flous. Dans le passage qui suit, seront présentées succinctement des fonctions qui permettent de mesurer

l'appartenance d'une donnée par rapport à des paramètres d'entraînement.

### 2.1.7.1 Données quantitatives

La fonction distance :

La fonction distance mesure linéairement la proximité d'un attribut  $x_i$  à la moyenne  $\rho_{c,i}$  de l'ensemble des éléments du ième attribut de la c-ième classe. Elle se calcule comme suit :

$$\mu(x_i) = 1 - |x_i - \rho_{c,i}| \quad (2.6)$$

Le fait que  $x_i$  et  $\rho_{c,i}$  appartiennent à l'ensemble  $[0;1]$  garantit que le résultat de la fonction sera sur l'ensemble  $[0;1]$ .

La fonction binomiale :

La fonction binomiale ne requière la connaissance que de la moyenne  $\rho_{c,i}$  des données d'une classe  $c$  pour un attribut  $x_i$  et se calcule de la manière suivante :

$$\mu(x_i) = \rho_{c,i}^{x_i} (1 - \rho_{c,i})^{1-x_i} \quad (2.7)$$

Cette fonction présente l'avantage d'avoir peu de paramètres et a un réel intérêt dans les problèmes à deux classes, sans quoi elle se montre souvent moins efficace que la binomiale centrée.

La fonction binomiale centrée :

La fonction binomiale centrée est une extension de la fonction binomiale qui a la particularité de présenter un maximum à 1 en un point fixé, ce qui présente plus d'intérêt dans les cas multi-classes.

$$\mu(x_i) = \phi_{c,i}^{1-|x_i-\rho_{c,i}|} (1 - \phi_{c,i})^{|x_i-\rho_{c,i}|} \quad (2.8)$$

Avec  $\rho_{c,i}$  la moyenne du ième attribut de la classe  $c$  et  $\phi_{c,i}$  un paramètre de dispersion des données du ième attribut dans la classe  $c$  qui se calcule de la manière suivante pour un ensemble d'apprentissage composé de  $N$  échantillons,  $x_{i,n}$  étant le ième attribut du n-ième échantillon :

$$\phi_{c,i} = 1 - \frac{1}{N} \sum_{n=1}^N |\rho_{c,i} - x_{i,n}| \quad (2.9)$$

La fonction gaussienne :

La fonction gaussienne suppose une répartition des données suivant une loi normale centrée sur la moyenne  $\rho_{i,c}$  du ième attribut pour la classe  $c$ . Elle nécessite donc la connaissance de la moyenne et de l'écart-type  $\sigma_{i,c}$  :

$$\mu(x_i) = e^{-\frac{(x_i - \rho_{i,c})^2}{2\sigma_{i,c}^2}} \quad (2.10)$$

### 2.1.7.2 Données qualitatives

Les données qualitatives pour un attribut  $i$  sont composées d'un ensemble de  $M$  modalités telles que  $D_i = \{Q_{1,i}, \dots, Q_{M,i}\}$ . Le degré d'appartenance est alors défini par la fréquence d'apparition de chacune des modalités pour une classe. On définit  $f_{m,i,c}$  la fréquence d'apparition de la modalité  $Q_{m,i}$  pour la classe  $c$ . Le calcul est alors le suivant :

$$\mu(x_i) = (f_{1,i,c})^{q_1} * \dots * (f_{M,i,c})^{q_M} \quad (2.11)$$

avec :

$$q_m = \begin{cases} 1 & \text{si } x_i = Q_{m,i} \\ 0 & \text{si } x_i \neq Q_{m,i} \end{cases} \quad (2.12)$$

### 2.1.7.3 Intervalles

La fonction d'appartenance d'un intervalle est dans le cas présent un calcul de similarité entre deux intervalles flous. Cette similarité comporte deux éléments dans son calcul : le taux de recouvrement des intervalles considérés et la distance qui les sépare. La fonction de similarité entre deux intervalles  $A$  et  $B$  dans un univers  $V$  s'écrit :

$$S(A, B) = \frac{1}{2} \left( \frac{\int_V \mu_{A \cap B}(\varepsilon) d\varepsilon}{\int_V \mu_{A \cup B}(\varepsilon) d\varepsilon} + 1 - \frac{\partial[A, B]}{\omega[V]} \right) \quad (2.13)$$

avec  $\partial[A, B]$  une mesure de distance entre les intervalles  $A$  et  $B$ ,  $\mu_X(x)$  qui est la valeur d'appartenance de  $x$  dans l'intervalle flou  $X$  et  $\omega[X]$  est la taille de l'intervalle  $X$  dans l'univers  $V$ .

$$\omega[X] = \int_V \mu_X(\varepsilon) d\varepsilon \quad (2.14)$$

Dans le cas d'intervalles non-flous, les équations 2.13 et 2.14 deviennent :

$$S(A, B) = \frac{1}{2} \left( \frac{\omega[A \cap B]}{\omega[A \cup B]} + 1 - \partial[A, B] \right) \quad (2.15)$$

avec

$$\omega[X] = \bar{X} - \underline{X} \quad (2.16)$$

$$\partial[A, B] = \max(0, \max(\underline{A}, \underline{B}) - \min(\bar{A}, \bar{B})) \quad (2.17)$$

où  $\underline{X}$  est la borne inférieure de l'intervalle  $X$  et  $\bar{X}$  sa borne supérieure.

Le paramètre de description d'une classe dans le cas des intervalles non flous est un intervalle moyen  $\rho_{c,i} = [\underline{\rho}_{c,i}; \bar{\rho}_{c,i}]$  dont la borne inférieure (respectivement supérieure) est la moyenne des bornes inférieures (respectivement supérieures) du ième attribut de la classe  $c$ .

#### 2.1.7.4 Agrégation et résultats de l'algorithme

Toutes ces fonctions d'appartenance permettent de placer l'ensemble des attributs d'entrée de natures différentes dans un espace commun (un hypercube) nommé espace des appartenances. Le vecteur d'entrée  $X$  est alors transformé en un ensemble de nouveaux vecteurs, un par classe, composés des degrés d'appartenance de chaque attribut pour une classe donnée. Ces vecteurs sont appelés degrés d'appartenance marginale (DAM).

Une fois tous les DAM calculés, l'algorithme procède à une agrégation de l'ensemble des valeurs pour créer un seul degré d'appartenance par classe qui représente l'appartenance globale du vecteur d'entrée  $X$  à une classe donnée. Cette valeur se nomme le degré d'appartenance globale (DAG). Un DAG pour une classe  $c$  se calcule de manière générale comme il suit :

$$DAG_c = \alpha \cdot \gamma(DAM_c) + (1 - \alpha) \cdot \beta(DAM_c) \quad (2.18)$$

avec les fonctions  $\gamma$  et  $\beta$  qui sont des fonctions d'agrégation floue, respectivement la t-norme (qui simule un ET logique ou une intersection) et la t-conorme (qui simule un OU logique ou une union). Ces fonctions peuvent être définies de plusieurs manières, néanmoins les deux qui reviennent le plus couramment sont :

- l'union et l'intersection probabilistes des éléments (somme pour la t-conorme et produit pour la t-norme)
- l'emploi des fonctions minimum et maximum (MIN pour la t-norme et MAX pour la t-conorme)

En résumé pour des variables floues  $x_1$  et  $x_2$  :

	Probabiliste	Min-Max
t-norme	$x_1 \cdot x_2$	$\min(x_1, x_2)$
t-conorme	$x_1 + x_2 - x_1 \cdot x_2$	$\max(x_1, x_2)$

Un autre paramètre apparaît dans l'équation 2.18, il s'agit d'un critère de sélectivité  $\alpha$  compris entre 0 et 1. Ce paramètre permet de choisir si la plus grande part du score final est donnée à l'agrégation par la t-norme ou par la t-conorme. Un  $\alpha$  proche de 0 va privilégier la t-conorme et ainsi produire un DAG grand si un seul des scores du vecteur de DAM est grand, alors qu'un  $\alpha$  qui s'approche de 1 va nécessiter que tous les scores du DAM soient grands pour que le DAG le soit également.

Une fois l'ensemble des DAG calculés, il en résulte un vecteur qui contient le degré d'appartenance pour chacune des classes. Il est possible de considérer ce vecteur comme la réponse de l'algorithme. Toutefois, une réponse non floue peut-être donnée en considérant que la classe à laquelle appartient l'échantillon est celle qui possède le DAG le plus élevé. Il peut aussi être ajouté une classe indéfinie (la classe NIC pour Non Informative Class) si aucun des DAG n'est au-dessus d'un seuil donné, ce qui indique que l'échantillon à classer ne ressemble à aucune classe connue et il sera alors mis dans la classe NIC.

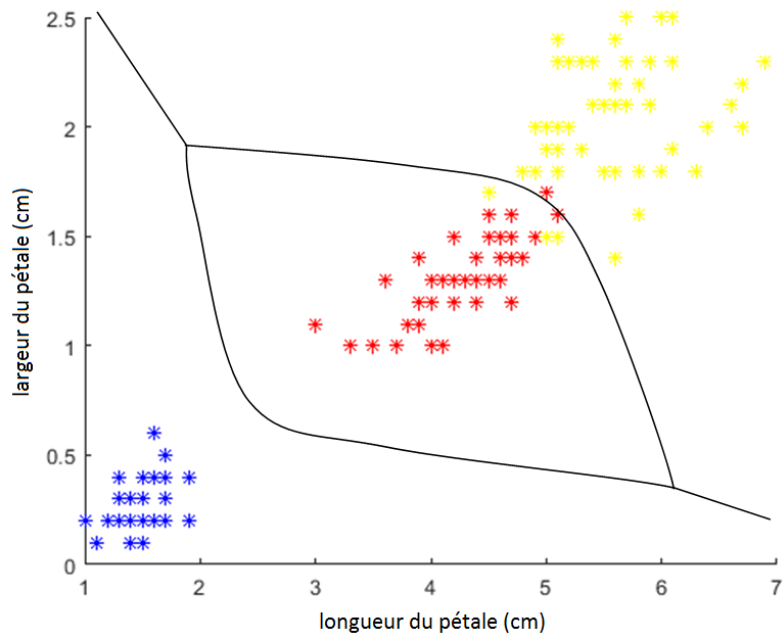


FIGURE 2.10 – Limites de classes pour l’algorithme LAMDA utilisant la fonction binomiale

Sur les figures 2.10 et 2.11 nous donnons les résultats obtenus pour la séparation des données Iris avec l’algorithme LAMDA avec un  $\alpha = 0,8$  en utilisant respectivement la fonction binomiale et la fonction binomiale centrée.

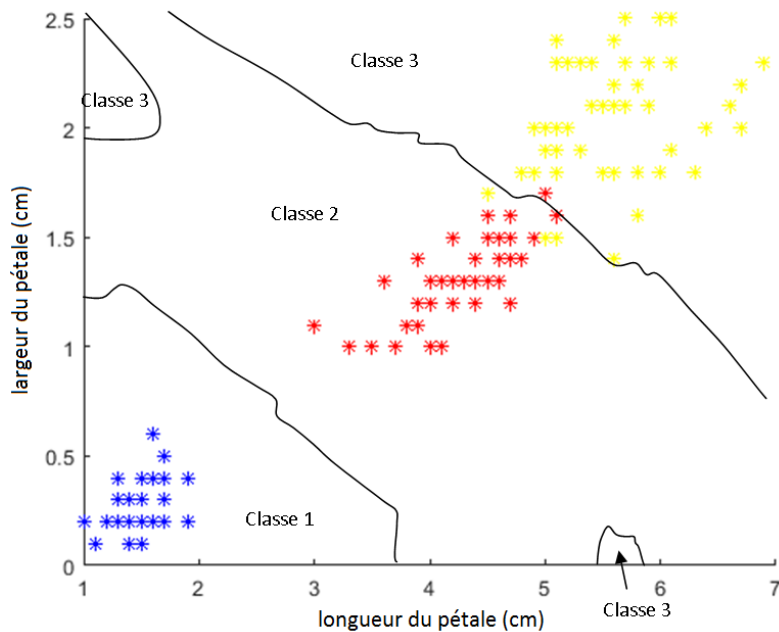


FIGURE 2.11 – Limites de classes pour l’algorithme LAMDA utilisant la fonction binomiale centrée

## 2.2 La sélection d’attributs

### 2.2.1 Introduction

Dans le cadre des problèmes de classification et de régression, la sélection d’attributs est l’étape qui consiste à réduire, à partir d’un ensemble de départ, le nombre de variables utilisées par l’algorithme. En effet, lors de la phase de collecte de données pour réaliser une classification, si l’utilisateur n’est pas expert du système étudié ou lorsque les variables enregistrées sont trop nombreuses, celles-ci peuvent ne pas être toutes utiles pour résoudre le problème. Les attributs considérés peuvent être redondants ou entièrement décorrélés du problème, ce qui dans le premier cas augmente inutilement la dimension du problème et dans le second introduit du bruit dans l’algorithme de classification, diminuant ainsi la précision du résultat. Par exemple, si on cherche à étudier la population à bord du Titanic, l’objectif étant de déterminer si une personne avait de grandes chances de survivre lors de l’accident ou non, on peut pour cela regarder plusieurs données comme le type de billet (1ère classe, 2ème classe, ...), le genre (homme ou femme) ou encore l’âge. L’ensemble de ces données semble être pertinent pour déterminer si une personne avait de grandes chances d’être secourue, par contre la connaissance du port dans lequel est montée la personne ne fournit pas d’information pour résoudre le problème, c’est un attribut qu’il faut éliminer, ce dernier apportant plus de bruit que d’information utile.

Les techniques pour réaliser une sélection d'attributs sont divisées en quatre grandes catégories :

- Les méthodes filtrantes : ces techniques utilisent des critères qui exploitent la répartition des données pour extraire un sous-ensemble de données.
- Les méthodes d'encapsulation : dans ces méthodes, les algorithmes de sélection sont englobés dans un algorithme de classification spécifique, ce qui fait que le processus de sélection est fortement lié à l'algorithme de classification utilisé.
- Les méthodes hybrides : les algorithmes hybrides utilisent à la fois des techniques filtrantes et d'encapsulation
- Les méthodes embarquées : dans le cas de ces méthodes, la sélection se fait lors de la phase d'apprentissage de l'algorithme de classification.

### 2.2.2 Les méthodes filtrantes

Ces techniques se basent sur l'usage de critères indépendants de toute technique de classification pour écarter les attributs moins pertinents. Ils incluent notamment des indices tels que le coefficient de Gini, la mesure d'entropie ou encore le score de Fisher, également utilisés dans la construction des arbres de décision.

Prenons par exemple, le score de Fisher (dans le cas continu). Soit un problème de classification donné composé de  $C$  classes, le score de Fisher pour le  $i$ ème attribut se calcule alors par :

$$S_i = \frac{\sum_{c=1}^C N_c (\mu_{i,c} - \mu_i)}{\sum_{c=1}^C N_c \sigma_{i,c}^2} \quad (2.19)$$

avec :

$N_c$  le nombre d'échantillons dans la classe  $c$

$\mu_i$  la moyenne globale du  $i$ ème attribut

$\mu_{i,c}$  la moyenne du  $i$ ème attribut dans la classe  $c$

$\sigma_{i,c}$  l'écart-type du  $i$ ème attribut dans la classe  $c$

Plus les scores calculés à l'aide de cette formule pour des attributs donnés sont élevés, plus la classification sera aisée si on utilise ces mêmes attributs.

Il existe d'autres approches statistiques qu'il est possible de rattacher aux méthodes filtrantes, comme par exemple l'utilisation de l'analyse par composantes principales (ACP). L'ACP est une technique statistique de réduction de l'espace. Un espace est construit en effectuant des combinaisons linéaires des différents axes de l'espace d'origine, de manière à exprimer au mieux la variance des données. L'algorithme construit autant d'axes qu'il en existe dans l'espace d'origine et c'est l'utilisateur qui choisit combien il souhaite en conserver (généralement deux ou trois pour permettre une représentation graphique). Chaque axe étant une combinaison linéaire de l'espace original, il est possible d'effectuer une sélection en conservant uniquement les attributs dont les poids possèdent les plus grandes valeurs absolues. Il existe néanmoins d'autres méthodes pour réaliser des sélections d'attributs



comme celle proposée par [Lu 2007].

Les méthodes filtrantes ont le grand avantage d'être très peu coûteuses en temps de calcul et de donner un résultat indépendant de tout algorithme de classification. Néanmoins, l'indépendance par rapport à la classification peut amener à supprimer des variables qui auraient pu apporter de l'information dans certains cas. Par exemple, le score de Fisher va donner des scores faibles aux données non convexes alors que de telles informations peuvent être traitées par certains algorithmes de classification. De plus, dans le cas de traitements mono-variables (un seul attribut est analysé à la fois), il n'est pas possible de détecter des éléments redondants.

### 2.2.3 Les méthodes d'encapsulation

La sélection d'attributs par encapsulation se base intégralement sur les résultats d'un algorithme de classification donné, utilisé avec un sous-ensemble de l'ensemble des attributs. Le sous-ensemble optimal est obtenu de la manière suivante :

1. Un sous-ensemble d'attribut est isolé
2. La classification est effectuée sur le sous-ensemble choisi lors de l'étape 1
3. On boucle sur 1 et 2 avec d'autres sous-ensembles jusqu'à trouver le sous-ensemble qui donne les meilleurs résultats de classification

Ce procédé est intéressant car il permet d'obtenir un sous-ensemble adapté à l'algorithme de classification qui sera utilisé par la suite. Néanmoins, ceci est très souvent coûteux en temps de calcul puisque la recherche du sous-ensemble optimal passe par un parcours combinatoire des attributs initiaux, ce qui peut être très long si la dimension de l'espace de base est grande.

Toutefois, il existe une autre solution introduite en 2011 par Lyamine Hedjazi [Hedjazi 2011] pour réaliser une sélection par encapsulation de manière non combinatoire. Cette méthode se nomme MEMBAS (MEMbership-Margin Based feAture Selection) et se base sur l'existence d'un « score » pour chaque attribut (degré d'appartenance, fonction de distance, ...). Soit un problème de classification qui comporte  $I$  attributs. On définit  $\mu_i^c(x_i)$  le score d'appartenance du  $i$ ème attribut à la classe  $c$ , ce score étant dépendant de la méthode de classification utilisée. De là, on construit un vecteur d'appartenances à la classe  $c$ ,  $M^c = \{\mu_1^c(x_1), \dots, \mu_I^c(x_I)\}$  à partir de l'entrée  $X = \{x_1, \dots, x_I\}$ . Si l'on se place dans un problème à deux classes  $c$  et  $nc$ , sachant que  $X$  appartient à la classe  $c$ , l'objectif de l'algorithme est de déterminer les poids  $W = \{w_1, \dots, w_I\}$  tels que :

$$\max(WM^c - WM^{nc}) \tag{2.20}$$

sous les contraintes  $\sum_{i=1}^I w_i^2 = 1$  et  $\forall i \in [1..I], w_i \geq 0$ . Ceci se traduit par le fait que les attributs qui sont discriminants auront un poids plus fort que les attributs dont la différence est faible ou négative.

L'avantage majeur de cette solution est que la recherche de l'optimal du problème possède une solution analytique (voir annexe A). Dans le cas de l'exploita-

tion d'une base de données avec beaucoup d'individus à classer, il est préférable de prendre la solution qui calcule les poids au fur et à mesure du traitement des individus. Il est possible de réaliser l'algorithme comme il suit :

1. Initialiser  $W = 0^I$
2. Calculer  $s = M_c - M_{nc}$
3.  $W = W + s$
4. Boucler sur l'étape 2 avec un nouvel échantillon
5.  $W = \max(W, 0)/I$

#### 2.2.4 Les méthodes hybrides

Les méthodes hybrides utilisent à la fois les méthodes filtrantes et d'encapsulation. On procède généralement par l'usage des filtres pour extraire un sous-ensemble d'attributs issu de l'ensemble initial puis dans un second temps on applique une méthode d'encapsulation sur le sous-ensemble.

L'hybride offre donc un bon compromis entre les méthodes filtrantes qui sont peu coûteuses en calcul mais également peu spécifiques et les méthodes d'encapsulation, qui sont quant à elles très coûteuses en temps de calcul.

#### 2.2.5 Les méthodes embarquées

Lors de l'utilisation de certains algorithmes de classification, il est possible d'obtenir une sélection de variable durant la phase d'entraînement, on parle alors de méthodes embarquées. On peut citer par exemple les arbres de décisions puisque leur construction intègre les mêmes calculs qu'une sélection par filtrage ou encore l'utilisation des poids de l'hyperplan avec les machines à vecteur de support.

Ces méthodes ont l'avantage d'offrir une interaction avec l'algorithme de classification et d'être moins coûteuses que les processus par encapsulation classiques. Par contre, la sélection obtenue est spécifique à l'algorithme de classification utilisé.

## 2.3 Les techniques de régression

### 2.3.1 Introduction

Les techniques de régression sont des méthodes statistiques qui ont pour objectif de créer un modèle expliquant l'évolution d'une variable aléatoire  $Y$  en fonction d'une variable  $X$ . Il s'agit d'un problème proche de la classification, la principale différence étant que la sortie d'une méthode de classification est qualitative là où une technique de régression a pour but d'exprimer une variable quantitative. Cette section présente les principales méthodes de régression pour des variables continues.

Comme dans beaucoup de problèmes statistiques, il est souvent nécessaire de mener une étude des données avant de tenter une modélisation par régression afin de détecter si des données aberrantes sont présentes, ou s'il existe des non linéarités évidentes qui peuvent être ramenées à un cas linéaire par une transformation.

### 2.3.2 Régression linéaire simple

Il s'agit du modèle de régression le plus simple et le plus élémentaire. L'objectif est de rechercher si une variable aléatoire  $Y$  peut être expliquée par une autre variable  $X$  à l'aide d'une fonction affine,  $X$  étant dans le cas présent composé d'une seule variable. Pour être plus précis, c'est l'espérance  $E(Y)$  qui est fonction de  $X$  sous la forme :

$$E(Y) = \beta_0 + \beta_1 X \quad (2.21)$$

Les paramètres du modèle  $\beta_0$  et  $\beta_1$  peuvent être déterminés en minimisant l'erreur quadratique entre le modèle et les observations. Pour un ensemble de valeurs  $\{X_n, Y_n\}$  avec  $n = 1, \dots, N$ , le problème d'optimisation est le suivant :

$$\min_{\beta_0, \beta_1} \sum_{n=1}^N (Y_n - \beta_0 - \beta_1 X_n)^2 \quad (2.22)$$

Il est montré que ce problème est résolu par :  $\beta_1 = \frac{s_{XY}}{s_X^2}$  et  $\beta_0 = \bar{Y} - \beta_1 \bar{X}$  sachant que  $s_X^2 = \frac{1}{N-1} \sum_{n=1}^N (X_n - \bar{X})^2$  et  $s_{XY} = \frac{1}{N-1} \sum_{n=1}^N (X_n - \bar{X})(Y_n - \bar{Y})$ .  $\bar{X}$  (respectivement  $\bar{Y}$ ) représente la valeur moyenne du vecteur  $X$  (respectivement  $Y$ ).

### 2.3.3 Régression linéaire multiple

La régression linéaire multiple étend le cas simple vu précédemment aux cas où  $X$  est composé de plusieurs variables [Besse 2013]. On étudie donc dans ce cas un ensemble de variables  $X_n, Y_n$  avec  $n = 1, \dots, N$  et  $X_n = x_n, 1, \dots, x_n, I$ .

Le problème étant cette fois matriciel, il est plus compliqué d'introduire de manière naturelle un terme constant afin de rendre le modèle affine. Pour contrer cela, la solution la plus simple est d'étendre le vecteur  $X$  avec un terme constant, généralement 1. Ce nouveau vecteur nommé  $P$  est alors défini par :  $P = \{1, x_1, \dots, x_I\}$  et permet alors de calculer  $E(Y)$  de la manière suivante :

$$E(Y) = P\beta = \beta_0 + x_1\beta_1 + \dots + x_I\beta_I \quad (2.23)$$

La détermination de l'ensemble des paramètres  $\beta$  est, de manière similaire au cas simple, effectuée à l'aide du critère des moindres carrés :

$$\min_{\beta} \sum_{n=1}^N (Y_n - P_n\beta)^2 = \min_{\beta} \|Y - P\beta\|^2 \quad (2.24)$$

Cette équation conduit à la recherche d'un minimum par recherche d'un gradient nul qui mène au résultat :

$$\beta = (P'P)^{-1}P'Y \quad (2.25)$$

Cette méthode garantit d'obtenir le meilleur modèle linéaire sans biais. Il existe plusieurs indicateurs pour permettre d'évaluer la pertinence du modèle (par exemple

le coefficient de détermination). Il convient également de faire attention au conditionnement de la matrice  $P'P$ , car dans le cas où celle-ci serait mal conditionnée (déterminant proche de zéro), l'inversion va générer un modèle avec une forte variance sur les estimateurs.

Le cas présenté ici est un cas particulier du modèle linéaire généralisé, les résultats présents étant valides pour une erreur sur la variable  $Y$  de type normale centrée. Le modèle général non abordé ici est valide pour tout type de loi de probabilité.

### 2.3.4 Régression polynomiale

La régression polynomiale cherche à établir un modèle sous la forme d'un polynôme entre une variable  $X$  (mono-variable) et une variable aléatoire  $Y$  [Zhou 2016]. Il s'agit d'une modélisation simple mais qui n'a dans la plupart des cas aucun sens physique. On considère un échantillon de données  $\{X_n, Y_n\}$ ,  $n = 1, \dots, N$ .

Le modèle pour un polynôme de degré  $M$  prend la forme suivante :

$$E(Y) = R(X) = \sum_{m=0}^M \beta_m X^m \quad (2.26)$$

De manière similaire à ce qui a été présenté pour la régression linéaire, la recherche des coefficients du polynôme  $\beta = \{\beta_1, \dots, \beta_M\}$  passe la plupart du temps par la minimisation de l'erreur quadratique.

$$\min_{\beta} \sum_{n=1}^N (Y_n - R(X_n))^2 \quad (2.27)$$

La solution de cette optimisation se présente sous la forme  $\beta = (T'T)^{-1}T'Y$  avec  $T_{n,m} = X_n^{m-1}$ ,  $n = 1, \dots, N$ ,  $m = 1, \dots, M$

### 2.3.5 Réseaux de neurones

Le modèle de réseau de neurones est présenté dans la sous-section 2.1.6. Comme indiqué dans cette partie, les réseaux de neurones peuvent être utilisés pour résoudre des problèmes de classification ou bien des problèmes de régression multi-variables (plusieurs entrées et plusieurs sorties). La différence principale étant que dans le cas de la classification l'objectif de sortie du réseau est un état discret (le domaine des classes) alors qu'en régression l'objectif de sortie est continu.

Dans le chapitre suivant, nous allons aborder la sélection des informations fournies par les capteurs de laboratoire afin de mieux identifier les état de cuisson d'aliments.

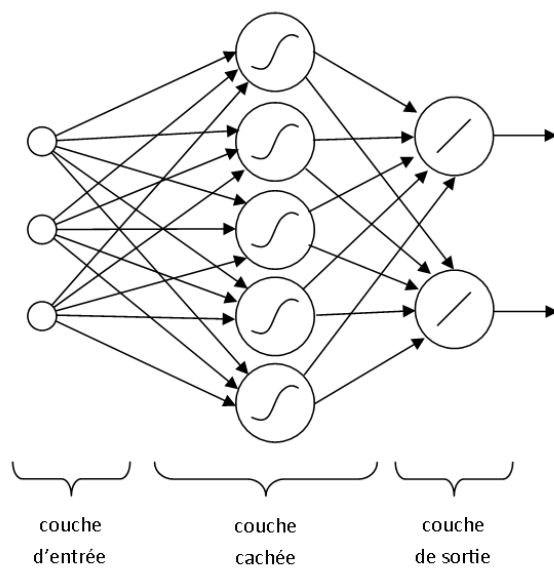


FIGURE 2.12 – Représentation graphique d'un réseau de neurones artificiels avec trois neurones sur la couche d'entrée, cinq neurones sur la couche cachée et deux neurones sur la couche de sortie

# Sélection d'attributs pour le suivi de cuisson d'aliments à partir de mesures expérimentales

---

## Sommaire

---

<b>3.1</b>	<b>Introduction</b>	<b>32</b>
<b>3.2</b>	<b>Matériel et méthodes</b>	<b>33</b>
<b>3.3</b>	<b>Études sur les composés organiques volatils</b>	<b>34</b>
3.3.1	Cabillaud	37
3.3.2	Saumon	38
3.3.3	Truite	39
3.3.4	Rosbif	39
3.3.5	Rôti de porc	40
3.3.6	Rôti de veau	42
3.3.7	Blanc de poulet	42
3.3.8	Cuisse de poulet	43
3.3.9	Bilan sur les composés organiques volatils	45
<b>3.4</b>	<b>Mesures spectrales suite à une excitation par lumière fluorescente</b>	<b>46</b>
3.4.1	Cabillaud	47
3.4.2	Saumon	48
3.4.3	Truite	49
3.4.4	Rosbif	49
3.4.5	Rôti de porc	49
3.4.6	Rôti de veau	51
3.4.7	Blanc de poulet	51
3.4.8	Cuisse de poulet	52
3.4.9	Bilan sur l'analyse par fluorescence	52
<b>3.5</b>	<b>Mesures de réflexion dans le visible et le proche infrarouge</b>	<b>54</b>
3.5.1	Cabillaud	55
3.5.2	Saumon	55
3.5.3	Truite	55
3.5.4	Rosbif	57

3.5.5	Rôti de porc . . . . .	57
3.5.6	Rôti de veau . . . . .	57
3.5.7	Blanc de poulet . . . . .	59
3.5.8	Cuisse de poulet . . . . .	59
3.5.9	Bilan sur les spectres dans le visible et le proche infrarouge . . . . .	59
<b>3.6</b>	<b>Bilan de la sélection d'attributs pour le suivi de l'état de cuisson d'aliments . . . . .</b>	<b>61</b>

---

### 3.1 Introduction

La première étape du projet Opticook en ce qui concerne les travaux de cette thèse fut de chercher parmi un ensemble de mesures, réalisées par différents laboratoires et entreprises, les descripteurs contenant le plus d'informations pour caractériser différents états de cuisson.

Les expériences ont été réalisées sur huit types d'aliments, trois poissons (cabillaud, truite et saumon) et cinq viandes (rosbif, blanc de poulet, cuisse de poulet, rôti de veau, rôti de porc). À chaque fois, les cuissons sont faites jusqu'à atteindre trois états qui correspondent à une température à cœur précise pour chaque état, la température étant mesurée à l'aide d'une sonde de température plantée dans l'aliment. Ces trois niveaux de cuisson à cœur sont étiquetés de la manière suivante : « sous cuit », « bien cuit » et « trop cuit » et sont explicités dans la table 3.1.

Type d'aliment	Température à cœur de l'état sous cuit (°C)	Température à cœur de l'état bien cuit (°C)	Température à cœur de l'état trop cuit (°C)
Cabillaud	50	60	70
Saumon	50	60	70
Truite	50	60	70
Roti de porc	65	75	85
Rosbif	45	52	68
Roti de veau	57	67	77
Blanc de poulet	56	66	76
Cuisse de poulet	75	82	93

TABLE 3.1 – Tableau récapitulatif des températures internes des différents aliments pour chaque état de cuisson

Les expériences étant réalisées par un ensemble d'intervenants différents, un protocole de cuisson est défini pour rendre homogène les résultats. Pour toutes les expériences, le four est préchauffé dans un premier temps à une température qui dépend de l'élément à faire cuire, puis l'aliment est enfourné et les éléments chauffants du four sont mis hors tension. Ainsi le produit à analyser va cuire selon l'inertie thermique du four, et sera retiré une fois que la température à son cœur

souhaitée sera atteinte. De même pour rendre les expériences homogènes, un facteur de forme est défini pour chaque produit à cuire, indiquant la masse de la ou des pièces à traiter, ainsi que dans certains cas des dimensions en termes de longueur et de diamètre. Ces informations sont données dans la table 3.2.

Type d'aliment	Nombre de pièces	Masse (g)	Diamètre (mm)	Longueur (mm)
Cabillaud	2	entre 100 et 200	x	entre 175 et 185
Saumon	1	140	x	entre 175 et 185
Truite	1	250	x	entre 280 et 300
Roti de porc	1	entre 500 et 700	80	100
Rosbif	1	entre 400 et 600	80	100
Roti de veau	1	entre 500 et 700	80	100
Blanc de poulet	2	140	x	x
Cuisse de poulet	2	250	x	x

TABLE 3.2 – Masses et dimensions des ingrédients utilisés

Les différentes mesures réalisées sont les suivantes : composés volatiles en suspension (ANSES et Alpha MOS), spectre visuel et proche infrarouge sous lumière blanche (PAM-PAPC) et spectre sous lumière ultra-violette à 380 nm (PAM-PAPC).

## 3.2 Matériel et méthodes

Les cuissons sont réalisées à l'aide d'un four domestique de la marque SEB (figure 3.1). Ce dernier est relié à un boîtier électronique d'interface qui recueille l'ensemble des informations des capteurs du four. Le capteur le plus important dans le cadre des cuissons pour réaliser la sélection d'attributs à partir de données hors-ligne est la sonde de température à cœur. Cette dernière est composée d'une pointe insérée dans le produit à cuire et maintenue en place grâce à un support métallique. Cette sonde permet notamment de savoir quand l'expérimentateur doit stopper la cuisson.

L'objectif est de réaliser une sélection d'attributs parmi l'ensemble de ceux mesurés par les différents partenaires, un algorithme de sélection est donc utilisé pour atteindre ce but. Le nombre d'attributs pouvant s'élever à plusieurs centaines, il n'est pas envisageable d'utiliser une méthode par encapsulation classique du fait du nombre de combinaisons à traiter. La plupart des données sont analysées à l'aide d'une analyse en composantes principales par les partenaires qui génèrent les mesures. Dans notre cas, nous avons choisi d'employer l'algorithme MEMBAS couplé avec une mesure d'appartenance floue à l'aide de la fonction binomiale centrée (équation 2.8, page 20). Cet algorithme est intéressant car il propose une solution analytique qui fournit les attributs qui maximisent l'écart entre les classes.

Une fois la sélection effectuée, il reste encore à chercher, parmi les attributs qui ont les poids les plus élevés, lesquels permettent de maximiser le taux de réussite



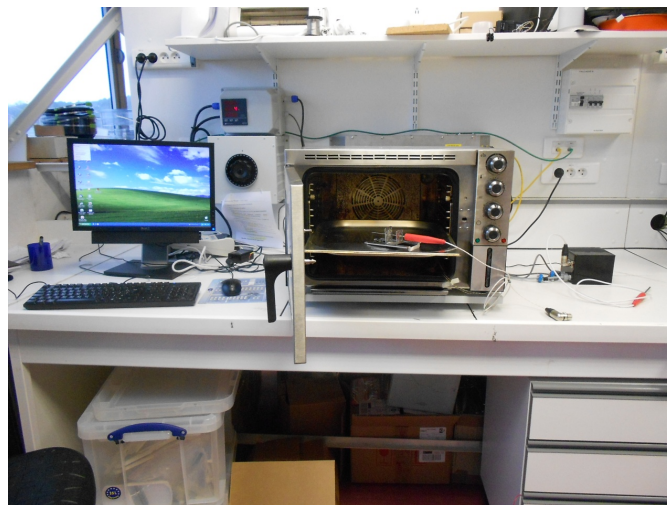


FIGURE 3.1 – Photographie d'un four servant à réaliser les cuissons pour la sélection d'attributs

de classification. On indiquera parfois pour information le taux de classification cumulé, qui indique le taux de réussite de la classification pour un vecteur de données composé des  $j$  attributs avec les scores de sélection les plus élevés. Par exemple, si on prend  $j = 1$ , la classification est réalisée uniquement avec l'attribut qui possède le score le plus haut, alors que pour  $j = 3$ , la classification se fait avec les trois attributs avec les scores de sélection les plus grands. Au final, il sera conservé un sous-ensemble d'attributs qui maximisent la classification, selon les règles suivantes : le nombre d'attributs conservé est le plus faible possible et le score des attributs est le plus grand possible.

### 3.3 Études sur les composés organiques volatils

Deux entités ont effectué des mesures de composés volatils dans le but d'identifier lesquels parmi ces composés permettent le mieux d'identifier l'état de cuisson d'un aliment : l'ANSES et Alpha MOS (entreprise Toulousaine spécialisée dans la conception de nez électroniques). Les mesures réalisées par ces deux groupes sont basées sur des protocoles différents dont est donnée ci-dessous une explication succincte.

L'équipe de l'ANSES a réalisé une analyse en piégeant les composés diffusés durant la cuisson à l'aide d'une fibre, puis lorsque la cuisson est terminée, la fibre est récupérée et on en extrait les molécules en la chauffant, pour enfin en réaliser une chromatographie gazeuse. La première partie du procédé qui consiste à recueillir les composés se nomme « Solid Phase Micro Extraction » (SPME). Une SPME permet d'extraire une quantité proportionnelle de composés par rapport à ce qui se trouve dans l'environnement de la fibre. Cette fibre est faite de silice fondue enrobée d'un matériau polymère, le tout permettant l'établissement d'un équilibre

thermodynamique entre les composés absorbés par la fibre et l'environnement. Il est cependant à noter que la fibre est sélective, elle n'est capable de stocker que certains types d'éléments en fonction de son diamètre et de sa nature. Pour faire sortir les molécules piégées dans la fibre, cette dernière est placée dans un injecteur chauffé à 250 °C. Le gaz extrait est alors soumis à une chromatographie gazeuse : la phase gazeuse à analyser passe à travers une phase liquide qui va faire remonter les composés à des vitesses différentes selon leur nature. Les composés sont donc séparés dans l'espace et dans le temps par cette opération. Une fois une molécule isolée, il reste à l'identifier à l'aide d'un spectromètre de masse, ce dernier permettant de reconnaître une substance grâce au spectre lumineux émis par cette dernière. Les résultats de la mesure seront l'intensité du spectre pour une molécule donnée.

Dans le cas d'Alpha MOS, un échantillon de 3 grammes du produit est prélevé en fin de cuisson puis placé dans un flacon. Ce dernier est maintenu à une température de 50 °C de manière à diffuser au maximum les composés dans l'air sans faire cuire d'avantage le produit. Un échantillon d'air est ensuite prélevé dans le flacon et est passé en chromatographie gazeuse, comme dans le cas des manipulations réalisées par l'ANSES. Néanmoins, l'identification des molécules n'est pas faite à l'aide d'un spectromètre mais provient de la connaissance du temps de rétention lors de la chromatographie. Comme à un temps peut correspondre plusieurs molécules, le dispositif utilisé par Alpha MOS comporte deux colonnes de chromatographie qui possèdent des propriétés différentes et donc des temps de rétention différents pour un même composé. La connaissance du temps des deux colonnes permet donc dans la plupart des cas d'identifier un composé. Une fois qu'une molécule sort du chromatographe, sa quantité est estimée à l'aide d'un FID (détecteur à ionisation de flamme). La base de données pour ces mesures contient des intensités pour chaque temps de rétention pour une colonne donnée. Le nombre d'expériences et le nombre de descripteurs associés est donné dans la table 3.3.

Les molécules identifiées comme importantes parmi celle issues des mesure d'Alpha MOS ont été associées à une odeur ressentie par un nez humain. Cela a été réalisé en utilisant la base de données [AroChemBase], qui permet notamment l'analyse d'odeurs et de composés chimiques à partir de la chromatographie en phase gazeuse.

Source de la mesure	Type d'aliment	Nombre d'expériences	Nombre de descripteurs
ANSES	Cabillaud	8	32
	Saumon	9	129
	Truite	9	100
	Bœuf	9	98
	Porc	9	59
	Veau	9	47
	Cuisse de poulet	9	66
	Blanc de poulet	9	70
Alpha MOS	Cabillaud	27	18
	Saumon	15	123
	Truite	15	87
	Bœuf	9	125
	Porc	9	105
	Veau	9	92
	Cuisse de poulet	9	100
	Blanc de poulet	9	95

TABLE 3.3 – Nombres d'expériences et de descripteurs pour les analyses de composés organiques volatils

### 3.3.1 Cabillaud

La sélection est effectuée sur chaque produit afin de trouver les descripteurs qui permettent une classification avec le plus de précision. Les scores de sélection pour les mesures de l'ANSES sont répertoriés dans la table 3.4 et ceux d'Alpha MOS dans la table 3.5.

Nom de variable	Poids MEMBAS	Taux de classification cumulé (%)
Pentanal	0,32199	100
2,3-butanedione	0,290245	100
Phenol	0,273042	100
Disulfide-dimethyl	0,254792	100
2-Decanone	0,249624	100
2-Butanone	0,242903	100
Furfural	0,233066	100
D-Limonene	0,228944	100
Benzothiazole	0,219249	100
Acetophenone	0,216848	100

TABLE 3.4 – Résultats de la sélection des composés organiques volatils par MEMBAS pour du cabillaud avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule associée	Score	Taux de classification cumulé (%)
405.40-1-Area	Butane	0.541620	80,25
658.95-1-Area	Butanal-3-methyl	0.449253	80,25
668.85-1-Area	Butanal-2-methyl	0.411386	80,25
641.08-2-Area	Butanal	0.410677	80,25
747.42-2-Area	Butanal-2-methyl	0.393406	88,89
380.70-1-Area	Triméthylamine	0.088006	88,89
995.98-2-Area	Decane	0.051276	91,36
999.40-1-Area	Decane	0.037120	92,59
776.55-1-Area	Toluene	0.019004	93,83

TABLE 3.5 – Résultats de la sélection des composés organiques volatils par MEMBAS pour du cabillaud avec la méthode d'extraction d'Alpha MOS

Dans le cas des données d'ANSES, la mesure d'une seule molécule, le pentanal, permet de classer correctement l'ensemble des échantillons. Les données issues des expériences d'Alpha MOS en revanche nécessitent plusieurs attributs pour maximiser les résultats pour l'algorithme de classification utilisé ici, à savoir le butane, le butanal-2-methyl et le toluène.

En équivalent olfactif, le butanal-3-methyl est associé à une odeur verte, herbacée et toastée, le butanal-2-methyl à du brûlé, la triméthylamine à du poisson et le

toluène à une odeur caramélisée. Les molécules identifiées comme étant «d'intérêt» ont donc une certaine logique par rapport aux sens humains.

### 3.3.2 Saumon

La sélection sur les cuissons de saumon dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.6 et 3.7, respectivement pour ANSES et Alpha MOS.

Nom de variable	Score	Taux de classification cumulé (%)
2-Decanone	0,356959	100
Isopropyl Alcohol	0,237029	100
2-Dodecanone	0,213393	100
Furfural	0,198466	100
Tetradecane	0,1954	100
2(3H)-Furanone-5-hexyldihydro	0,191845	100

TABLE 3.6 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du saumon avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule associée	Score	Taux de classification cumulé (%)
657,61-1-Area	Butanal-3-methyl	0,677283	73,3
752,60-1-Area	octane isomer	0,474911	75,6
747,41-2-Area	Butanal-2-methyl	0,330336	77,8
949,51-1-Area	alpha-Pinene	0,256517	73,3
640,16-2-Area	Butanal	0,247498	75,5
431,33-1-Area	Acetaldehyde	0,21287	73,3
828,77-2-Area	Toluene	0,165704	84,4

TABLE 3.7 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du saumon avec la méthode d'extraction d'Alpha MOS

Dans le cadre des mesures de l'ANSES, le 2-Decanone permet de classer correctement 100 % des échantillons, c'est également le cas avec l'alcool isopropyl et le 2-dodecanone. Néanmoins l'alcool isopropyl est à écarter du fait de la forte probabilité qu'il soit un polluant. En ce qui concerne les données d'Alpha MOS, la précision de la classification est maximisée avec le butanal-3-methyl, le butanal-2-methyl et le toluène.

Il est à noter que l'on retrouve des molécules communes dans les deux bases entre le saumon et le cabillaud.

### 3.3.3 Truite

La sélection sur les cuissons de truite dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.8 et 3.9, respectivement pour l'ANSES et Alpha MOS. Encore une fois, une seule molécule permet de classer correctement à 100 % les données de l'ANSES, le 1-butanol, mais cela fonctionne aussi avec le dodécanal et le furan-2-méthyl. Il est par contre beaucoup plus difficile de classer les données d'Alpha MOS avec une oscillation entre 57,8 % et 60 % de réussite, ce qui est assez faible. Les molécules les plus importantes sont toutefois présentes dans les cas du saumon et du cabillaud.

Nom de variable	Score	Taux de classification cumulé (%)
1-Butanol	0,280464	100
Dodécanal	0,270693	100
Furan-2-méthyl	0,224802	100
Hexanal	0,213224	100
Nonanal	0,205548	100
Pentanal	0,196149	100
Furfural	0,182927	100

TABLE 3.8 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la truite avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
656.12-1-Area	Butanal-3-méthyl	0,693323	57,8
666.59-1-Area	Butanal-2-méthyl	0,512318	60
746.19-2-Area	Butanal-2-méthyl	0,498338	57,8
640.26-2-Area	Butanal	0,092158	57,8

TABLE 3.9 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la truite avec la méthode d'extraction d'Alpha MOS

### 3.3.4 Rosbif

La sélection sur les cuissons de bœuf dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.10 et 3.11.

L'acide pentanoïque permet de classer correctement à 100 % les données issues des cuissons de l'ANSES, le second attribut le plus important, le 2-pentadécanone fait diminuer la précision. Du côté des mesures d'Alpha MOS, la combinaison de l'hexanal, de 2-méthyl furane et de méthyl-2 méthylbutanoate permet une classification avec 85 % de réussite.

Nom de variable	Score	Taux de classification cumulé (%)
Pentanoic acid	0,533181	100
2-Pentadecanone	0,40986	77,8
2-Butanone	0,306692	100
D-Limonene	0,232188	100
Phenol	0,215155	100
2-Octenal,(E)-	0,207305	100
2,4-Decadienal,(E,E)-	0,183214	100
2-Butanone,3-hydroxy-	0,17914	100

TABLE 3.10 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rosbif avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
898.86-2-Area	hexanal	0,4305	62%
807.16-1-Area	hexanal	0,4188	66%
644.06-2-Area	2-methyl furane	0,3565	81%
832.47-2-Area	methyl-2 methylbutanoate	0,3233	85%
757.17-1-Area	dimethyldisulfide	0,2878	85%
794.09-2-Area	dimethyldisulfide	0,2839	85%

TABLE 3.11 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rosbif avec la méthode d'extraction d'Alpha MOS

L'hexanal est associé à une odeur de gras et le 2-methyl furane à une odeur de brûlé.

### 3.3.5 Rôti de porc

Sur les tables 3.12 et 3.13 ont été portés les résultats de la sélection sur les cuissons de rôti de porc dans le cadre de la mesure des composés organiques volatils.

Pour obtenir un taux de classification de 100 % avec les données de l'ANSES, le duo de molécules 2,4-octadienal,(E) et 2-decanone est suffisant. En ce qui concerne les données d'Alpha MOS, c'est le triplet pyrazine, benzaldehyde et triméthylamine qui maximise le nombre de bons résultats avec un taux de 81,48 %.

La pyrazine peut être liée à une odeur de type gras ou toasté, le benzaldehyde à du boisé ou une odeur d'amande et la triméthylamine à du piquant.

Nom de variable	Score	Taux de classification cumulé (%)
2,4-Octadienal,(E)	0,476468	88,89
Furfural	0,377251	88,89
2-Decanone	0,35709	100
Benzaldehyde	0,332235	100
2-Heptenal,(Z)-	0,25996	100
1-Pentanol	0,188359	100
2,4-Nonadienal,(E,E)-	0,164989	100
Decanal	0,15938	100

TABLE 3.12 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de porc avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
827.81-2-Area	pyrazine/pyridine	0,409893	74,07
555.26-1-Area	Carbon disulfide	0,369094	62,96
459.91-1-Area	acetaldehyde	0,366452	59,25
637.91-2-Area	Carbon disulfide/2-methylpropanal	0,324839	59,25
699.06-2-Area	2-butanone	0,30505	66,67
1111.64-2-Area	benzaldehyde	0,274779	77,78
753.27-1-Area	pentanol	0,229601	77,78
445.69-1-Area	Trimethylamine	0,205734	81,48

TABLE 3.13 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de porc avec la méthode d'extraction d'Alpha MOS



### 3.3.6 Rôti de veau

La sélection sur les cuissons de rôti de veau dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.14 et 3.15.

Nom de variable	Score	Taux de classification cumulé (%)
2-Undecenal	0,478145	88,89
2-Octenal,(E)-	0,429024	100
Undecanal	0,361188	100
Benzene acetalde- hyde	0,279161	100
Octanal	0,243765	100
2-Heptenal,(E)-	0,222307	100
Heptanal	0,212486	100
Heptanoic acid	0,185424	100

TABLE 3.14 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de veau avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
826.83-2-Area	pyrazine/pyridine	0,499067	81,48
446.00-1-Area	acetaldehyde	0,424549	85,19
752.90-1-Area	pyrazine	0,423665	85,19
487.59-2-Area	acetaldehyde	0,305268	81,48
637.57-2-Area	Carbon disulfide	0,305173	81,48
555.13-1-Area	Carbon disulfide	0,270709	81,48
444.54-2-Area	Trimethylamine	0,212375	85,19

TABLE 3.15 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du rôti de veau avec la méthode d'extraction d'Alpha MOS

Les données de l'ANSES peuvent être classées à 100 % avec les molécules 2-undecenal et 2-octenal,(E)-. Les mesures d'Alpha MOS sont classées au mieux avec la pyrazine et l'acetaldehyde.

Comme indiqué dans la partie sur le rôti de porc, la pyrazine est associée au gras et toasté, l'acetaldehyde quant à elle est liée à une odeur de frais.

### 3.3.7 Blanc de poulet

La sélection sur les cuissons de blanc de poulet dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.16 et 3.17.

Les mesures de l'ANSES sont correctement classées à 100 % avec le p-xylène et le 2-nonanone. Quant aux données d'Alpha MOS, le 2-butanol seul permet de les

classer sans erreur. Le 2-butanol est associé à une odeur de gras.

Nom de variable	Score	Taux de classification cumulé (%)
p-Xylene	0,403368	66,67
Benzaldehyde	0,399051	66,67
2-Nonanone	0,393942	100
1-Hexanol,2-ethyl-	0,329078	100
Pentanoic acid	0,297543	100
2-Octenal,(E)-	0,278338	100
1-Nonanol	0,24549	100
Undecanal	0,227198	100

TABLE 3.16 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du blanc de poulet avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
703.28-2-Area	2-butanol/2-butanone	0,266253	81,48
903.64-1-Area	2,5-Dimethylpyrazine/Pentanoic acid	0,258518	88,89
657.88-1-Area	3-methylbutanal	0,251728	96,30
602.40-1-Area	2-butanol/2-butanone	0,228817	100,00

TABLE 3.17 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour du blanc de poulet avec la méthode d'extraction d'Alpha MOS

### 3.3.8 Cuisse de poulet

La sélection sur les cuissons de cuisse de poulet dans le cadre de la mesure des composés organiques volatils donne les résultats des tables 3.18 et 3.19.

Les molécules 2,4-octadienal,(E,E)-, 2-nonanone et tridecanal permettent de classer avec un taux de réussite de 100 % les mesures de l'ANSES. Les données d'Alpha MOS sont caractérisées avec un taux de 88,89 % avec le 3-methyl butanal et l'heptanal.

Le 3-methyl butanal est associé à une odeur toastée et l'heptanal à du gras.

Nom de variable	Score	Taux de classification cumulé (%)
2,4-Octadienal,(E,E)-	0,339397	88,89
Benzeneacetaldehyde	0,314567	88,89
2-Nonanone	0,289065	88,89
Tridecanal	0,286502	100
1-Nonanol	0,284899	100
2-Hexanone	0,27005	100
2-Decanone	0,240277	100
2-Butenal	0,230294	100
Undecanal	0,229536	100

TABLE 3.18 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la cuisse de poulet avec la méthode d'extraction de l'ANSES

Nom de variable	Nom de molécule	Score	Taux de classification cumulé (%)
666.55-1-Area	3-methyl butanal	0.537943	62,96
744.64-2-Area	3- methyl butanal / 2-methyl butanal	0.494428	62,96
642.00-2-Area	butanal	0.416727	62,96
1209.08-2-Area	Ethyl cyclohexane carboxylate	0.322068	81,48
895.97-1-Area	heptanal	0.304317	88,89
1111.46-1-Area	Ethyl cyclohexane carboxylate	0.241995	85,18
561.11-1-Area	butanal	0.193905	85,18

TABLE 3.19 – Résultats de la sélection des composés organiques volatils par MEM-BAS pour de la cuisse de poulet avec la méthode d'extraction d'Alpha MOS

### 3.3.9 Bilan sur les composés organiques volatils

Il a été établi dans cette partie quels composés volatils permettaient pour chaque aliment et chaque type de mesure de mieux détecter par classification, les états de cuisson. Les données d'Alpha MOS sont caractérisables en terme de ressenti humain, ce qui permet de valider dans une certaine mesure la pertinence des résultats. Néanmoins dans le cadre du projet, il est nécessaire de réduire le nombre d'attributs sélectionnés au minimum afin de permettre la construction d'un nombre de capteurs limité pour rentrer dans les besoins de l'industrie. Il s'agit donc de retenir de cette sélection quelques molécules issues des deux types de mesure.

En ce qui concerne les poissons, on se limitera à trois molécules par type de prélèvement. Les poissons ont beaucoup de molécules en commun, on prend donc le triplet 2-decanone, pentanal et furfural pour les mesures de l'ANSES et le triplet butanal-3-methyl, butanal-2-methyl et butanal pour les mesures d'Alpha MOS. Ces choix mènent aux résultats de la table 3.20.

	<b>ANSES</b>	<b>Alpha MOS</b>
Cabillaud	100 %	74 %
Saumon	100 %	75.5 %
Truite	100 %	57 %

TABLE 3.20 – Taux de classification en utilisant les composés retenus pour les poissons

Pour les viandes, cinq molécules sont retenues pour les données de l'ANSES, le 2,4-octadienal,(E,E)-, le 2-undecenal, le p-xylène, l'acide pentanoïque et le 2-decanone (commun avec le poisson). Coté Alpha MOS, quatre molécules sont sélectionnées : le 3-methyl butanal, la pyrazine, le 2-butanol et l'hexanal. Les taux de réussite de classification sont donnés dans la table 3.21.

	<b>ANSES</b>	<b>Alpha MOS</b>
Rosbeef	100%	66%
Veau	89%	85%
Porc	100%	74%
Blanc de poulet	100%	100%
Cuisse de poulet	100%	63%

TABLE 3.21 – Taux de classification en utilisant les composés retenus pour les viandes

Il n'y a pas de molécules communes qui ressortent de l'analyse entre les deux types de mesures (cumule sur la cuisson dans le cas de l'ANSES et récupération en fin de cuisson dans le cas d'Alpha MOS). Ceci est dû grandement à la méthode de prélèvement, ce qui rend la prise en compte des deux techniques importante pour avoir une information plus complète avant la création des capteurs à intégrer dans le prototype.

### 3.4 Mesures spectrales suite à une excitation par lumière fluorescente

Les mesures de fluorescence ont été réalisées par l'équipe Procédés Alimentaires et PhysicoChimie (PAPC) de l'unité de recherche Procédés Alimentaires et Microbiologiques (PAM). Les aliments sont cuits jusqu'à atteindre la température à cœur définie par le protocole de cuisson (voir table 3.1), puis sont sortis du four et émiettés à l'aide d'un couteau.

Pour chaque cuisson, trois extraits émiettés sont conservés pour réaliser une mesure. L'appareil utilisé est un Perkin Elmer LS50 réglé pour émettre une lumière dans l'ultraviolet avec une longueur d'onde de 380 nm. Le spectre de lumière mesuré est fixé entre 450 nm et 700 nm. Il s'agit d'un choix préliminaire pour restreindre la plage de mesures à un ensemble qui pourra être acquis avec un capteur intégré au four avec un coût raisonnable.

Neuf mesures ont été effectuées par aliment, comme chaque aliment est émetté trois fois cela fait un total de dix-huit mesures à traiter par type d'aliment. Un exemple de résultats de mesures moyennées sur le cabillaud est montré en figure 3.2.

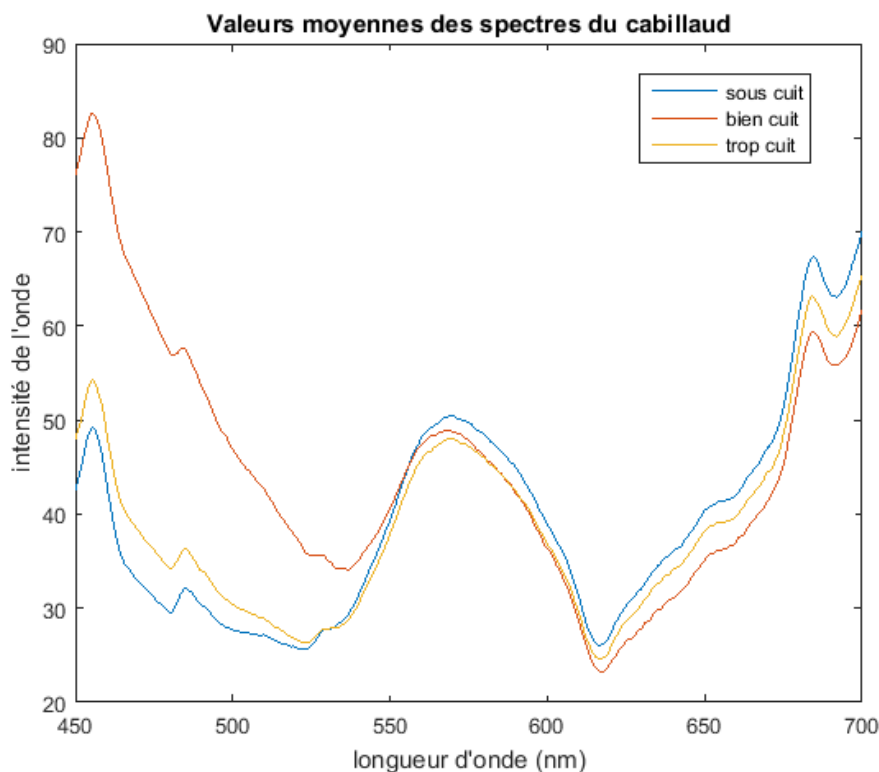


FIGURE 3.2 – Valeurs moyennes des différentes classes des spectres par fluorescence du cabillaud

Les mesures ne sont pas traitées à l'état brut mais avec la suppression des fonds continus, ce qui consiste à mettre le minimum du spectre à zéro (figure 3.3). Cette transformation est effectuée pour supprimer des artéfacts produits par les oscillations de la lampe et pour éliminer des perturbation provenant de l'environnement. L'ensemble des valeurs mesurées aux différentes longueurs d'onde (tous les 0,5 nm) représente 501 descripteurs.

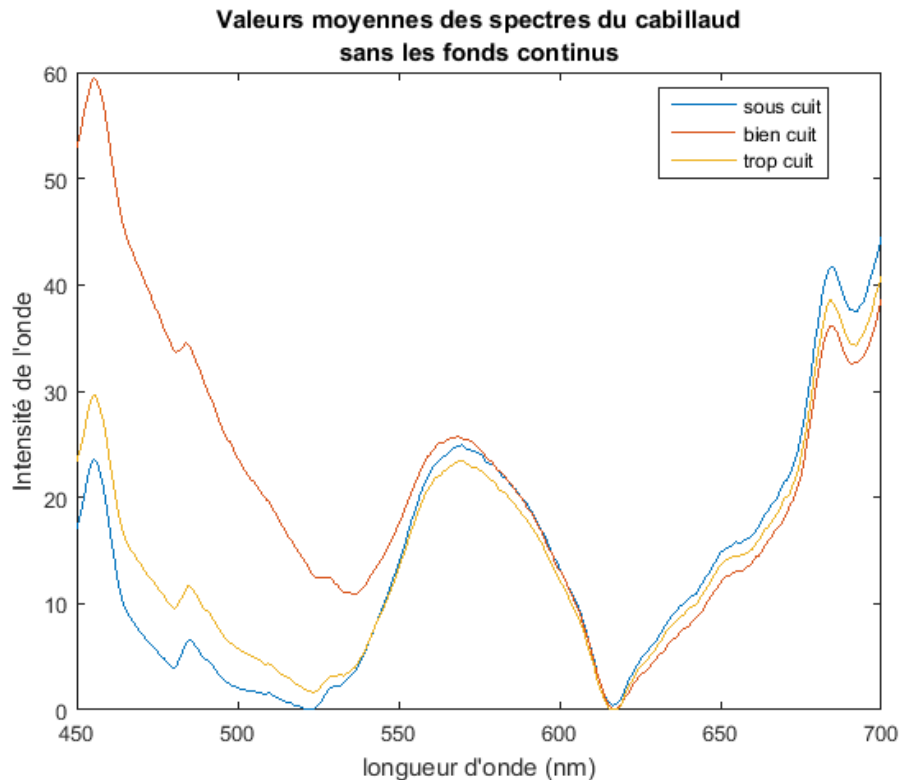


FIGURE 3.3 – Valeurs moyennes des différentes classes des spectres par fluorescence du cabillaud avec suppression des fonds continus

### 3.4.1 Cabillaud

La sélection consiste donc à déterminer dans un premier temps les poids (pertinences) de toutes ces longueurs d'onde. Dans le cas du cabillaud, il en résulte la courbe (figure 3.4) représentant les valeurs de ces poids en fonction des longueurs d'onde.

Pour réaliser une classification sans erreur des échantillons de cabillaud (100 % de réussite), il faut dans le cas des tests présents utiliser la mesure à 627 nm et ainsi qu'une autre de manière indifférenciée sur la bande entre 450 nm et 520 nm.

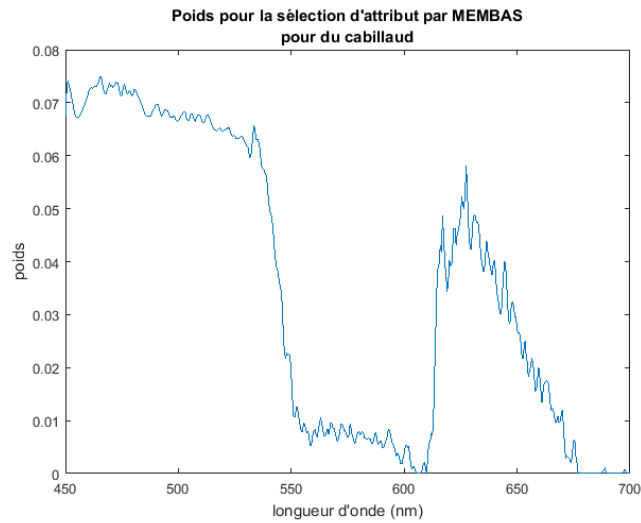


FIGURE 3.4 – Poids résultant de la sélection d’attributs effectuée avec MEMBAS pour du cabillaud après excitation par lumière fluorescente

### 3.4.2 Saumon

L’allure générale des poids pour le saumon est proche de celui du cabillaud, ce sont toujours les longueurs d’onde les plus basses qui ont le plus d’importance dans la classification des états de cuisson. Le taux de bonne classification maximal est de 92 % avec les trois longueurs d’ondes 490 nm, 540 nm et 674 nm. La confusion dans ce cas est due à des erreurs entre les classes « sous cuit » et « bien cuit », les échantillons de la classe « trop cuit » étant tous bien classés.

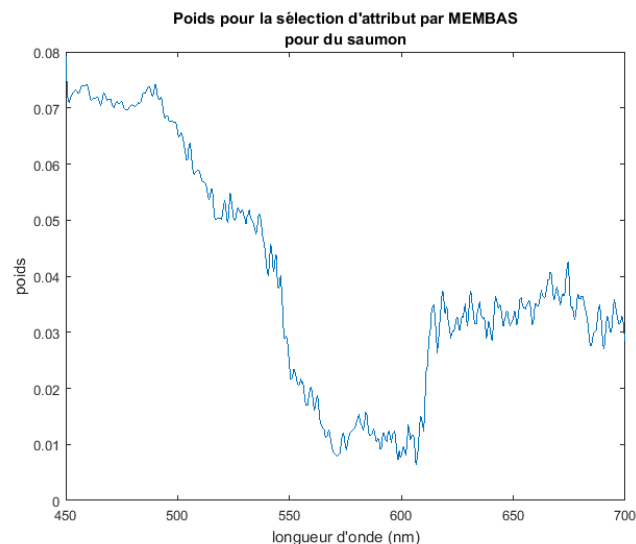


FIGURE 3.5 – Poids résultant de la sélection d’attributs effectuée avec MEMBAS pour du saumon après excitation par lumière fluorescente

### 3.4.3 Truite

On observe sur la figure 3.6 un ensemble de pics discontinus autour de 625 nm. L'absence de continuité dans les résultats indique que ces points ne sont pas fiables pour la classification, la variation d'intensité brutale dans un spectre mesuré est très improbable. De plus, il est impossible de mesurer l'intensité d'une émission lumineuse avec un écart de la largeur des pics. Seront alors retenues comme longueurs d'onde 450 nm, 574 nm et 657 nm pour un taux de classifications réussies de 89 %.

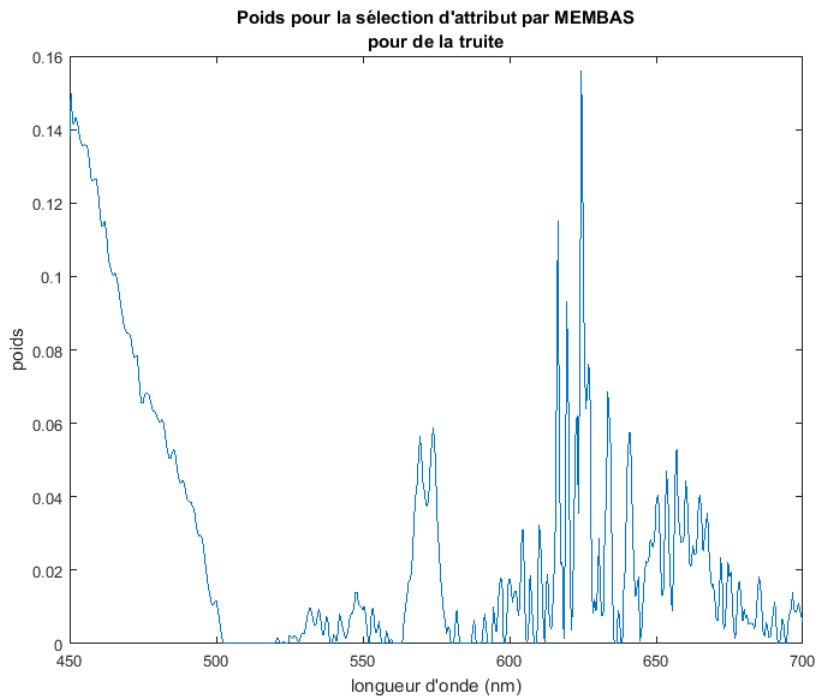


FIGURE 3.6 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour de la truite après excitation par lumière fluorescente

### 3.4.4 Rosbif

Le maximum de réussite (85 %) est atteint avec les longueurs d'ondes 518 nm, 608 nm et 635 nm. Il y a très peu de zones avec des scores vraiment bas, ce qui permet des mesures avec un filtre large en termes de longueurs d'onde pour un capteur intégré.

### 3.4.5 Rôti de porc

Les états de cuisson du rôti de porc sont assez compliqués à classer à l'aide des mesures de fluorescence, puisque que le pourcentage de mesures correctement classées s'élève à 74% en utilisant les longueurs d'onde 570 nm, 600 nm et 640 nm. Un essai de classification sur SVM linéaires a été effectué pour observer si les



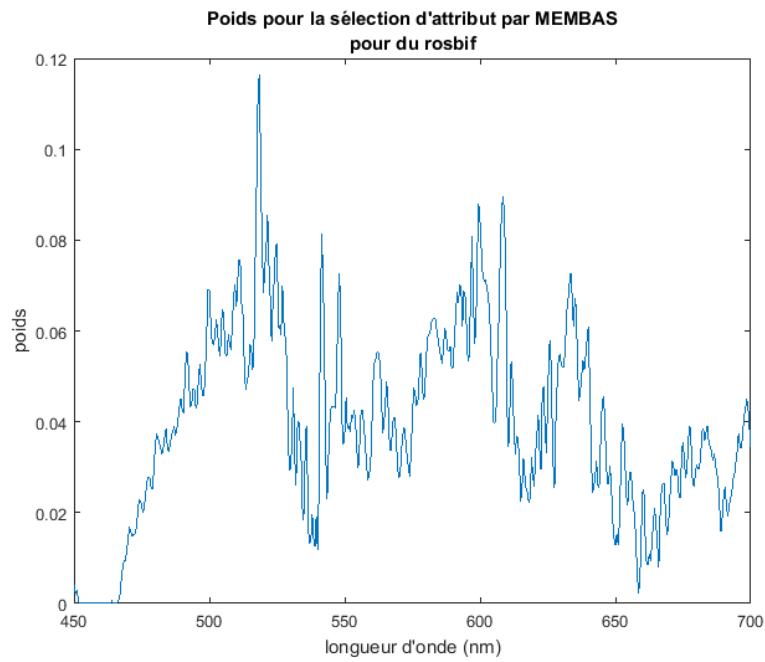


FIGURE 3.7 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du ros bif après excitation par lumière fluorescente

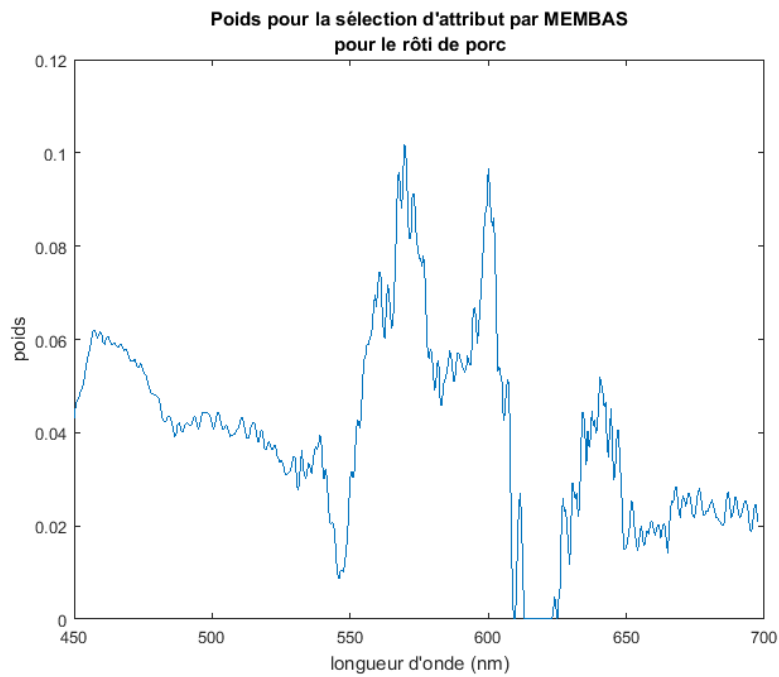


FIGURE 3.8 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de porc après excitation par lumière fluorescente

résultats pouvaient être améliorés, ce n'est pas le cas puisque cet algorithme classe convenablement 56% des échantillons, ce qui est encore moins bon que ce que nous obtenons avec la méthode MEMBAS.

### 3.4.6 Rôti de veau

Les poids déterminés dans le cas de la cuisson des rôtis de veau présentent plusieurs discontinuités fortes au niveau des pics d'intérêt. On exclura de la sélection les longueurs d'onde immédiatement au bord des discontinuités et seront conservées celles qui ont au minimum 5 nm d'écart. Un taux de précision de 85 % pour la classification est vérifié avec les longueurs d'onde 528 nm et 623 nm.

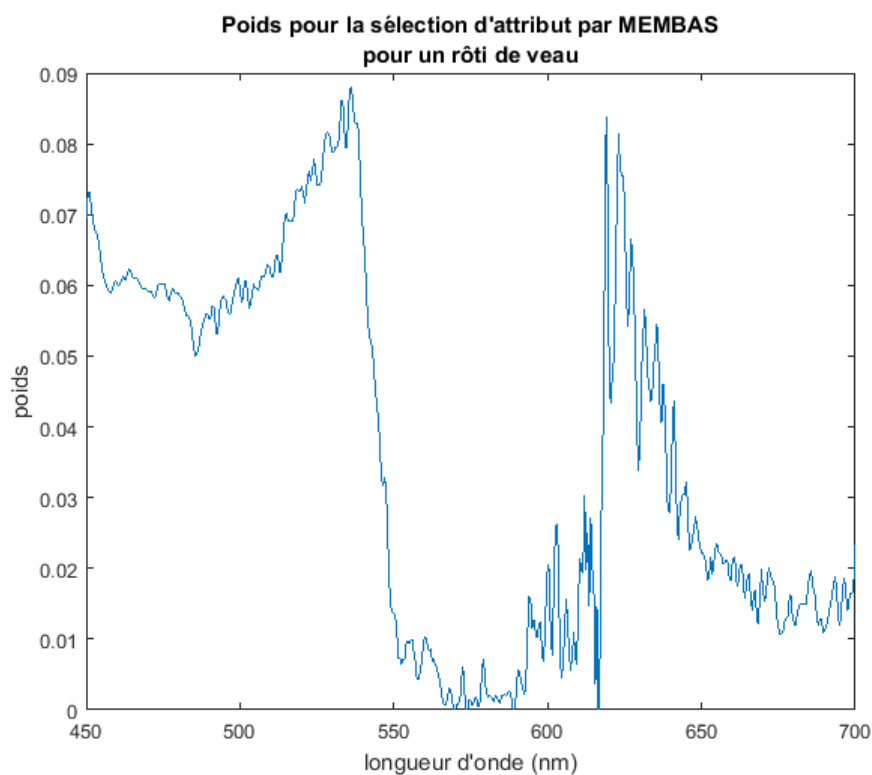


FIGURE 3.9 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de veau après excitation par lumière fluorescente

### 3.4.7 Blanc de poulet

Suite à des données incomplètes, l'analyse sur les données de cuisson du blanc de poulet ont été effectuées sur l'intervalle 500-700 nm.

Le taux de classification maximal (74 %) est obtenu avec la longueur d'onde 500 nm, les erreurs étant majoritairement au niveau de la classe « trop cuit » qui n'est pas reconnue.

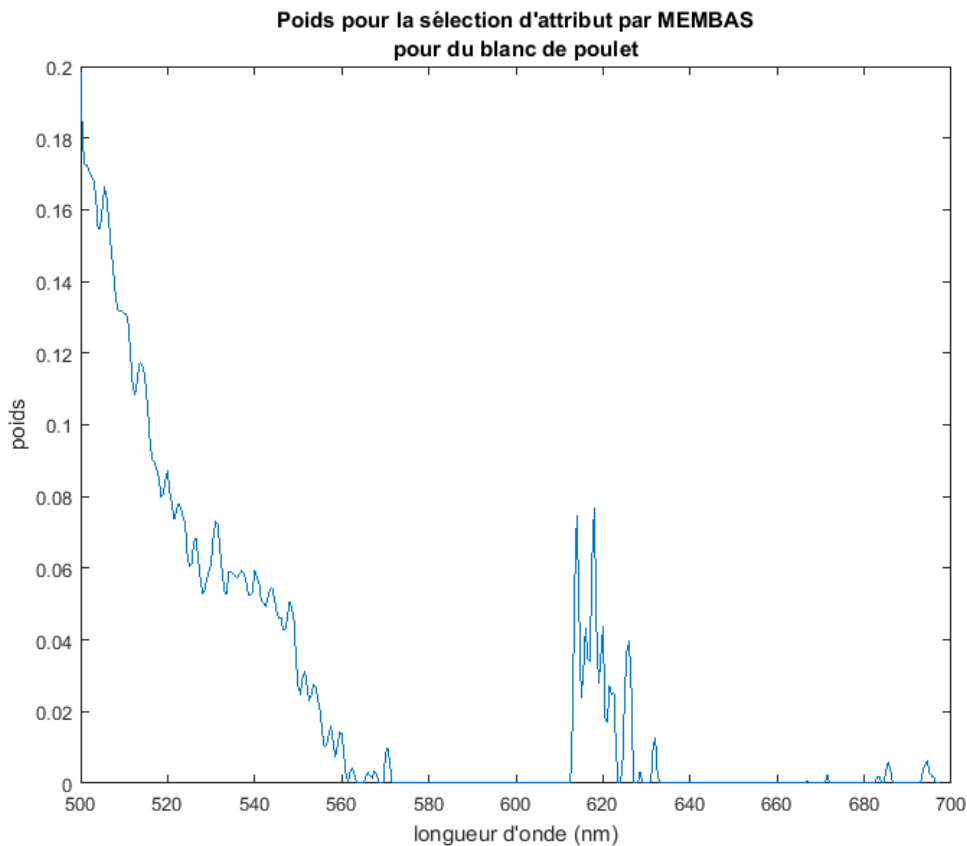


FIGURE 3.10 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du blanc de poulet après excitation par lumière fluorescente

### 3.4.8 Cuisse de poulet

Le taux de classifications réussies est de 87 % avec les longueurs d'onde 479 nm et une moyenne autour de la zone de pic à 620 nm (entre 615 nm et 624 nm). Cette bande du spectre semblait peu fiable à priori étant donné ces fortes variations, mais après vérification sur la valeur moyenne de la zone, elle semble porter une information pour résoudre le problème de classification.

### 3.4.9 Bilan sur l'analyse par fluorescence

La table 3.22 montre un résumé sur la sélection par l'algorithme MEMBAS des longueurs d'onde pour le suivi de cuisson par fluorescence. Il est à présent utile de chercher à réduire cet ensemble à un nombre de longueurs d'ondes plus petit afin de pouvoir dimensionner les filtres du capteur de fluorescence. Pour cela on fixe les contraintes suivantes : chaque type d'aliment doit avoir au moins une de ses longueurs d'onde représentée dans un regroupement et le nombre maximal de groupes est fixé à cinq.

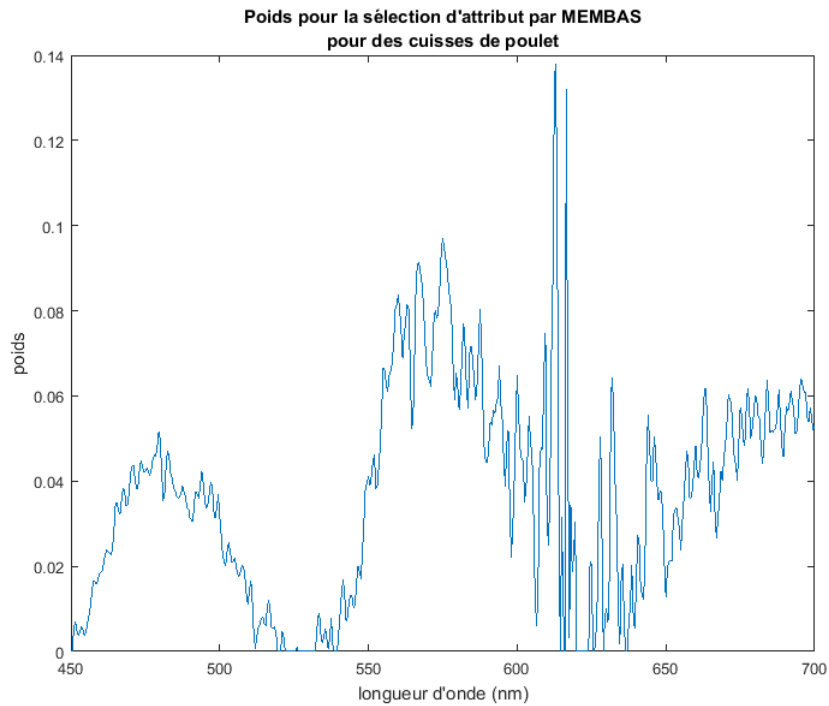


FIGURE 3.11 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour des cuisses de poulet après excitation par lumière fluorescente

Cabillaud	Saumon	Truite	Rosbif	Rôti de porc	Rôti de veau	Blanc de poulet	Cuisse de poulet
450-520	490	450	518	570	528	500	479
627	540	574	608	600	623		620
	674	657	635	640			

TABLE 3.22 – Tableau récapitulatif des longueurs d'onde sélectionnées pour l'analyse par fluorescence

Les longueurs d'onde retenues sont :

- 450 nm (cabillaud, saumon, truite)
- 514 nm (blanc de poulet, bœuf, rôti de veau)
- 570 nm (truite, rôti de porc)
- 600 nm (bœuf, rôti de porc, cuisse de poulet)
- 650 (saumon, truite, rôti de porc, rôti de veau)

Les résultats de classification avec les longueurs d'onde de la sélection finale sont donnés dans la table 3.23.

Cabillaud	Saumon	Truite	Rosbif	Rôti de porc	Rôti de veau	Blanc de poulet	Cuisse de poulet
63 %	85,2 %	74 %	80 %	66 %	70 %	74 %	81 %

TABLE 3.23 – Taux de classification pour l’analyse par fluorescence avec les cinq longueurs d’onde retenues

### 3.5 Mesures de réflexion dans le visible et le proche infrarouge

Tout comme dans le cas des mesures de fluorescence, des mesures dans le visible et le proche infrarouge ont été réalisées sous lumière blanche afin de mesurer le spectre réfléchi par les différents aliments. La mesure du spectre est effectuée pour des longueurs d’onde entre 350 nm (zone dans l’ultraviolet) et 1000 nm (proche infrarouge). Le visible commençant aux alentours de 400 nm, on ne prendra pas en compte les potentiels résultats entre 350 nm et 400 nm. L’ensemble des informations pour chaque cuisson représentait 651 couples (intensité, longueur d’onde) . Le principe de la sélection des longueurs d’onde les plus pertinentes dans le cas de la réflexion dans le visible et l’infra-rouge est identique à celui exposé précédemment dans le cas de la fluorescence.

[Liu 2000] montre qu’il peut être intéressant d’observer les longueurs d’onde à 544 nm et 580 nm, plus particulièrement le ration entre leur intensités  $ratio = \frac{I_{580}}{I_{544}}$  qui permet de détecter une évolution de la coloration rouge vif vers rouge foncé. Cependant ce ratio n’a pas montré d’information pertinente dans notre cas d’étude (exemple sur du bœuf en figure 3.12). Les résultats sont trop dispersés pour pouvoir en tirer des informations fiables.

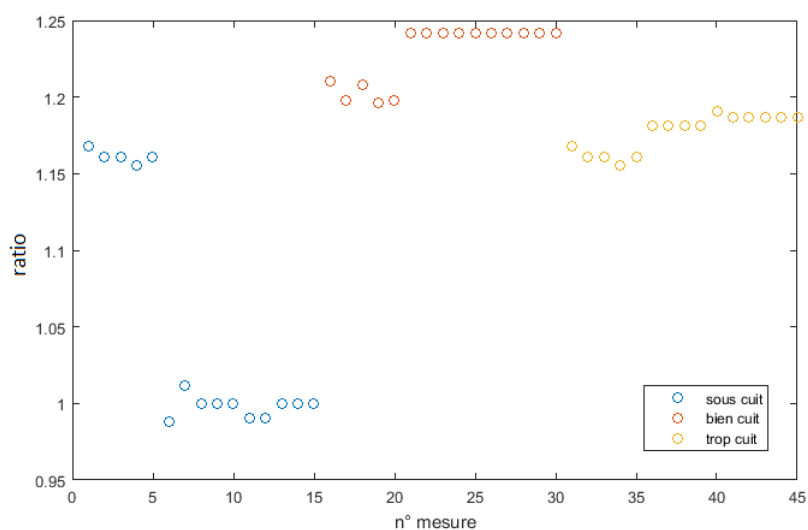


FIGURE 3.12 – Ratio entre les longueurs d’onde 580 nm et 544 nm pour du rosbif

### 3.5.1 Cabillaud

Comme dans le cas de la fluorescence, après utilisation de l'algorithme MEMBAS, on obtient une valeur de poids par longueur d'onde, que l'on peut représenter sur un graphe, comme la figure 3.13 dans le cas du cabillaud.

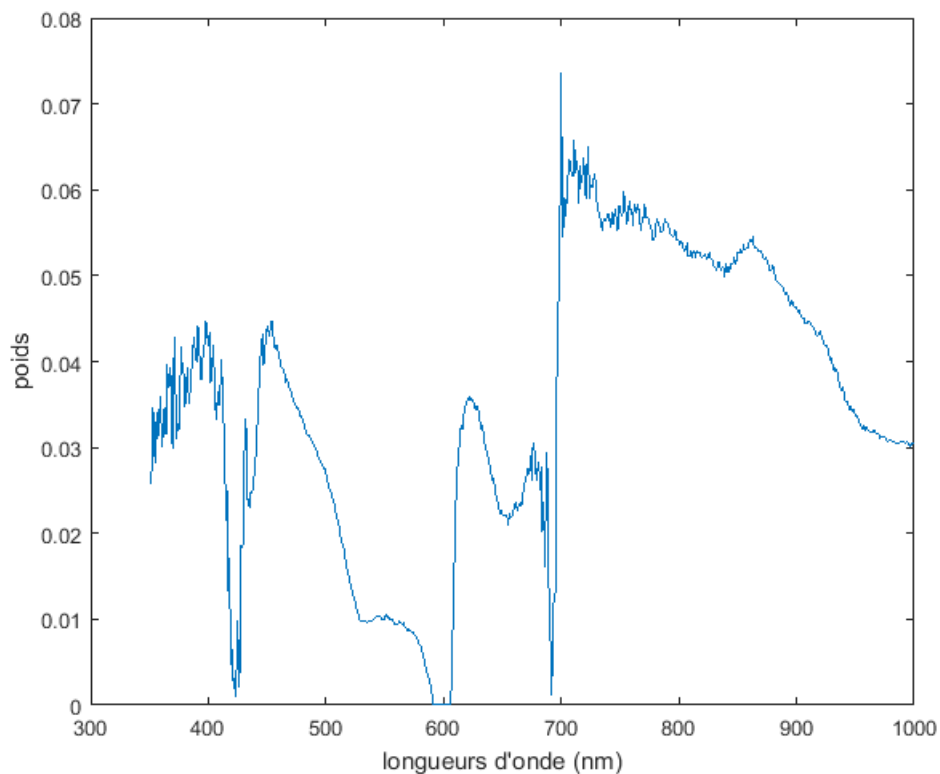


FIGURE 3.13 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du cabillaud sous lumière blanche

Le pic autour de 700 nm n'est en réalité pas porteur d'information, l'écart d'amplitude des mesures autour de cette fréquence est très faible, ce qui se voit avec une information quasi-nulle juste à côté. On retiendra plutôt les longueurs d'ondes suivantes : 462 nm, 551 nm et 899 nm avec lesquelles on obtient 100 % de classifications réussies sur ces données.

### 3.5.2 Saumon

Dans le cas du saumon, avec les mesures à 523 nm et 853 nm, on obtient un taux de classification de 100 % de ces données.

### 3.5.3 Truite

Les meilleurs résultats sont obtenus pour les longueurs d'onde 419 nm et 443 nm. Ces longueurs d'onde sont très proches (environ 20 nm d'écart), il sera dans

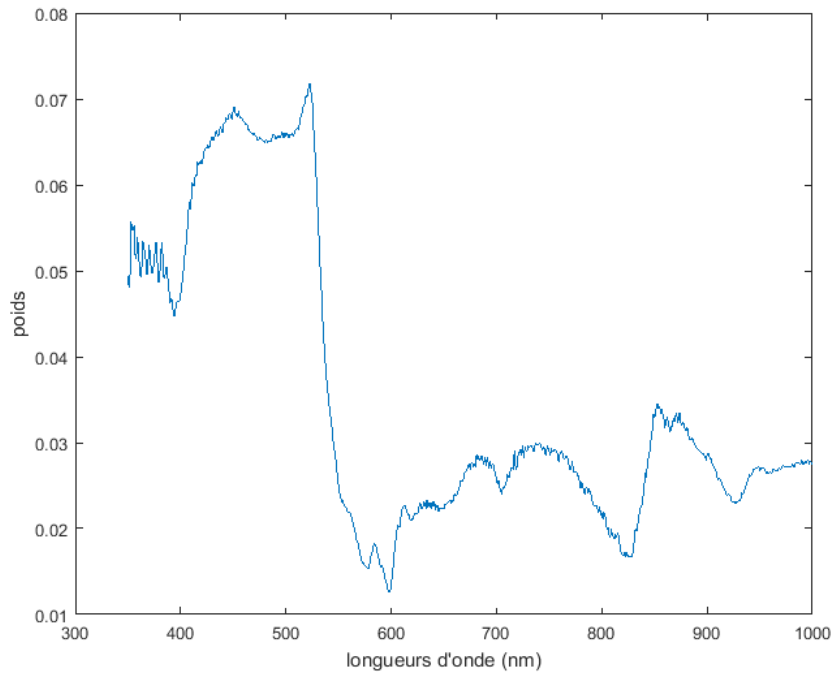


FIGURE 3.14 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du saumon sous lumière blanche

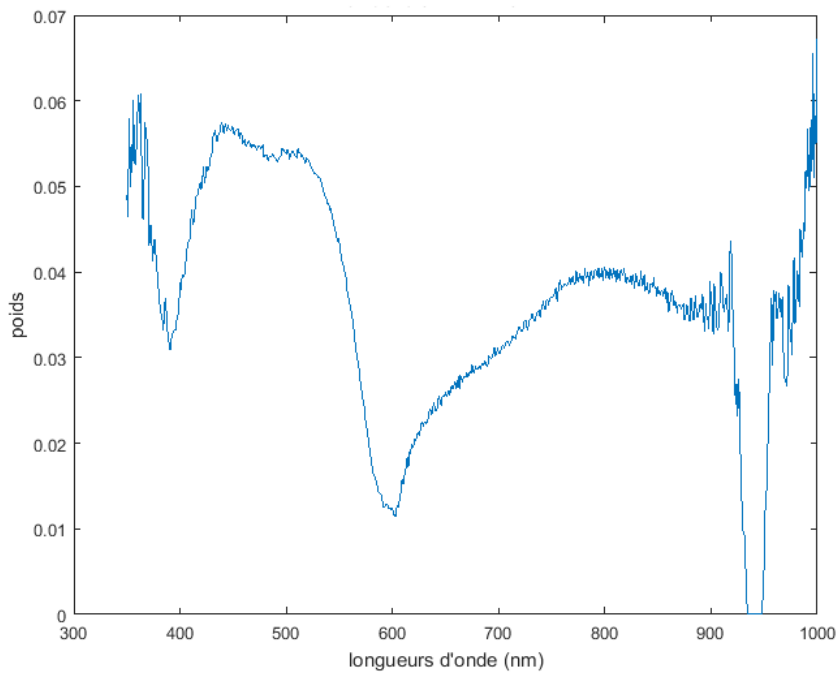


FIGURE 3.15 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour de la truite sous lumière blanche

tous les cas difficile de les mesurer séparément. Une approche avec un filtre gaussien centré entre les deux (430 nm avec 20 nm d'écart-type) donne un résultat de 82 % de bonnes classifications alors que l'utilisation uniquement des deux longueurs d'onde précédentes donne des résultats inférieurs à 70 %.

#### 3.5.4 Rosbif

On retiendra 438 nm, 539 nm et 799 nm qui donnent un taux de classification de 89 %. En réalité, on obtient le même taux de classification avec uniquement 438 nm mais il est préférable de retenir ces trois valeurs car la mesure à 438 nm est proche de la plus petite longueur d'onde mesurable et donc potentiellement compliquée à obtenir.

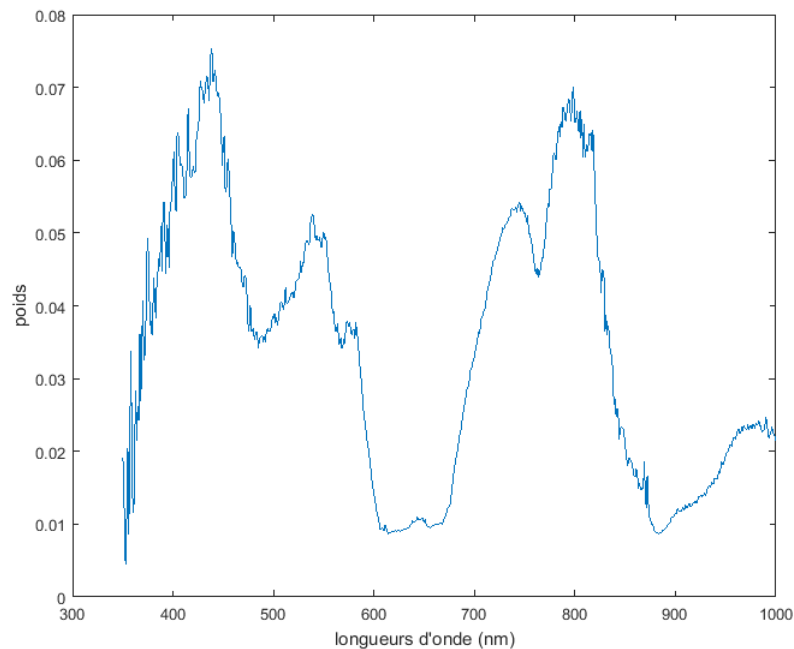


FIGURE 3.16 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rosbif sous lumière blanche

#### 3.5.5 Rôti de porc

Le pic à 350 nm n'est pas retenu pour deux raisons : il est hors de la zone exploitable (inférieur à 400 nm) et il est issu d'un artefact des fichiers de la base de données. En conservant uniquement l'information à 650 nm, on classe correctement ces mesures dans 84 % des cas.

#### 3.5.6 Rôti de veau

Avec uniquement la longueur d'onde 420 nm, on parvient à classer correctement l'intégralité des échantillons.



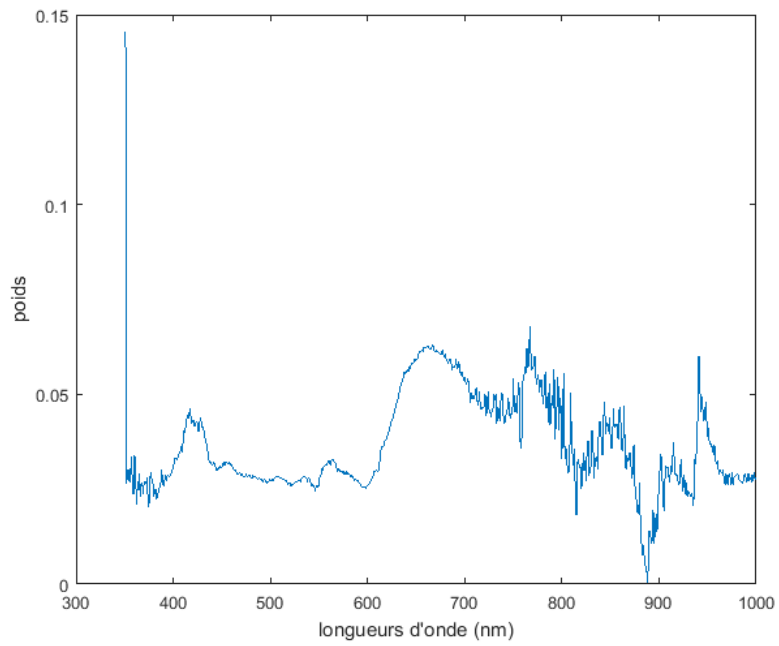


FIGURE 3.17 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de porc sous lumière blanche

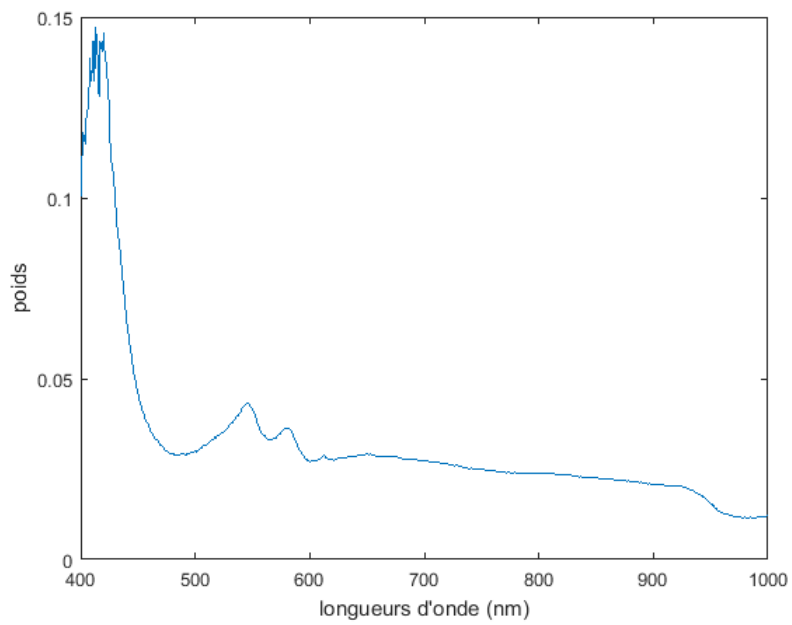


FIGURE 3.18 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du rôti de veau sous lumière blanche

### 3.5.7 Blanc de poulet

En utilisant les trois longueurs d'ondes 717 nm, 789 nm et 820 nm, on obtient un taux de classification de 92 %.

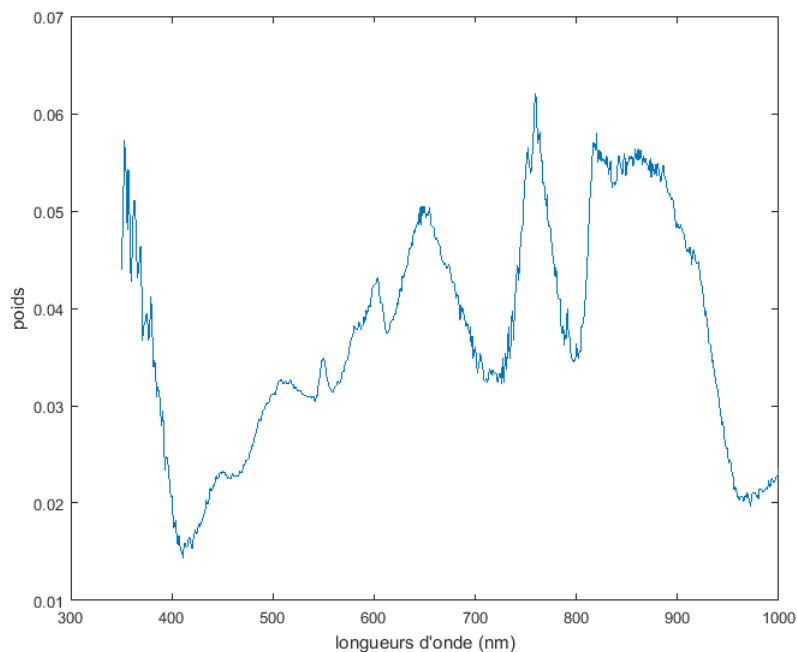


FIGURE 3.19 – Poids résultant de la sélection d'attributs effectuée avec MEMBAS pour du blanc de poulet sous lumière blanche

### 3.5.8 Cuisse de poulet

Les longueurs d'ondes 550 nm, 738 nm et 948 nm permettent d'obtenir un taux de classification de 100 %.

### 3.5.9 Bilan sur les spectres dans le visible et le proche infrarouge

L'ensemble des longueurs d'onde sélectionnées à l'aide de l'algorithme MEMBAS sont regroupées dans la table 3.24. Comme cela a été fait pour les composés volatils et les mesures de fluorescence, on cherche à réduire le nombre de variables à observer pour réaliser la classification des états de cuisson.

Une fois les résultats regroupés, les longueurs d'onde suivantes sont obtenues :

- 420 nm (rôti de veau et rosbif)
- 514 nm (cabillaud, saumon et truite)
- 550 nm (cuisse de poulet)
- 615 nm (truite)
- 650 nm (rôti de porc)
- 680 nm (cabillaud)
- 720 nm (cuisse de poulet et blanc de poulet)

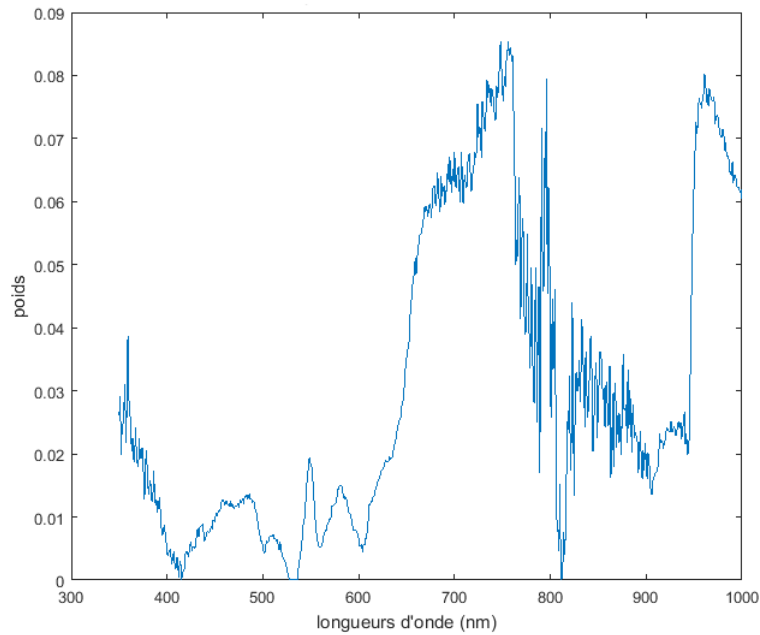


FIGURE 3.20 – Poids résultant de la sélection d’attributs effectuée avec MEMBAS pour des cuisses de poulet sous lumière blanche

	Longueurs d’onde sélectionnées (nm)		
Cabillaud	495	680	937
Saumon	523	853	
Truite	514	539	615
Rosbif	438	539	799
Veau	420		
Porc	650		
Blanc de poulet	717	789	820
Cuisse de poulet	550	738	948

TABLE 3.24 – Tableau récapitulatif des longueurs d’onde sélectionnées pour l’analyse des spectres dans le visible et le proche infrarouge

- 790 nm (rosbif et blanc de poulet)
- 853 nm (saumon)

Le sous ensemble choisi conduit aux résultats de classification donnés dans la table 3.25.

Cabillaud	Saumon	Truite	Rosbif	Rôti de porc	Rôti de veau	Blanc de poulet	Cuisse de poulet
89 %	89 %	80 %	89 %	84 %	100 %	89 %	89 %

TABLE 3.25 – Taux de classification pour l’analyse des spectres dans le visible et le proche infrarouge avec les neuf longueurs d’onde retenues

### 3.6 Bilan de la sélection d’attributs pour le suivi de l’état de cuisson d’aliments

Dans ce chapitre, ont été utilisées des méthodes de sélection afin de déterminer pour plusieurs types de mesures physiques quels étaient les meilleurs attributs pour appliquer un algorithme de classification afin de déterminer l’état de cuisson d’aliments. Il faut toutefois prendre en compte que les résultats donnés ici n’offrent pas une connaissance absolue et fiable étant donné le faible nombre d’expériences qui ne permet pas une généralisation statistique, particulièrement dans le cas des mesures de l’ANSES pour lesquelles on ne dispose que de neuf mesures au maximum. Cela reste néanmoins une approche intéressante et novatrice. Les résultats globaux de ce chapitre sont donnés dans la table 3.26.

	Mesures de composés organiques volatils de l'ANSES	Mesures de composés organiques volatils d'Alpha MOS	Longueurs d'onde issues des mesures du visible et proche infra-rouge	Longueurs d'onde des mesures en réaction à une lumière fluorescente
<b>Cabillaud</b>	2-Decanone Pentanal Furfural	Butanal-3-methyl Butanal-2-methyl Butanal	514 nm 680 nm	450 nm
<b>Saumon</b>	2-Decanone Pentanal Furfural	Butanal-3-methyl Butanal-2-methyl Butanal	514 nm 853 nm	450 nm 650 nm
<b>Truite</b>	2-Decanone Pentanal Furfural	Butanal-3-methyl Butanal-2-methyl Butanal	514 nm 615 nm	450 nm 570 nm 650 nm
<b>Rosbif</b>	Pentanoic acid	hexanal	420 nm 550 nm 790 nm	514 nm 600 nm
<b>Veau</b>	2-Undecenal 2-Decanone	pyra- zine/pyridine	420 nm	514 nm 650 nm
<b>Porc</b>	2,4- Octadienal, (E,E)- 2-Decanone	pyra- zine/pyridine 2-butanol/2- butanone	650 nm	570 nm 600 nm 650 nm
<b>Cuisse de poulet</b>	2,4- Octadienal, (E,E)-	3-methyl butanal	550 nm 720 nm	600 nm
<b>Blanc de poulet</b>	p-Xylene Pentanoic acid	3-methyl butanal 2-butanol/2- butanone	720 nm 790 nm	514 nm

TABLE 3.26 – Bilan global de la sélection d'attributs pour le suivi de la cuisson d'aliments

# Modélisation des incertitudes pour la classification par l’algorithme LAMDA

---

## Sommaire

<b>4.1</b>	<b>Modélisation d’incertitudes à l’aide d’intervalles . . . . .</b>	<b>63</b>
4.1.1	Utilisation d’intervalles non-flous . . . . .	64
4.1.2	Intervalles flous pour la modélisation de lois normales . . . . .	65
<b>4.2</b>	<b>Utilisation d’agrégation pondérée dans la méthode LAMDA</b>	<b>69</b>
<b>4.3</b>	<b>Bilan sur la prise en compte d’incertitudes . . . . .</b>	<b>69</b>

Les mesures issues du projet sont entachées de bruit et d’incertitudes pour plusieurs raisons : bruit thermique, variations dues aux différences entre les aliments, composés résiduels dans le four, biais des capteurs, etc. Nous présentons dans ce chapitre deux méthodes que nous avons mises en place afin d’essayer de prendre en compte ces incertitudes. Ces méthodes ont été développées comme des extensions de la méthode LAMDA, présentée dans le chapitre 2. La première solution est la mise en place d’intervalles flous pour modéliser une erreur gaussienne autour d’une donnée, la seconde est l’introduction d’une agrégation pondérée pour donner plus d’importance à certains attributs jugés plus fiables lors du regroupement des degrés d’appartenance (DAM).

## 4.1 Modélisation d’incertitudes à l’aide d’intervalles

Ont déjà été introduits dans le chapitre 2 (page 19), l’algorithme LAMDA et son principe de fonctionnement basé sur la logique floue. Cet algorithme intègre déjà le concept d’intervalle pour permettre de modéliser des ensembles de valeurs. Néanmoins, même si la formulation théorique existe, l’extension au traitement des intervalles flous, n’a jamais été réellement testée. L’incertitude de mesure dans le cas des procédés physiques est dans la grande majorité des cas une erreur qui suit une loi normale, modélisable par une fonction gaussienne.

Afin de représenter cette incertitude, nous avons considéré deux approches :

- la représentation de l’erreur par un intervalle non-flou à l’aide d’intervalle de confiance

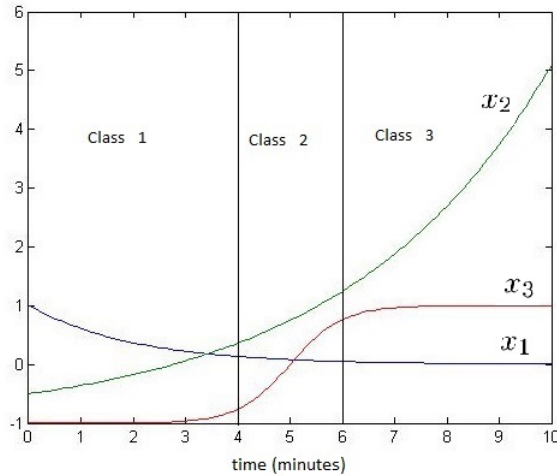


FIGURE 4.1 – Fonctions servant de base aux tests des fonctions d’appartenance par intervalles

- la modélisation par un intervalle flou de la fonction gaussienne.

Tous ces mécanismes sont introduits dans [Monrousseau 2015].

Afin d’illustrer de manière simple ces deux concepts, nous avons choisi un exemple théorique. Il s’agit d’un problème à trois classes utilisant des données générées par des fonctions mathématiques avec 101 échantillons. Les fonctions sont les suivantes et sont tracées sur la figure 4.1 :

- $x_1 = e^{-\frac{t}{2}}$
- $x_2 = \frac{1}{2}e^{\frac{t}{4}} - 1$
- $x_3 = \tanh(t - 5)$

Ces données ont été bruitées avec une variable aléatoire  $Y$  telle que  $\hat{x}_i = x_i + Y$ , avec  $\hat{x}_i$  qui représente la variable  $x_i$  bruitée. Pour tester l’efficacité des méthodes proposées sur cet exemple, une classification en validation croisée est a été réalisée en découpant les données en cinq sous-bases de même dimension, de manière aléatoire. De plus, le bruit ajouté étant par nature aléatoire, le test de classification est effectué 5000 fois avec du bruit généré à chaque essai pour obtenir un résultat statistiquement représentatif, les résultats donnés seront alors la valeur moyenne des réussites exprimées en termes de bonnes classifications.

#### 4.1.1 Utilisation d’intervalles non-flous

On propose dans un premier temps de tester l’impact de la modélisation d’incertitudes à l’aide d’intervalles non-flous. Le bruit  $Y$  ajouté au signal suit une loi uniforme bornée entre deux valeurs et l’intervalle créé autour de la valeur « mesurée »  $\hat{x}_i$  est un intervalle de confiance à 95 %, ce qui signifie que la donnée non bruitée  $x_i$  a 95 % de chance de se trouver dans l’intervalle.

Intervalles de la loi uniforme	[-0.3; 0.3]	[-0.5; 0.5]	[-2; 2]
Pourcentage de réussites moyen avec une fonction binomiale	89,9 %	84,7 %	79.6 %
Pourcentage de réussites moyen avec des intervalles de confiance	91.9%	89.8%	70.3%

TABLE 4.1 – Table des résultats de classification pour des données bruitées avec une loi uniforme

Afin d’appréhender l’apport d’une telle méthode, une étude de sensibilité pour différents paramètres caractérisant le bruit a été effectuée et les résultats ont été reportés dans la table 4.1.

On constate qu’avec un bruit faible ou moyennement fort (deux premiers cas), l’utilisation d’intervalles améliore les résultats de la classification par rapport à l’usage d’une fonction qui manipule uniquement des données scalaires. En revanche, lorsque le bruit est vraiment élevé, les intervalles produisent des résultats moins pertinents que la fonction scalaire.

#### 4.1.2 Intervalles flous pour la modélisation de lois normales

Le principal intérêt d’employer des intervalles flous est de chercher à modéliser des lois probabilistes autres que des lois de distribution uniforme. La loi la plus intéressante à modéliser est la loi normale, décrite par une densité de probabilité gaussienne  $f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\rho)^2}{2\sigma^2}}$  avec  $\rho$  l’espérance de la loi normale et  $\sigma$  son écart-type. Cependant, il est relativement compliqué de manipuler directement des gaussiennes (pas d’intégrale analytique, bornes à fixer, ...). Pour pallier cela, on propose d’effectuer une approximation de la fonction gaussienne par un intervalle triangulaire.

**Rappel :** on note  $\mu_X(x)$  l’appartenance floue de la variable  $x$  dans un ensemble  $X$

**Définition 1** *Un intervalle flou triangulaire  $X$  est exprimé par sa borne inférieure  $\underline{x}$  et sa borne supérieure  $\bar{x}$  et suit les propriétés :*

$$\left\{ \begin{array}{l} \mu_X(x) = \frac{2x - 2\underline{x}}{\bar{x} - \underline{x}}, \text{ pour } \underline{x} \leq x \leq \frac{\underline{x} + \bar{x}}{2} \\ \mu_X(x) = \frac{-2x + 2\bar{x}}{\bar{x} - \underline{x}}, \text{ pour } \frac{\underline{x} + \bar{x}}{2} < x \leq \bar{x} \\ \mu_X(x) = 0 \text{ pour toute autre valeur de } x \end{array} \right.$$

Dans l’objectif d’approximation recherché, le critère utilisé est celui de la « largeur à mi-hauteur » de la gaussienne. Cet indicateur dépend du paramètre  $\sigma$  et se calcule selon l’équation 4.1.



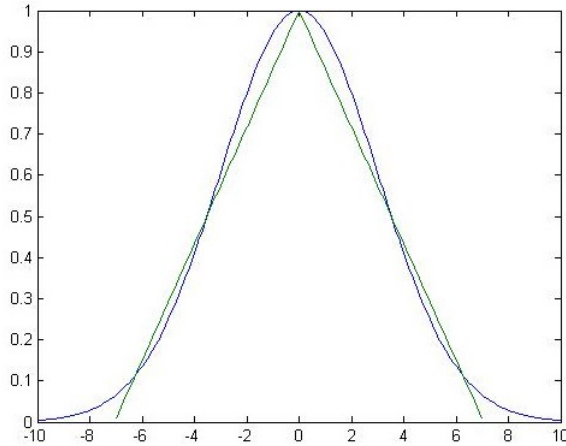


FIGURE 4.2 – Exemple d’approximation d’une gaussienne par un intervalle triangulaire

$$L(\sigma = 2\sqrt{2\ln(2)} \cdot \sigma) \quad (4.1)$$

La fonction gaussienne telle qu’elle est définie dans ce chapitre est modélisée par un intervalle flou triangulaire  $X$  de la manière suivante :

$$X = [\rho - 2\sqrt{2\ln(2)} \cdot \sigma, \rho + 2\sqrt{2\ln(2)} \cdot \sigma] \quad (4.2)$$

Un exemple d’approximation est donné en figure 4.2. Il est à noter que ce modèle ne représente pas la loi de probabilité, puisque le triangle a toujours une hauteur de 1 quel que soit l’écart-type. C’est principalement la répartition qui est modélisée dans ce cas.

Dans l’application à l’algorithme de classification LAMDA, l’analyse par intervalle implique la recherche de similarité entre deux intervalles. La méthode générale est donnée par l’équation 2.13 (chapitre 2, page 21). Toutefois, la mise en œuvre de cette équation avait été donnée dans le cas non-flou mais elle doit être précisée dans le cas des intervalles triangulaires, notamment la fonction distance et l’intersection. Pour mesurer la distance entre deux intervalles triangulaires, on choisit d’utiliser une distance telle que présentée par [Hsieh 1999] qui consiste à mesurer l’écart entre les centres des deux triangles. Soit pour deux intervalles triangulaires  $A$  et  $B$  :

$$\partial[A, B] = \left| \frac{\bar{a} + a - \bar{b} - b}{2} \right| \quad (4.3)$$

En ce qui concerne l’intersection entre deux intervalles  $A$  et  $B$  sur un univers  $V$ , le calcul de  $Inter(A, B) = \int_V \mu_{A \cap B}(x) dx$  est divisé en quatre cas. Pour simplifier les notations, on se place dans une situation où  $B$  est à droite de  $A$ , c’est-à-dire que  $\frac{\bar{a} + a}{2} \leq \frac{\bar{b} + b}{2}$ . L’ensemble des cas sont schématisés sur la figure 4.3.

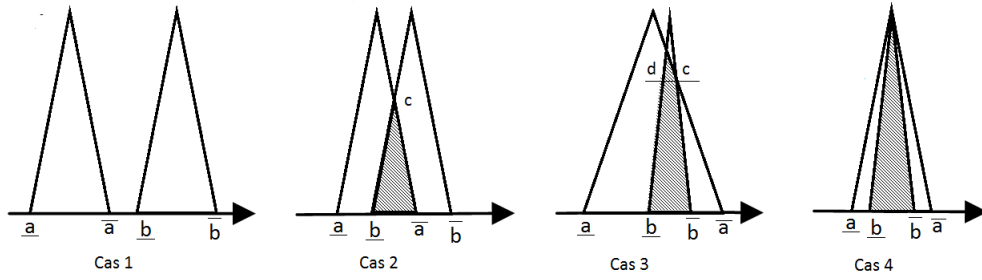


FIGURE 4.3 – Schémas des différents cas possibles d'intersection d'intervalles triangulaire

Cas 1 : intervalles disjoints

$$Inter(A, B) = 0 \quad (4.4)$$

Cas 2 : intervalles en jonction partielle

Dans ce cas, on recherche les angles des triangles à l'aide de leur sinus (voir figure 4.3), soit :

$$\sin(\widehat{bac}) = \frac{1}{\sqrt{1 + \frac{(\bar{a} - a)^2}{4}}} \quad \text{et} \quad \sin(\widehat{abc}) = \frac{1}{\sqrt{1 + \frac{(\bar{b} - b)^2}{4}}} \quad (4.5)$$

Connaissant alors les angles du triangle dont on cherche la surface, on en déduit :

$$Inter(A, B) = \frac{(\bar{a} - \bar{b})^2}{2} \cdot \frac{\sin(\widehat{bac}) \cdot \sin(\widehat{abc})}{\sin(\widehat{bac} + \widehat{abc})} \quad (4.6)$$

Cas 3 : intervalles en jonction totale

L'objectif dans ce cas est de calculer la surface de deux figures : un trapèze de base  $\underline{b}\bar{b}$  et le triangle de base  $\underline{c}\bar{d}$  (voir figure 4.3). En exploitant les équations 4.5, on peut déduire les angles  $\widehat{bbc}$  et  $\widehat{bac}$ , ce qui permet de déterminer  $\bar{bc}$  :

$$\bar{bc} = \frac{(\bar{a} - \bar{b}) \cdot \sin(\widehat{bac})}{\sin(\pi - \widehat{bbc} + \widehat{bac})} \quad (4.7)$$

De là, on peut déduire la longueur  $cd$  et la hauteur  $h$  du trapèze.

$$h = \bar{bc} \cdot \sin(\widehat{bbc}) \quad \text{et} \quad cd = \bar{b} - \underline{b} - \bar{bc} \cdot \cos(\widehat{bbc}) \quad (4.8)$$

$\sigma$	0,2	0,5	0,7	1
Pourcentage de réussites moyen avec une fonction binomiale	83,2 %	79,8 %	79,8 %	79,6 %
Pourcentage de réussites moyen avec des intervalles de confiance	86,8 %	82,5 %	77,2 %	71,3 %
Pourcentage de réussites moyen avec des intervalles triangulaires flous	93,1 %	84,5 %	79,3 %	74,8 %

TABLE 4.2 – Table des résultats de classification pour des données bruitées avec une loi normale

Il s'en suit le calcul final :

$$Inter(A, B) = \frac{(\bar{b} - \underline{b} + cd) \cdot h}{2} + \frac{cd^2}{2} \cdot \frac{\sin(\widehat{bbc}) \cdot \sin(\widehat{bac})}{\sin(\widehat{bac} + \widehat{bbc})} \quad (4.9)$$

Cas 4 : intervalles en jonction centrée

$$Inter(A, B) = \frac{\bar{b} - b}{2} \quad (4.10)$$

L'intérêt de cette méthode est de pouvoir traiter deux intervalles triangulaires. Dans le cadre de la méthode proposée ici, les références des classes qui permettront de calculer les degrés d'appartenance des différents échantillons sont construites comme des intervalles dont les bornes sont la moyenne arithmétique des bornes des intervalles qui composent la base d'apprentissage pour la classe considérée (de manière similaire au cas non-flou).

Un test de cette méthode a été effectué sur l'exemple présenté au début de la section. Les signaux sont bruités avec une variable aléatoire normale centrée avec plusieurs écart-types. La classification est réalisée avec trois méthodes : la fonction binomiale qui ne manipule que des scalaires, les intervalles non-flous sous forme d'intervalles de confiance à 95 % et les intervalles flous triangulaires. Les résultats sont donnés dans la table 4.2.

On constate, de façon analogue à l'exemple du cas non-flou, une amélioration des résultats avec les deux types d'intervalles par rapport à l'usage d'une fonction scalaire pour des bruits faibles mais néanmoins significatifs. Dès que le bruit devient assez fort, dans le cas présent s'il présente un écart-type supérieur à 0,7, l'utilisation d'intervalles n'améliore pas la classification et au contraire détériore les résultats. On remarque également que sur l'exemple, les intervalles flous proposent

des classifications en moyenne plus précises que les intervalles non-flous.

## 4.2 Utilisation d'agrégation pondérée dans la méthode LAMDA

Dans le cas de la méthode usuelle d'agrégation de la méthode LAMDA telle qu'elle est présentée dans le chapitre 2 (équation 2.18, page 22), celle-ci ne favorise pas un descripteur par rapport à un autre. Ceci traduit le fait que sur l'ensemble des degrés d'appartenance marginal (DAM) calculés, l'algorithme recherche uniquement les extrêmes par l'utilisation des opérateurs ET et OU de la logique floue. Récemment, un autre moyen de réaliser l'agrégation des DAM a été proposé par Elisa Roux dans le cadre de l'implémentation d'une structure multi-couches pour la méthode LAMDA [Roux 2015]. Le principe est d'effectuer une moyenne pondérée sur l'ensemble des DAM afin d'accorder plus ou moins d'importance à chaque attribut dans le calcul du score d'appartenance final. L'agrégation par moyenne pondérée se calcule selon l'équation 4.11.

$$DAG_c = \frac{\sum_{i=1}^I \omega_i DAM_{c,i}}{\sum_{i=1}^I \omega_i} \quad (4.11)$$

avec :

$DAG_c$  : degré d'appartenance global à la classe  $c$

$DAM_{c,i}$  : degré d'appartenance marginal du  $i$ ème attribut pour la classe  $c$

$I$  : nombre d'attributs

$\omega_i$  : coefficient de pondération lié au  $i$ ème attribut

Le choix des  $\omega_i$  peut-être fait selon plusieurs critères, ce dernier représentant d'une certaine manière la fiabilité de l'attribut auquel il est lié. On peut par exemple supposer qu'un expert humain pourrait définir, dans le cas d'un procédé particulier dont il aurait une connaissance a priori, les différents coefficients. Une autre manière d'envisager l'attribution des  $\omega_i$  serait l'usage d'une méthode algorithmique, notamment les algorithmes de sélection d'attributs font de très bons candidats, étant donné qu'ils reflètent l'importance d'un attribut dans la classification. On pense particulièrement ici à la méthode MEMBAS (section 2.2.3) qui produit des poids qui maximisent l'écart entre la classe à laquelle appartient l'échantillon étudié et la classe à laquelle ce dernier n'appartient pas avec le score le plus grand.

## 4.3 Bilan sur la prise en compte d'incertitudes

Il a été vu dans ce chapitre plusieurs moyens qui permettent de prendre en compte des incertitudes dans les données. Il a notamment été appliqué une transformation des données numériques scalaire en intervalles, flous ou non-flous, afin de modéliser des incertitudes ou du bruit sur les signaux d'entrée de l'algorithme de classification LAMDA. Une autre solution a été de considérer que chaque attri-

but possède une importance intrinsèque dans la classification et de faire ressortir cette importance au moyen d'un coefficient de pondération utilisé pour moyennner les scores d'appartenances.

Ces méthodes seront testées sur des données réelles dans le chapitre suivant afin d'évaluer si elles peuvent offrir des améliorations sur les résultats de la classification.

# Reconstitution de l'état au cœur d'aliments pendant une cuisson

---

## Sommaire

<b>5.1</b>	<b>Introduction</b>	<b>71</b>
<b>5.2</b>	<b>Présentation des prototypes et des protocoles</b>	<b>72</b>
5.2.1	Matériel	72
5.2.2	Paramètres pour la classification	73
<b>5.3</b>	<b>Étude par l'équation de la chaleur</b>	<b>74</b>
5.3.1	Présentation du modèle	74
5.3.2	Recherche de $r$ à posteriori	75
5.3.3	Recherche de $r$ à priori	77
5.3.4	Conclusion sur la méthode par calcul de l'équation de la chaleur	82
<b>5.4</b>	<b>Techniques de classification pour le suivi de cuisson</b>	<b>84</b>
5.4.1	Présentation	84
5.4.2	Analyses effectuées sur le prototype v1 première génération	84
5.4.3	Prototype v1 seconde génération	91
5.4.4	Analyses sur le prototype v2	92
<b>5.5</b>	<b>Reconstitution de la température à cœur à l'aide de réseaux de neurones artificiels</b>	<b>98</b>

---

## 5.1 Introduction

Suite à la sélection d'attributs permettant la classification de l'état à cœur de différents types d'aliments, les résultats ont été transmis à plusieurs partenaires du projet afin de les aiguiller dans la fabrication de capteurs pour effectuer le suivi de cuisson en temps réel. Ceci a mené à la fabrication de deux prototypes de four nommés « prototype v1 » et « prototype v2 ».

Le prototype v1 est conçu de manière à embarquer quatre technologies de capteur : un capteur de composés organiques volatiles (odeur), capteur de fluorescence, un capteur de mesure du spectre rayonné dans le visible et proche infrarouge et un capteur de température en surface de l'aliment. Cette version a été construite en

juillet 2015. Néanmoins suite à plusieurs contraintes techniques, il n'était pas possible de faire fonctionner simultanément les capteurs de fluorescence et visuel, et le capteur de température de surface n'était pas encore opérationnel. Ce dernier a été remplacé par une thermopile du commerce pour palier le problème. L'interaction entre les capteurs visuel et fluorescent a été résolue courant mai 2016, ce qui n'a pu donner des résultats sur la totalité des capteurs que tardivement dans le projet, l'objectif initial de ce prototype étant de réaliser une sélection d'attributs pour la classification de l'état à cœur avec tous les capteurs qui opèrent simultanément. Ceci a pu être fait mais sur un nombre très réduit de cuissons, ce qui donne des résultats qui sont plus une première estimation que de véritables résultats statistiques.

Le second prototype a vu le jour en février 2016. Il intègre une thermopile pour la mesure de la température de surface et un capteur de composés organiques volatiles. Le choix des capteurs intégrés dans ce four s'est fait principalement sur des raisons économiques et de possibilité d'industrialisation, les résultats du premier prototype étant incomplets à ce stade de développement. L'objectif sur ces prototypes est de reconstituer l'information sur l'état en température au cœur de l'aliment. Pour cela deux solutions ont été testées : soit calculer la température à cœur à partir de paramètres mesurés à l'aide d'un modèle physique ou de régression, soit effectuer une classification sur dix classes allant de cru à immangeable. On utilisera notamment comme modèle l'équation de la chaleur discrétisée en une dimension et la régression sera effectuée à l'aide de réseaux de neurones.

## 5.2 Présentation des prototypes et des protocoles

### 5.2.1 Matériel

Comme présenté en introduction, les fours construits dans le cadre du projet embarquent un ensemble de capteurs afin de mesurer différents phénomènes physiques. En voici une brève description :

- Capteur olfactif : mis en place par la société Alpha MOS, ce capteur est en réalité constitué d'un ensemble de trois micro-capteurs à oxydes métalliques (Metal Oxide Sensors ou MOS). Ce sont des capteurs capables de couvrir un large « spectre » de composés volatils, avec un niveau de détection proche de la perception humaine pour un large choix de produits chimiques. La détection est effectuée en mesurant la résistance des capteurs, la conductivité de l'oxyde métallique variant en présence de gaz combustibles et de molécules organiques. Pour chaque micro-capteur, trois modes opératoires ont été intégrés, ce qui porte le nombre de variables potentielles issues du capteur à douze.
- Capteur visuel : ce capteur est intégré par le LE2I et est composé d'un micro-capteur monochrome et de huit filtres optiques qui permettent de sélectionner une bande de longueurs d'ondes. Il comporte également une source de lumière de manière à couvrir tout le spectre observable.



FIGURE 5.1 – Les prototypes de four : à gauche le v2 et à droite le v1

- Capteur de fluorescence : l'excitation ultra-violette est assurée par une diode UV haute puissance de 405 nm. La détection est ensuite assurée par une photodiode avec un filtre optique passe-bande entre 437 nm et 800 nm.
- Thermopile : il s'agit d'un dispositif électronique qui génère une tension proportionnelle à un gradient de température. Il permet d'extraire la température d'un objet à partir de son rayonnement thermique.

Le four possède également diverses informations comme la température d'enceinte du four, l'état des relais d'alimentation, etc. L'ensemble des informations de tous les capteurs est envoyé à une carte électronique qui centralise les informations et effectue les calculs pour le traitement des données. Cette carte transmet ensuite les données à un ordinateur par liaison WIFI, ce dernier étant capable d'enregistrer les données de l'expérience.

### 5.2.2 Paramètres pour la classification

Durant ce chapitre, l'algorithme de classification LAMDA et l'algorithme de sélection MEMBAS seront utilisés à plusieurs occasions. Comme présenté dans le chapitre 2 (pages 19 et 26), ces algorithmes nécessitent de fixer plusieurs paramètres, notamment la fonction d'appartenance, la fonction d'agrégation floue et le critère de sélectivité  $\alpha$ . Sauf indication contraire, les paramètres suivants seront appliqués :



- Fonction d'appartenance : binomiale centrée
- Fonction d'agrégation : min-max
- $\alpha = 0,8$

L'utilisation de la binomiale centrée est un des meilleurs choix pour un traitement de données multi-classes, puisque cette fonction possède un maximum paramétrable et inclut la notion de dispersion des données. Cela ressemble fortement à une gaussienne, mais la fonction gaussienne de base est trop sélective et présente souvent des résultats moins précis. L'agrégateur flou min-max est le plus pertinent puisqu'il offre des calculs qui ne propagent pas les différentes erreurs, contrairement à la solution probabiliste. Le choix de  $\alpha$  se base sur l'expérience. Sur de nombreux essais de classification supervisée avec des jeux de données différents, l'erreur de classification est minimale lorsque  $\alpha$  est compris entre 0,7 et 0,9 selon les cas. La valeur 0,8 est la valeur médiane de l'intervalle.

## 5.3 Étude par l'équation de la chaleur

### 5.3.1 Présentation du modèle

Une première approche pour retrouver la température à cœur est d'utiliser un modèle physique : l'équation de la chaleur. Il s'agit d'une équation différentielle partielle qui décrit l'évolution de la température dans l'espace et dans le temps (équation 5.1).

$$\frac{\partial T}{\partial t}(x, t) = c \nabla^2 T(x, t) \quad (5.1)$$

avec :

$t$  : temps

$x$  : point de l'espace  $\Omega$ ,  $\Omega \subset \mathbb{R}^3$

$c$  : coefficient de diffusion, propre au matériau considéré

Afin de permettre la résolution par un microcontrôleur, l'équation est discrétisée en temps et dans l'espace. Les capteurs ne permettent pas la connaissance de la répartition dans l'espace de la température, de même que la forme de l'aliment placé dans le four. De fait, l'emploi d'un modèle en trois dimensions est inadapté. On propose donc d'utiliser un modèle en une dimension dans un repère cartésien. Ceci fait que l'aliment sera considéré comme une plaque infinie dont les seuls effets viennent du dessus et du dessous de l'aliment de manière symétrique, les effets se propageant par couche (voir figure 5.2). Ce type de modèle est inspiré par [Ziouche 2015].

L'équation 5.1 devient alors :

$$T(t + \Delta t, n) = T(t, n) + r[T(t, n - 1) - 2T(t, n) + T(t, n + 1)] \quad (5.2)$$

avec :

$T(t, n)$  : température pour un instant  $t$  à la couche  $n$

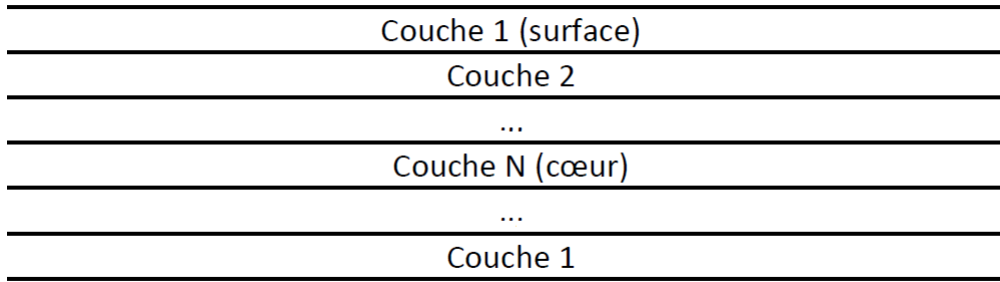


FIGURE 5.2 – Présentation graphique du modèle à base de couches pour la résolution de l'équation de la chaleur discrétisée

$\Delta t$  : pas de temps

$r$  : coefficient égale à  $c \frac{\Delta t}{\Delta x^2}$  où  $\Delta x$  correspond à l'épaisseur d'une couche

La température de la couche 1 est celle mesurée par la thermopile, qui sert donc d'entrée pour résoudre l'équation. La couche  $N$  qui est celle au centre du produit est la donnée de sortie. Elle se calcule en posant l'hypothèse de symétrie, la couche  $N + 1$  étant alors identique à la couche  $N - 1$ .

Le fait est que pour exécuter la résolution de cet algorithme, il faut la connaissance du paramètre  $r$ , qui dépend de trois autres paramètres. On peut considérer le coefficient de diffusion comme étant constant pour différents morceaux d'un même aliment, et le temps d'échantillonnage est un paramètre parfaitement maîtrisé. Néanmoins, la taille des couches varie de manière non contrôlée avec la taille des morceaux mis à cuire.

### 5.3.2 Recherche de $r$ à posteriori

Dans un premier temps, l'objectif a été d'identifier le paramètre  $r$  à partir d'expériences pour lesquelles on dispose de la mesure de la température de surface et de la température au cœur des aliments. Ceci permettra de définir une plage de valeurs pour ce paramètre et également de tester la viabilité du modèle en connaissant le paramètre de manière optimale.

La méthode employée pour rechercher ce paramètre fut l'optimisation d'un critère basé sur une erreur quadratique. Soit  $f(T_{surf}(t), r)$  une fonction capable de calculer un vecteur de température au centre de l'aliment en fonction du paramètre  $r$  et du vecteur de température de surface. On pose alors  $T_{cf}(t) = f(T_{surf}(t), r)$ . La recherche du paramètre  $r$  optimal passe alors par la minimisation du critère suivant :

$$\min_r \left( \sum_t (T_{cf}(t) - T_{mesure}(t))^2 \right) \quad (5.3)$$

avec  $T_{mesure}(t)$  qui est un vecteur temporel de la température à cœur mesurée.

La figure 5.3 montre un exemple de température calculée sur un rôti de bœuf en déterminant le paramètre  $r$  optimal pour un temps d'échantillonnage de 1s et pour un modèle comportant six couches. La valeur de  $r$  pour ce cas est de 0,0043 avec une erreur quadratique de 16534, ou une erreur moyenne de 1,37 °C, ce qui est un résultat acceptable dans le cas d'application présent.

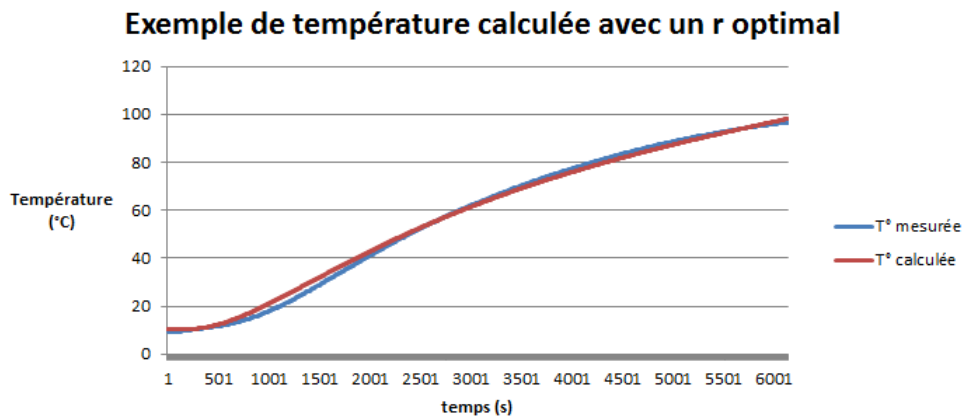


FIGURE 5.3 – Exemple de température recalculée à l'aide de l'équation de la chaleur et d'un  $r$  optimal

L'intégralité des optimisations ne sera pas montrée ici, mais les résultats sur l'ensemble des expériences sont du même ordre de précision (pour d'autres exemples, voir annexe C). Le modèle est donc viable pour le suivi de cuisson. L'ensemble des résultats obtenus sur les expériences de bœuf à l'aide du prototype v2 sont donnés dans le tableau 5.1.

Il est possible de chercher à établir des corrélations entre les différents paramètres des morceaux de bœuf et le paramètre  $r$ , calculé comme il suit :  $corr(X, Y) = \frac{cov(X, Y)}{\sigma_X \sigma_Y}$  où  $cov(X, Y)$  est la covariance entre les vecteurs  $X$  et  $Y$  et  $\sigma_X$  et  $\sigma_Y$  leurs écart-type. Les valeurs de coefficients de corrélation entre différentes données du morceau de bœuf et  $r$  sont les suivants :

- masse -  $r$  : -0.77
- longueur -  $r$  : -0.58
- hauteur -  $r$  : -0.88
- ratio masse/longueur : -0.88

Le modèle semble bien corrélé à la section du morceau de viande, puisque les plus grands coefficients de corrélation sont pour la hauteur et le ratio masse/longueur, ce dernier étant une image de la section du morceau (un ratio masse/longueur élevé rendant état d'une grande section). Ceci peut être visualisé sur la figure 5.4. Les coefficients négatifs expriment le fait que si la section augmente, la chaleur sera diffusée plus lentement et donc  $r$  va diminuer.

Tous les paramètres des expériences étant connus, la prochaine étape est de trouver  $r$  dans les premiers instants de la cuisson afin de pouvoir appliquer l'équation

Masse (g)	Longueur (mm)	Hauteur (mm)	Largeur (mm)	Ratio masse/longueur	r
525	90	70	90	5,83	0,0052
559	100	65	85	5,59	0,0038
625	100	70	100	6,25	0,0043
632	100	70	100	6,32	0,0046
492	90	70	90	5,47	0,0043
540	100	70	90	5,40	0,0044
897	100	95	110	8,97	0,0029
1178	200	70	90	5,89	0,0033
1818	200	85	120	9,09	0,0027
499	100	70	100	4,99	0,0047
990	140	85	110	7,07	0,003
506	100	60	80	5,06	0,006
505	150	50	65	3,37	0,0058
2410	250	80	110	9,64	0,0027
566	90	67	93	6,29	0,0047
539	98	64	93	5,50	0,0042
566	121	61	83	4,68	0,005
470	106	55	66	4,43	0,0059
319	102	47	65	3,13	0,0061
350	103	52	65	3,40	0,0059
461	142	64	53	3,25	0,0052
778	117	74	95	6,65	0,0031
1085	150	75	103	7,23	0,0029
1912	200	82	114	9,56	0,0023

TABLE 5.1 – Tableau récapitulatif des données physiques des expériences de rosbif

de la chaleur pour le suivi de l'état à cœur.

### 5.3.3 Recherche de $r$ à priori

#### 5.3.3.1 Recherche par algorithmes de classification

Avant de chercher à retrouver directement la valeur de  $r$  à partir des mesures durant les premiers instants de la cuisson, on propose d'essayer de résoudre un problème plus simple en réalisant une classification en trois classes correspondant à des concepts de taille que l'on qualifiera de « petit », « moyen » et « gros ». La principale difficulté est que l'on base l'analyse sur les résultats du prototype v2 qui dispose d'un nombre de capteurs limité. Par ailleurs, le capteur olfactif n'est pas utilisable dans ce cas car en début de cuisson ses mesures ne sont pas stables et fournissent des résultats très différents selon les expériences. Le seul capteur réellement utilisable est la thermopile.

La première étape va être de réaliser une sélection sur différentes variations de

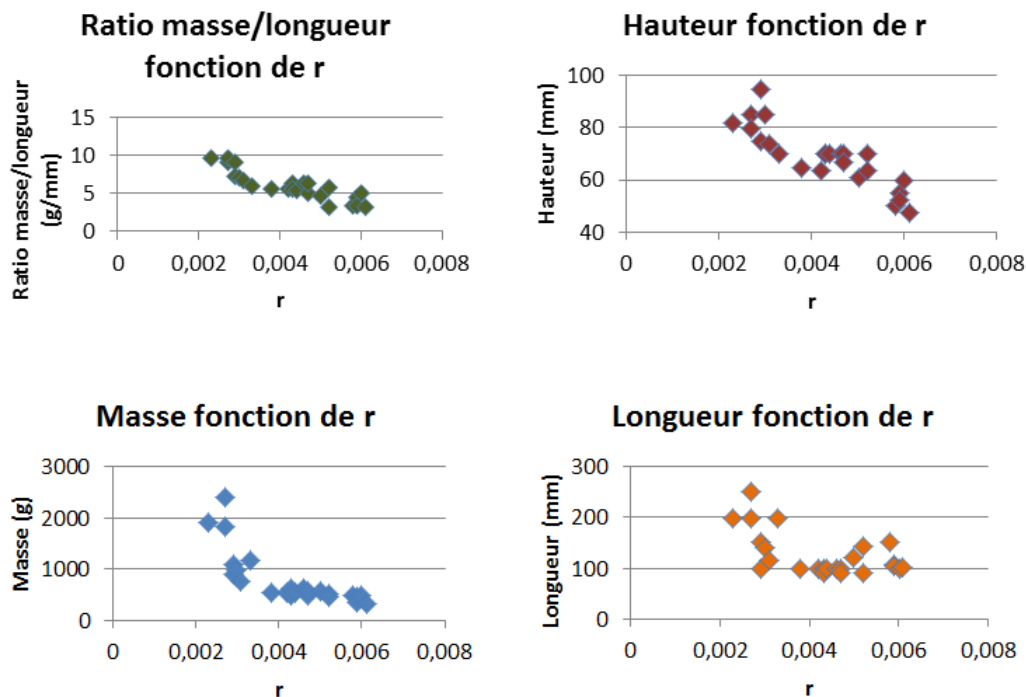


FIGURE 5.4 – Visualisation du paramètre  $r$  en fonction de plusieurs données physiques de rosbif

la température de surface. La liste des attributs créés est la suivante :

- $T_{surf}(t)$  : valeur brute du capteur de température de surface
- $T_{surf\_relative}(t) = T_{surf}(t) - T_{surf}(0)$  : température de surface relative à la température mesurée à l'enfournement
- $Int\_T_{surf}(t)$  : intégrale temporelle de la température de surface
- $Int\_T_{surf\_relative}(t)$  : intégrale temporelle de la température de surface relative

La sélection est effectuée sur vingt-deux expériences de bœuf réalisées sur le prototype v2 (six « petits », huit « moyens » et huit « gros »). Cette dernière est réalisée de deux manières : en utilisant l'algorithme MEMBAS couplé avec les fonctions d'appartenance floue de l'algorithme de classification LAMDA et également par l'observation des poids issus du calcul d'un modèle de type SVM linéaire paramétré en « un contre tous ». Le vecteur de données est constitué des quatre attributs détaillés ci-dessus, chacun d'eux étant pris sur les 800 premières secondes de la cuisson, soit un total de 3200 attributs. Les résultats de l'algorithme MEMBAS sont donnés sur la figure 5.5 et les poids des séparateurs des SVM en figure 5.6.

De ces résultats plusieurs choses apparaissent. La première est que les deux méthodes de sélection testées ici produisent des informations assez proches, sachant que dans le cas de l'analyse à l'aide des SVM, c'est la valeur absolue des poids

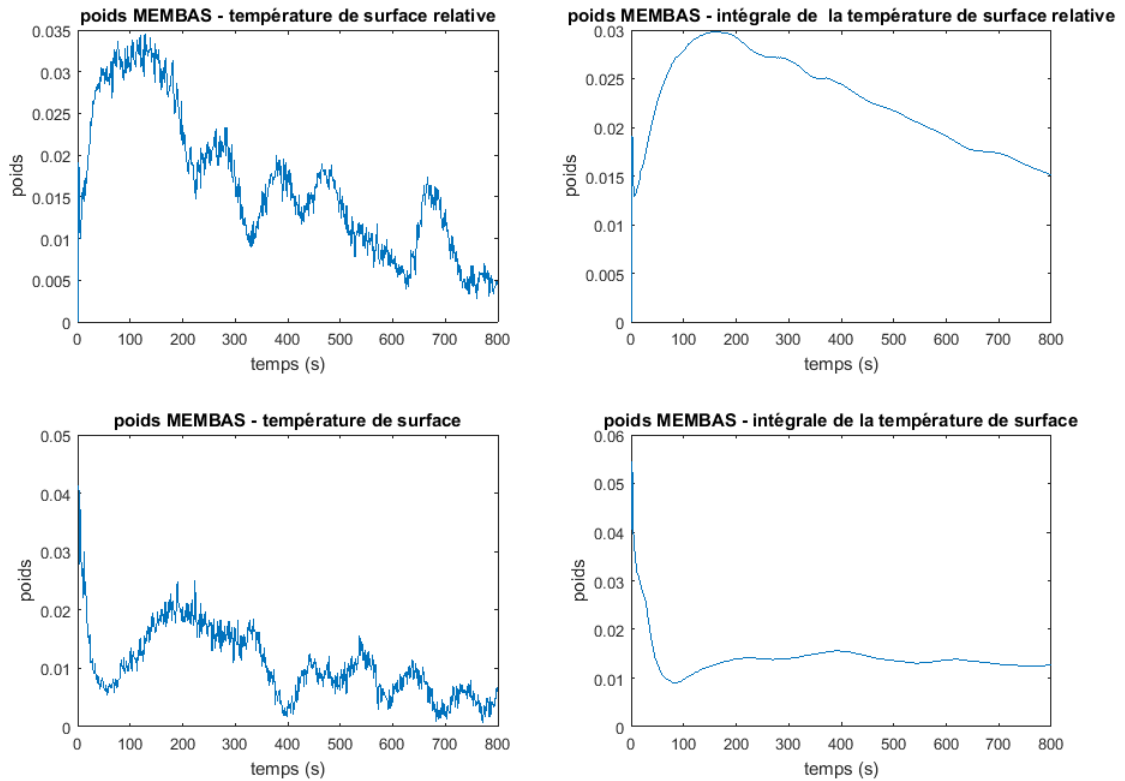


FIGURE 5.5 – Valeurs des poids de l’algorithme MEMBAS pour la classification de la taille d’un rosbif

qu’il faut prendre en compte. Ensuite, les données issues de la température relative semblent plus intéressantes pour la classification que celles issues de la température brute. Un pic d’intérêt est observable en tout début de cuisson sur les données de température brute, néanmoins il est certainement très risqué de se fier à cette information étant donné que cela dépend de la température initiale de la viande en dehors du four, ce qui est une donnée non corrélée à l’épaisseur. Les matrices de confusion pour le vecteur complet (3200 attributs) calculées en « leave one out » pour LAMDA et un SVM linéaire sont données en figure 5.7. La manière de lire ces matrices de confusion est montrée en annexe B.

Sur ce problème, les SVM obtiennent un résultat légèrement supérieur à l’algorithme LAMDA, mais reste tout de même très peu fiable. Pour essayer d’améliorer le taux de réussite de la classification, on va chercher à réduire le nombre d’attributs en utilisant les scores de la sélection. Seulement trois variables sont conservées : la température relative à la 128<sup>ème</sup> seconde, l’intégrale de la température relative à la 158<sup>ème</sup> seconde et la température brute à la 196<sup>ème</sup> seconde. Les résultats de classification sont donnés en figure 5.8.

Dans les deux cas, la classification est plus performante, mais reste imparfaite en l’état, ce qui confirme l’importance du choix des informations prises pour effectuer

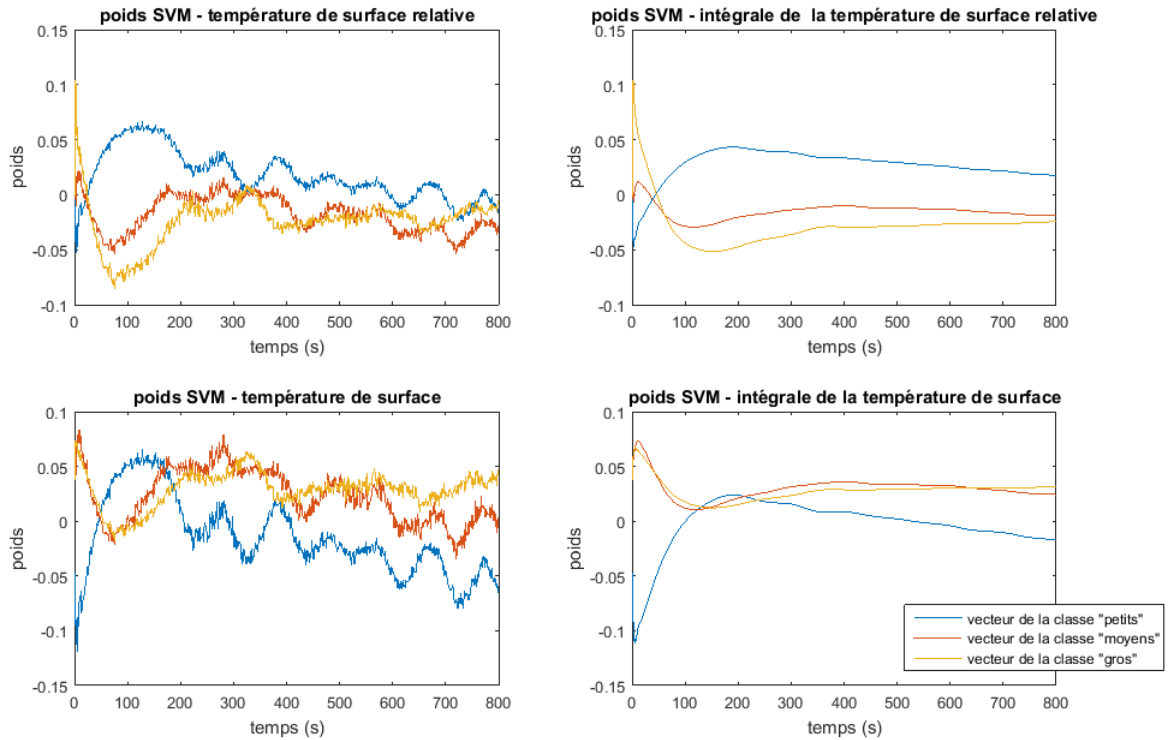


FIGURE 5.6 – Valeurs des poids des séparateurs de l’algorithme SVM pour la classification de la taille d’un rosbif

la classification, même dans le cas d’un faible nombre de descripteurs. Les deux problèmes majeurs sont :

- La nécessité d’employer la température de surface brute sans quoi la précision de la classification chute à des taux proches de cinquante pourcents, ce qui révèle un certain manque de robustesse des modèles.
- Les algorithmes testés proposent dans tous les cas une erreur de deux classes d’écart (classé en « petit » à la place de « gros »), ce qui révèle également un manque de robustesse du modèle.

### 5.3.3.2 Calcul du paramètre $r$

Une première approche pour calculer le paramètre  $r$  est d’utiliser les résultats de la classification par l’algorithme LAMDA. En effet, ce dernier utilise comme base la logique floue, de fait il est possible de réaliser une défuzzification à partir des DAG de chaque classe et ainsi d’exprimer la valeur  $r$  comme la somme des  $r$  moyens de chacune des classes pondérée par les DAG respectifs. Le paramètre se calcul alors selon l’équation 5.4.

$$r = \frac{\sum_{c=1}^3 Rmoy_c DAG_c}{\sum_c DAG_c} \quad (5.4)$$

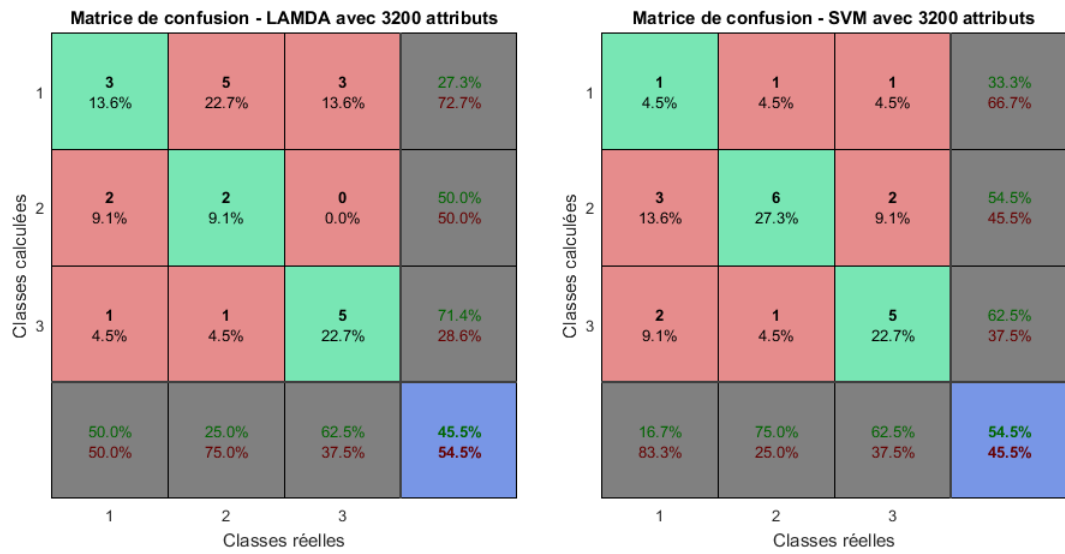


FIGURE 5.7 – Matrices de confusion pour une classification de la taille des aliments avec l’algorithme LAMDA et par SVM linéaire (3200 attributs)

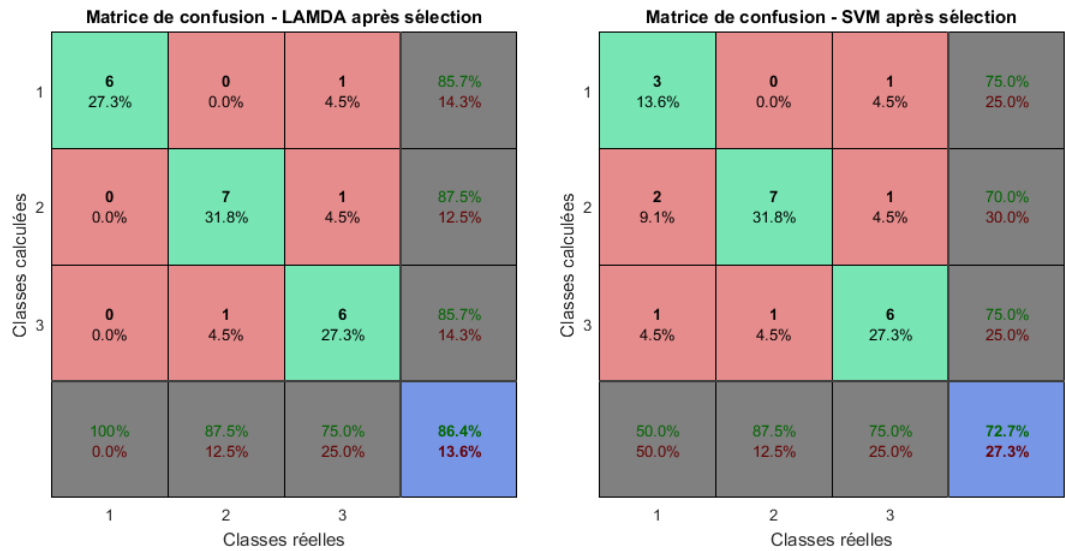


FIGURE 5.8 – Matrices de confusion pour une classification de la taille des aliments avec l’algorithme LAMDA et par SVM linéaire (3 attributs)

en prenant  $R_{moy} = \{0,00275 \ 0,0042 \ 0,0056\}$

Les résultats du calcul de  $r$  par défuzzification sont donnés en figure 5.9. Il est notable que les résultats s’approchent tous de la valeur des  $r$  « moyens », ceci est provoqué par les  $DAG$  qui sont tous très proches en terme de valeur. Il n’y a pas assez de distinction pour permettre d’obtenir un résultat valide par ce calcul.



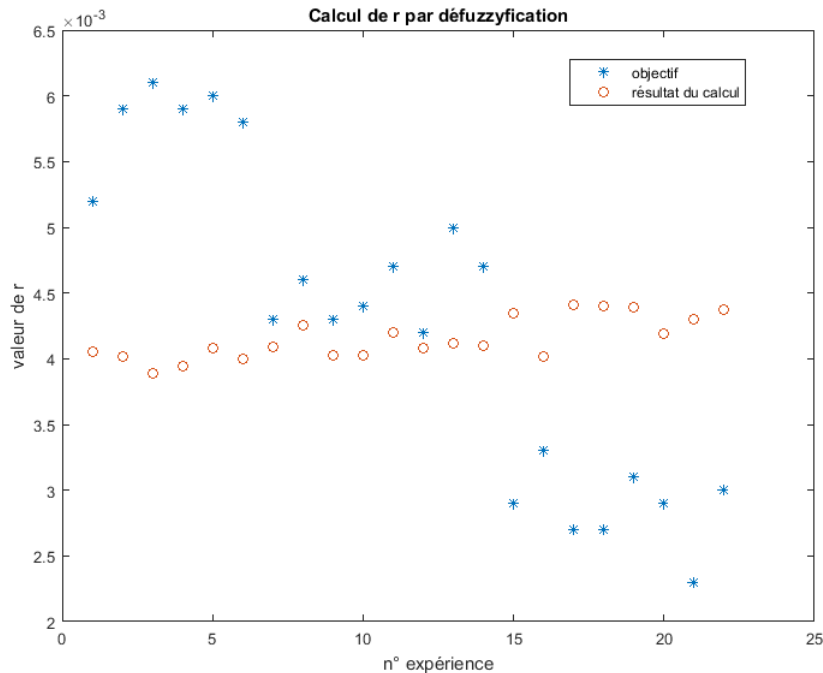


FIGURE 5.9 – Comparaison entre les paramètres  $r$  optimaux et ceux calculés par défuzzification

Une autre approche consiste à effectuer une régression du paramètre à l'aide de réseaux de neurones. Deux réseaux sont testés : le premier comporte en entrées les trois attributs retenus lors de la sélection pour la classification, le second comporte soixantes entrées, la température de surface, la température de surface relative et la température de surface relative intégrée, le tout toutes les dix secondes sur les deux cent premières secondes de la cuisson. Le premier réseau comporte vingt neurones dans sa couche cachée, le second cent neurones. Les résultats pour ces deux modèles sont présentés figures 5.10 et 5.11.

Même si l'usage de la régression par réseaux de neurones offre dans les deux cas de meilleurs résultats que la logique floue, il reste encore de nombreux résultats aberrants qui rendent ce type de modèle non-fiable. De plus la puissance de calcul et la mémoire nécessaires pour exécuter le second réseau commencent à devenir non négligeables pour un microcontrôleur à bas coût.

### 5.3.4 Conclusion sur la méthode par calcul de l'équation de la chaleur

Il a été montré dans cette section qu'il était possible de déterminer à l'aide d'un modèle simplifié de l'équation de la chaleur la température au cœur d'un morceau de bœuf. Néanmoins pour obtenir avec précision le paramètre nécessaire à la résolution numérique du modèle, il est nécessaire de disposer de l'expérience dans son

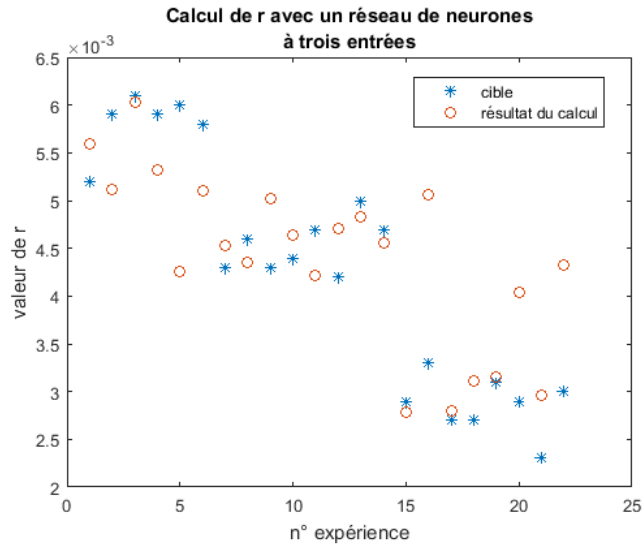


FIGURE 5.10 – Comparaison entre les paramètres  $r$  optimaux et ceux calculés par un réseau de neurones à trois entrées

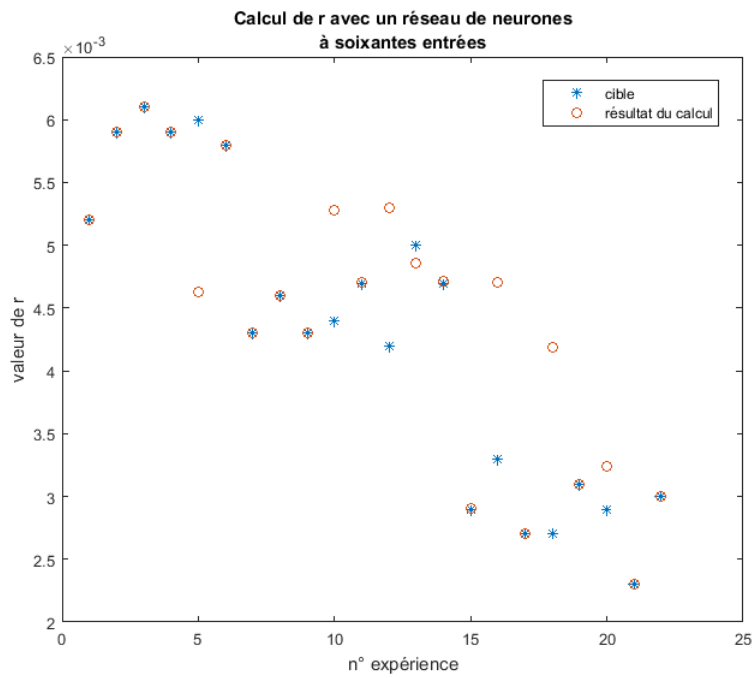


FIGURE 5.11 – Comparaison entre les paramètres  $r$  optimaux et ceux calculés par un réseau de neurones à soixante entrées

intégralité et d'une mesure de la température recherchée. En effet, en utilisant les mesures présentes dans les prototypes de four du projet, il est extrêmement compliqué d'obtenir des résultats fiables que ce soit par classification de données ou

par régression. D'autres informations comme la masse ou un facteur de taille mesurés par des capteurs ou saisis par l'utilisateur pourraient possiblement permettre d'améliorer les résultats pour la recherche en ligne du paramètre de l'équation et donc d'utiliser de manière fiable l'équation de la chaleur pour permettre un suivi de cuisson en ligne.

## 5.4 Techniques de classification pour le suivi de cuisson

### 5.4.1 Présentation

L'idée d'utiliser la classification supervisée pour déterminer l'état à cœur des viandes et poissons est un des éléments qui a initialement motivé cette thèse. L'objectif est d'utiliser la mesure de la température à cœur effectuée lors des essais pour construire des classes qui seront l'image de l'état à cœur. A terme, la solution retenue doit pouvoir être embarquée dans le four, ce qui implique des contraintes de mémoire et de complexité de calcul sur l'algorithme, la puissance de calcul étant celle d'un microcontrôleur.

Cette partie aborde l'étude des données sur les deux prototypes de four. Le prototype v1 étant lui-même construit en deux temps, seront présentés les résultats de sélection et de la classification sur le modèle de four incomplet avant de donner quelques informations de sélection d'attributs sur le prototype disposant de tous les capteurs. Le même type de travail fut également réalisé sur le second prototype.

### 5.4.2 Analyses effectuées sur le prototype v1 première génération

Comme annoncé précédemment, la première version du prototype v1 a été finalisée en juillet 2015, et comportait plusieurs capteurs opérationnels. Néanmoins, le capteur par fluorescence et le capteur de bandes spectrales étant incapables de travailler simultanément, seules la température de surface mesurée par une thermopile et les mesures du capteur de composés organiques volatils étaient exploitables. Un exemple de mesures réalisées sur cette version comportant la phase de préchauffe du four et la cuisson d'un rosbif est donné en figure 5.12.

Les données brutes du capteur de composés organiques volatils n'offrent que peu d'informations en l'état. Les fabricants du capteur conseillent de travailler en variations relatives par rapport à un instant de l'expérience, le point typique étant l'enfournement (début de la phase de cuisson). Le calcul est donné par l'équation 5.5.

$$\hat{o}(t) = \frac{o(t) - o(t_0)}{o(t_0)} \quad (5.5)$$

avec :

$t$  : temps

$o(t)$  : donnée brute à l'instant  $t$

$\hat{o}(t)$  : donnée relative à l'instant  $t$

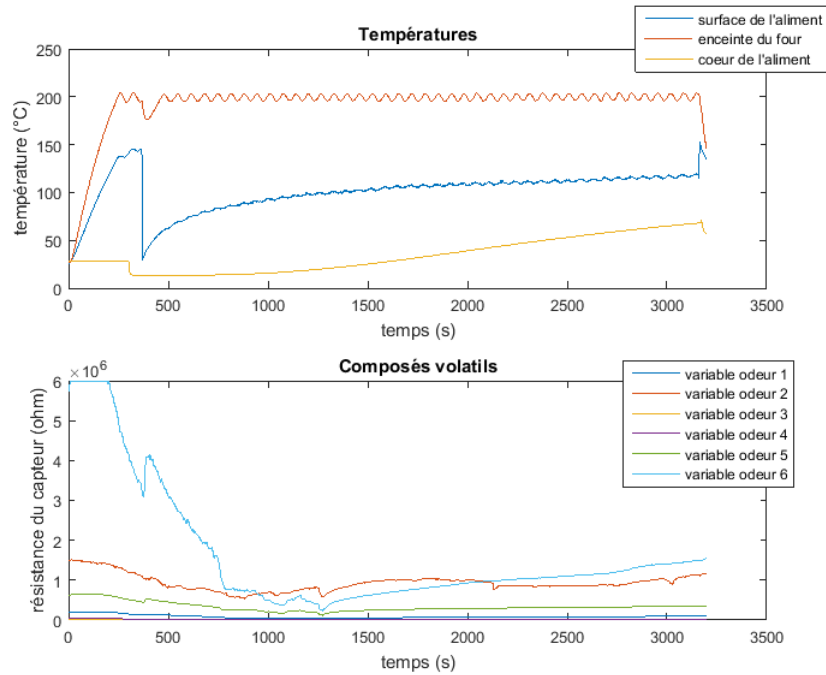


FIGURE 5.12 – Exemple de mesures réalisées dans le premier prototype v1

$t_0$  : temps de référence

En suivant cette transformation, les données de la figure 5.12 deviennent telles que montrées sur la figure 5.13.

Avec cette version des capteurs, il est à remarquer que les valeurs décroissent en début de cuisson avant de croître de nouveau. Les courbes ne sont donc pas monotones, ce qui a pour conséquence de rendre plus ardue la classification des états de cuisson. Pour remédier à cela, l'idée est de prendre comme valeur de référence pour la transformation des données non pas le temps d'enfournement mais le minimum entre la valeur de référence actuelle et la mesure à transformer à chaque pas de temps. Ceci est explicité dans l'algorithme suivant :

```

 $t_0 \leftarrow 0$ 
pour chaque  $t$  faire
    si  $valeur[t_0] > valeur[t]$  alors
        |  $t_0 \leftarrow t$ 
    fin
     $valeurRelative[t] \leftarrow \frac{valeur[t] - valeur[t_0]}{valeur(t_0)}$ 
fin

```

**Algorithme 1** : Transformation des données odeur

Les mesures transformées suivant cet algorithme sont visualisables sur la figure

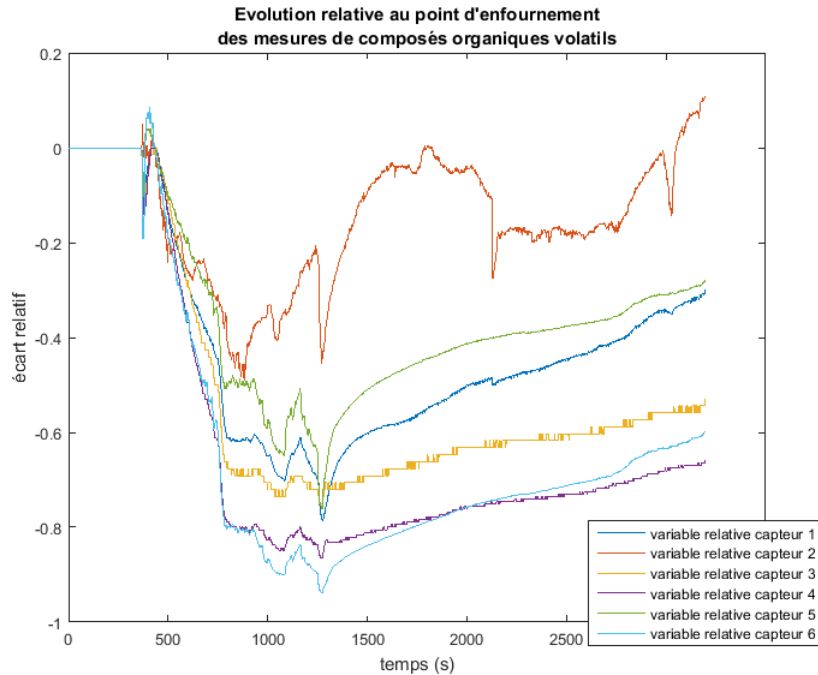


FIGURE 5.13 – Exemple de mesures du capteur olfactif normalisées

5.14.

Pour valider la pertinence de ces modifications sur les données, une sélection à l'aide de l'algorithme MEMBAS est réalisée sur l'ensemble des cuissons de rosbif entre les trois types de données (mesures brutes, valeurs relatives avec référence fixe et valeurs relatives avec référence dynamique). Les résultats sont donnés sur la figure 5.15. La sélection montre nettement que seules les valeurs construites avec une référence dynamique offrent une information viable pour réaliser une classification de l'état à cœur. Parmi elles, deux se dégagent du lot, la mesure 1 et la mesure 2. Seules les valeurs avec la référence dynamique seront conservées pour la suite de cette étude.

Une fois les variables olfactives mises sous une forme exploitable par un algorithme de classification, une nouvelle sélection est réalisée sur un plus grand ensemble de variables issues des capteurs du four. Sur la version considérée, la seule autre donnée potentiellement utile est la température de surface, la température d'enceinte du four étant quasiment constante, elle n'apporte pas d'information. Une variable construite à partir de la température de surface est ajoutée aux attributs à classer, il s'agit de la valeur cuisatrice [Zuber 2007]. Elle se calcule comme indiqué dans l'équation 5.6. Cette donnée permet la description de l'évolution temporelle des qualités nutritionnelles et organoleptiques des produits alimentaires (couleur, flaveur, texture ...).

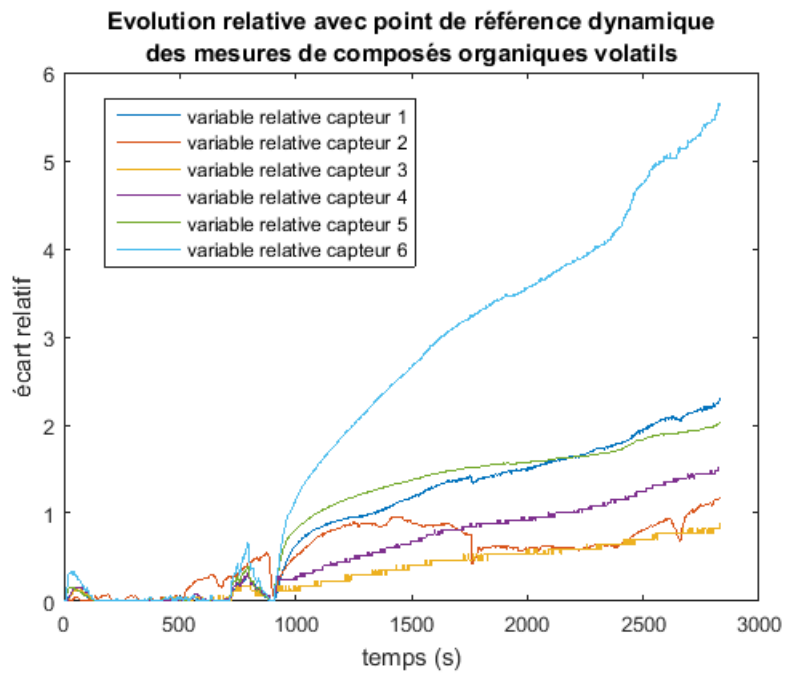


FIGURE 5.14 – Exemple de mesures du capteur olfactif normalisées avec une référence dynamique

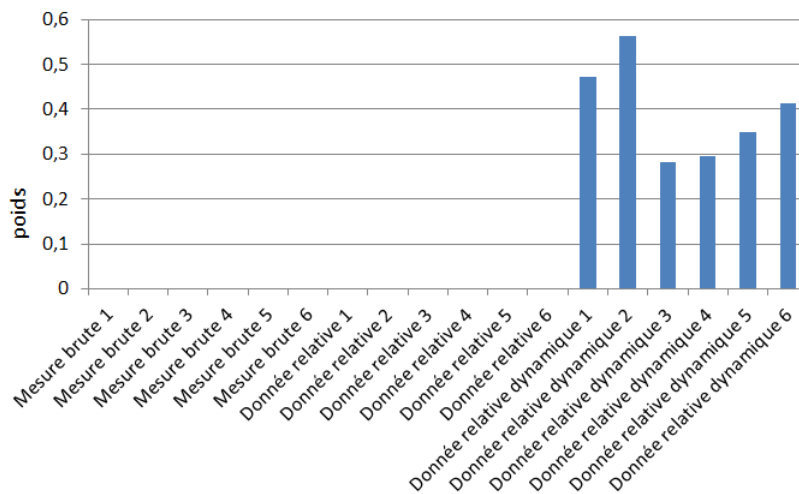


FIGURE 5.15 – Poids MEMBAS pour les trois types de données olfactives sur le prototype v1

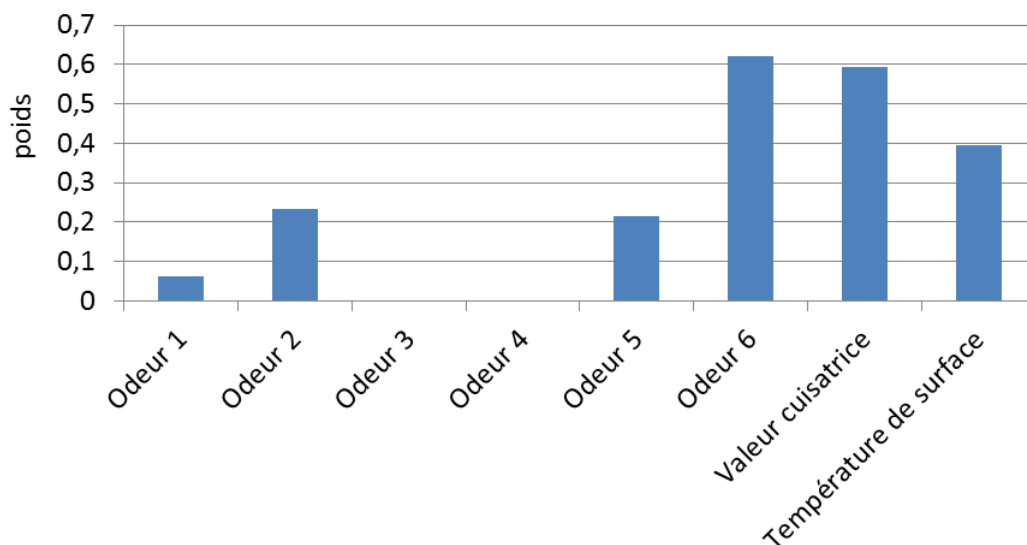


FIGURE 5.16 – Poids MEMBAS pour l’ensemble des données du prototype v1

$$Vc(t) = \int_0^t 10^{\frac{T(t)-100}{Z_c}} dt \quad (5.6)$$

avec :  $T(t)$  la température en surface au temps  $t$  et  $Z_c = 25^\circ C$

On construit un premier vecteur de données avec les six mesures olfactives relatives avec une référence dynamique, la température de surface et la valeur cuisatrice de surface. Les résultats de la sélection avec MEMBAS sont portés sur la figure 5.16.

Les résultats de classification en utilisant l’algorithme LAMDA en mode « leave one out » (dans le sens où l’expérience testée est mise à l’écart durant la phase d’apprentissage) sont donnés figures 5.17 et 5.18. Un essai avec des SVM linéaires est également présenté en figure 5.19. L’information issue des SVM est légèrement plus précise de manière générale mais n’offre cependant pas une précision suffisante pour un suivi de cuisson idéal.

La réduction du nombre d’attributs pour réaliser la classification de l’état à cœur n’offre que peu d’amélioration, les résultats ne seront donc pas détaillés ici. Au bilan, ce que l’on peut tirer de cette première version du prototype v1 est l’existence de variables d’intérêt pour réaliser des classifications de l’état à cœur parmi les capteurs présents dans le four. Néanmoins, ces variables ne permettent pas dans tous les cas un suivi de cuisson réaliste même avec l’utilisation de différents algorithmes de classification, les erreurs pouvant aller jusqu’à un écart de cinq classes dans les pires cas. Plusieurs causes de ces erreurs sont notables, tout d’abord la base d’apprentissage est très limitée, puisque réduite à neuf cuissons, et les capteurs olfactifs n’en étaient qu’à leur première version et ont évolué par la suite. On retiendra donc des études de cette version qu’il est certainement possible d’arriver

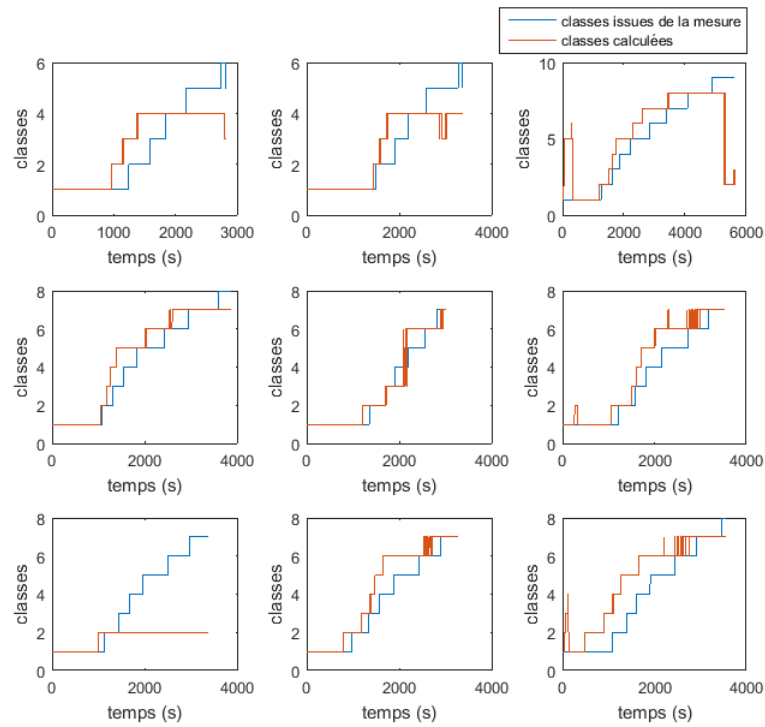


FIGURE 5.17 – Résultats de la classification par LAMDA des données de la cuisson des rosbif

**Matrice de confusion - LAMDA avec huit attributs**

1	8715 26.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100%
2	1525 4.7%	1561 4.8%	242 0.7%	296 0.9%	539 1.7%	473 1.5%	396 1.2%	0 0.0%	313 1.0%	29.2%
3	344 1.1%	773 2.4%	273 0.8%	215 0.7%	138 0.4%	13 0.0%	0 0.0%	0 0.0%	22 0.1%	15.4%
4	21 0.1%	577 1.8%	926 2.9%	737 2.3%	1141 3.5%	146 0.4%	0 0.0%	0 0.0%	0 0.0%	20.8%
5	238 0.7%	136 0.4%	685 2.1%	949 2.9%	246 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10.9%
6	14 0.0%	0 0.0%	0 0.0%	683 2.1%	2752 8.5%	782 2.4%	80 0.2%	0 0.0%	0 0.0%	18.1%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	259 0.8%	1904 5.9%	2135 6.6%	364 1.1%	0 0.0%	45.8%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	643 2.0%	808 2.5%	405 1.2%	43.5%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	NaN%
	80.3%	51.2%	12.8%	25.6%	4.8%	23.6%	65.6%	68.9%	0.0%	47.0%
	19.7%	48.8%	87.2%	74.4%	95.2%	76.4%	34.4%	31.1%	100%	53.0%
	1	2	3	4	5	6	7	8	9	

Classes réelles

FIGURE 5.18 – Matrice de confusion de la classification avec LAMDA des données rosbif du prototype v1



**Matrice de confusion - SVM**

Classes calculées	1	10403 32.0%	403 1.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	96.3% 3.7%
	2	452 1.4%	2116 6.5%	565 1.7%	60 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	66.3% 33.7%
	3	0 0.0%	525 1.6%	890 2.7%	492 1.5%	36 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	45.8% 54.2%
	4	0 0.0%	3 0.0%	638 2.0%	1544 4.8%	873 2.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50.5% 49.5%
	5	0 0.0%	0 0.0%	33 0.1%	708 2.2%	3048 9.4%	1543 4.8%	206 0.6%	0 0.0%	0 0.0%	55.0% 45.0%
	6	2 0.0%	0 0.0%	0 0.0%	75 0.2%	716 2.2%	1073 3.3%	802 2.5%	0 0.0%	0 0.0%	40.2% 59.8%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	399 1.2%	690 2.1%	2192 6.8%	1090 3.4%	740 2.3%	42.9% 57.1%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	54 0.2%	82 0.3%	0 0.0%	60.3% 39.7%
	9	0 0.0%	0 0.0%	0 0.0%	1 0.0%	3 0.0%	12 0.0%	0 0.0%	0 0.0%	0 0.0%	0.0% 100%
			95.8% 4.2%	69.4% 30.6%	41.9% 58.1%	53.6% 46.4%	60.1% 39.9%	32.3% 67.7%	67.4% 32.6%	7.0% 93.0%	0.0% 100%
		1	2	3	4	5	6	7	8	9	
		Classes réelles									

FIGURE 5.19 – Matrice de confusion de la classification avec des SVM des données rosbif du prototype v1

à un résultat acceptable pour une application industrielle, mais que ça ne sera pas forcément trivial.

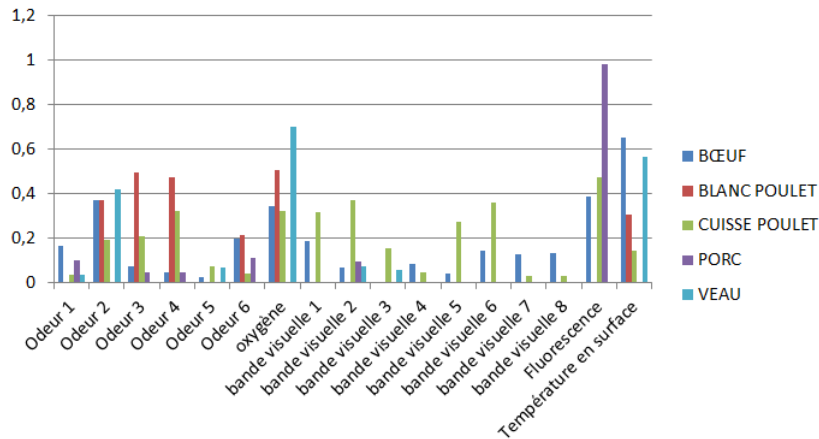


FIGURE 5.20 – Poids de l’algorithme MEMBAS pour la sélection de capteur sur le prototype v1 avec les viandes

### 5.4.3 Prototype v1 seconde génération

Suite à divers travaux de recherche et de développement menés par les partenaires du projet, une seconde version du prototype v1 a vu le jour durant le mois de mai 2016. Cette deuxième version était capable de faire fonctionner simultanément l’ensemble des capteurs initialement prévus de manière simultanée. Le peu de temps disponible à partir de l’obtention de ce prototype a été mis à profit pour effectuer une vérification à posteriori de la pertinence des choix effectués concernant les capteurs intégrés dans le prototype v2. On rappelle que sur des choix de coût, de facilité d’industrialisation et de disponibilité, le capteur choisi pour intégrer le prototype v2 fut le capteur de composés organiques volatils (odeurs) développé par Alpha MOS. Les résultats présentés ici seront uniquement des indicateurs et en aucun cas de véritables informations statistiques puisqu’uniquement deux cuissons ont été réalisées sur chaque produit. Comme mesures sur le prototype v1 seconde génération, nous disposons des données suivantes :

- six variables de mesure de composés organiques volatils
- une mesure de fluorescence
- une mesure du taux d’oxygène
- huit bandes spectrales dans le visible et le proche infrarouge
- la température à la surface de l’aliment

On réalise de nouveau une sélection d’attributs en utilisant l’algorithme MEMBAS pour chaque type d’aliment en intégrant l’ensemble des attributs disponibles. Ne disposant pas des limites de classes en fonction de la température à cœur, les classes ont été générées avec un seuil linéaire tous les 10°C à cœur (la classe 1 correspondant à une température à cœur entre 0 et 10°C). Les résultats sont sur les figures 5.20 et 5.21.

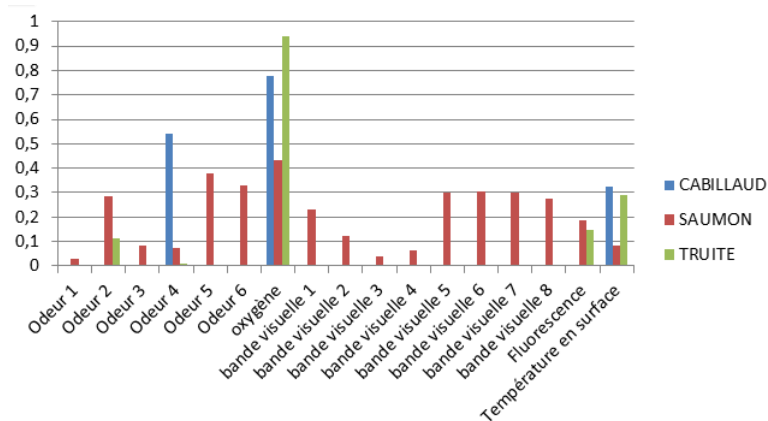


FIGURE 5.21 – Poids de l’algorithme MEMBAS pour la sélection de capteur sur le prototype v1 avec les poissons

On constate à partir de ces résultats que les choix effectués précédemment sur la première version du prototype v1 sont plutôt cohérents. Néanmoins, il aurait été intéressant d’avoir deux capteurs en plus à exploiter de manière expérimentale. La température de surface et le capteur olfactif offrent des scores qui indiquent une certaine pertinence, mais il aurait été utile d’ajouter le capteur de fluorescence et le capteur d’oxygène, même si ce dernier est très sensible aux ouvertures de porte du four durant la cuisson. On peut également constater une certaine difficulté à trouver des capteurs intéressants pour suivre la cuisson des poissons autres que le saumon.

#### 5.4.4 Analyses sur le prototype v2

Le prototype v2 est semblable en termes d’équipement à la première génération du prototype v1, les mesures utiles étant celles du capteur de composés organiques volatils et celles de la thermopile. Les capteurs de fluorescence et de mesures spectrales n’ont pas été retenus pour cette dernière version du prototype. Néanmoins sur cette version, le capteur d’odeurs extrait douze variables et non plus six comme c’était le cas précédemment. Un autre changement vient dans le type de réponse générée par les capteurs olfactifs. Il a été montré précédemment que la transformation des données à l’aide d’une étude des variations relatives avec référence dynamique permettait d’obtenir de meilleurs résultats, ce qui n’est pas le cas sur le capteur du prototype v2. En effet les signaux issus de ces capteurs sont décroissants en moyenne durant la cuisson, ce qui rend l’exploitation précédente non pertinente. Un exemple de données d’une cuisson de bœuf est présenté en figure 5.22. Les données analysées dans le cadre de cette section seront du rosbif et des pavés de saumon.

Le calcul des variations relatives est donc effectué en suivant l’équation 5.5. Comme précédemment, la première étape avant de procéder à une classification est une étape de sélection. Les variables utilisées sont :

- les douze mesures des composés organiques volatils relatives au point d’en-

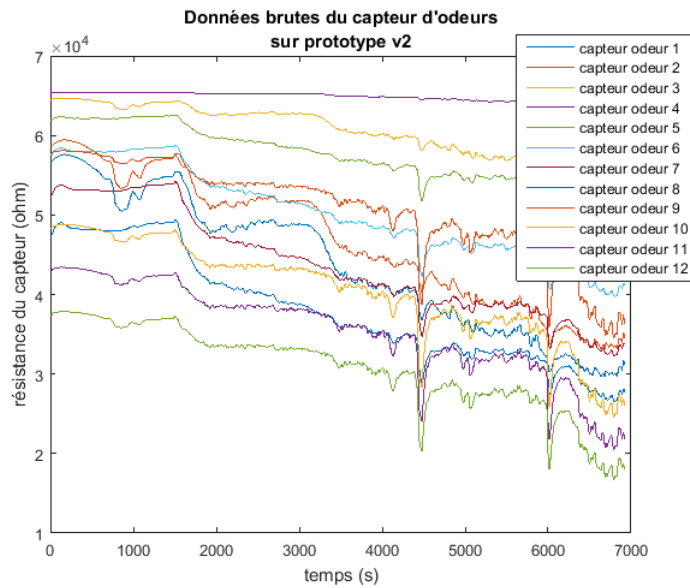


FIGURE 5.22 – Exemple de données olfactives recueillies sur le prototype v2

fournement,

- ces mêmes variables intégrées au cours du temps
- la température de surface
- la valeur cuisatrice

Les résultats de la sélection à l'aide de l'algorithme MEMBAS pour le rosbif sont donnés sur la figure 5.23 et ceux du saumon sur la figure 5.24. On constate, dans les deux cas, une grande importance de la température de surface et dans une moindre mesure de la valeur cuisatrice. Néanmoins, il est clair que ces informations de surface ne suffisent pas à classer correctement les différents cuissons, il va donc être nécessaire d'ajouter le capteur olfactif.

Le capteur olfactif est composé de trois micro-capteurs, et chacun d'entre eux fournit quatre mesures. Généralement, les mesures d'un même micro-capteur sont très fortement corrélées. On constate ici que dans le cas du rosbif c'est le second capteur (variables 5 à 8) qui offre la meilleure information et dans le cas du saumon c'est plutôt le capteur 3 (variables 9 à 12). Dans les deux types de cuisson, c'est la variable intégrée au cours du temps qui semble la plus pertinente. On a procédé alors à des tests de classification pour valider la pertinence des méthodes et des modèles générés à partir des données. Trois algorithmes de classification sont utilisés pour ce test : des SVM linéaires, des réseaux de neurones artificiels comportant quinze neurones dans leur couche cachée et la méthode LAMDA, cette dernière étant appliquée soit avec l'agrégation floue min-max comme décrit au début de ce chapitre, soit en utilisant l'agrégation pondérée comme décrit dans la section 4.2. Toujours concernant LAMDA, les intervalles flous triangulaires (voir section 4.1.2) sont utilisés ici pour modéliser l'incertitude sur certains attributs, l'écart-type uti-

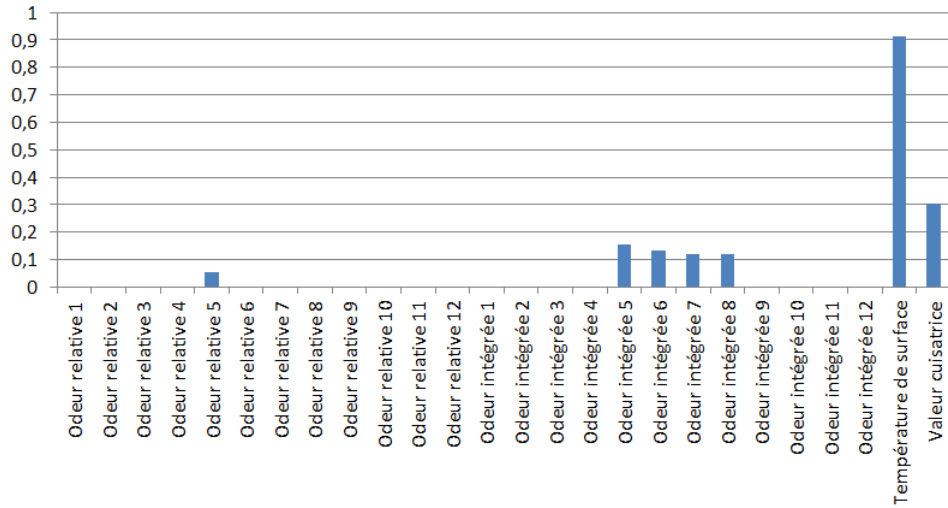


FIGURE 5.23 – Poids MEMBAS des différentes mesures dans le prototype v2 pour du rosbif

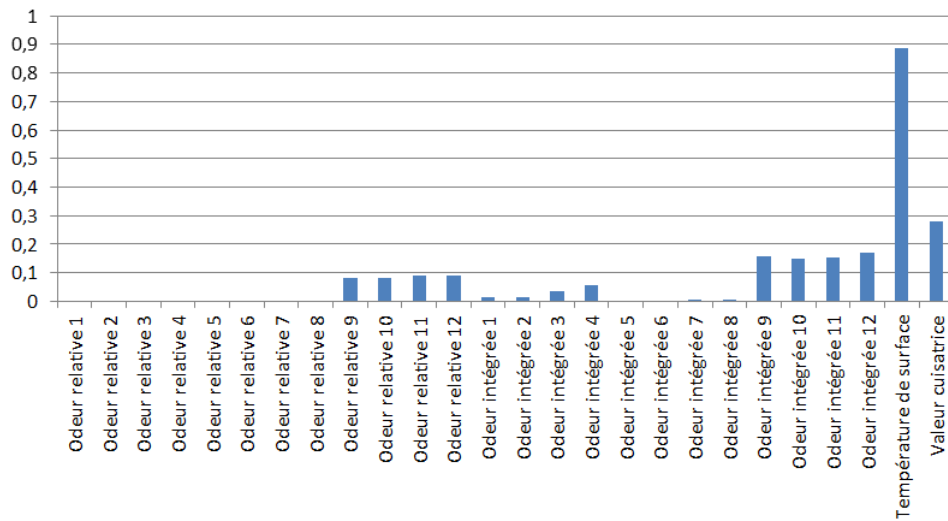


FIGURE 5.24 – Poids MEMBAS des différentes mesures dans le prototype v2 pour du saumon

lisé pour la construction des triangles étant un écart-type moyen des points des différentes expériences à chaque instant  $t$ . Pour déterminer à priori si ces intervalles ont un impact positif sur la classification, on utilise l'algorithme MEMBAS sur les données sous forme d'intervalles triangulaires et les données scalaires. Il en sort que dans les cas du saumon et du bœuf, les scores de MEMBAS sont plus élevés pour les données du capteur d'odeurs sous forme d'intervalles et pour la température de surface et la valeur cuisatrice sous forme scalaire. On conservera ces choix pour la classification avec l'algorithme LAMDA. Le saumon sera classé avec :

- la variable d'odeur intégrée n°12
- la valeur cuisatrice
- la température de surface

et le rosbif le sera avec :

- la variable d'odeur intégrée n°5
- la valeur cuisatrice
- la température de surface

Les tests sont effectués de telle sorte que chaque expérience testée est sortie de la base d'apprentissage, ce qui constitue une validation par mise à l'écart à l'échelle d'une expérience. Des résumés informatifs sur les résultats de ces classifications sont donnés figure 5.25 pour le rosbif et figure 5.26 pour le saumon. Les données utilisées pour le rosbif sont les mêmes que celles utilisées dans le cas de l'équation de la chaleur, et celles pour le saumon sont issues de huit cuissons avec des quantités différentes.

Dans les deux cas, les taux de classification globaux sont assez faibles (entre 32 % et 58 %). Les algorithmes ont du mal à effectuer les transitions entre les classes au bon moment, ce qui explique des taux de bonnes classifications par classes bas dans l'ensemble, à quelques exceptions près. Si la plupart des algorithmes classent correctement les données dans la classe 1, qui correspond à l'état cru, ils deviennent par la suite moins capables d'effectuer les transitions lors des changements d'état de cuisson. Il est également notable que la méthode LAMDA est dans l'ensemble plus performante dans sa version d'agrégation floue classique que lors de l'utilisation de la moyenne pondérée. Ceci est principalement causé par une valeur des poids qui est bien plus grande que les deux autres, ce qui revient à classer avec une seule variable. Sur l'ensemble des résultats le principal problème vient des différences de masse. Comme dans le cas de l'équation de la chaleur, nous ne disposons pas réellement de variable qui soit fortement corrélée à la masse des aliments. Au final les plus gros sont dans l'ensemble bien classés, alors que les petits et les moyens ont une ou deux classes de retard par rapport à la référence, ce qui représente potentiellement une différence entre un aliment saignant et à point.

Tout comme dans le cas de l'équation de la chaleur, les algorithmes de classification ont un potentiel certain pour recomposer l'état au cœur des aliments, mais l'incapacité actuelle à déterminer ou prendre en compte la taille des aliments ne permet pas un résultat fiable. La section suivante présente une dernière méthode pour estimer la température à cœur, les réseaux de neurones artificiels.

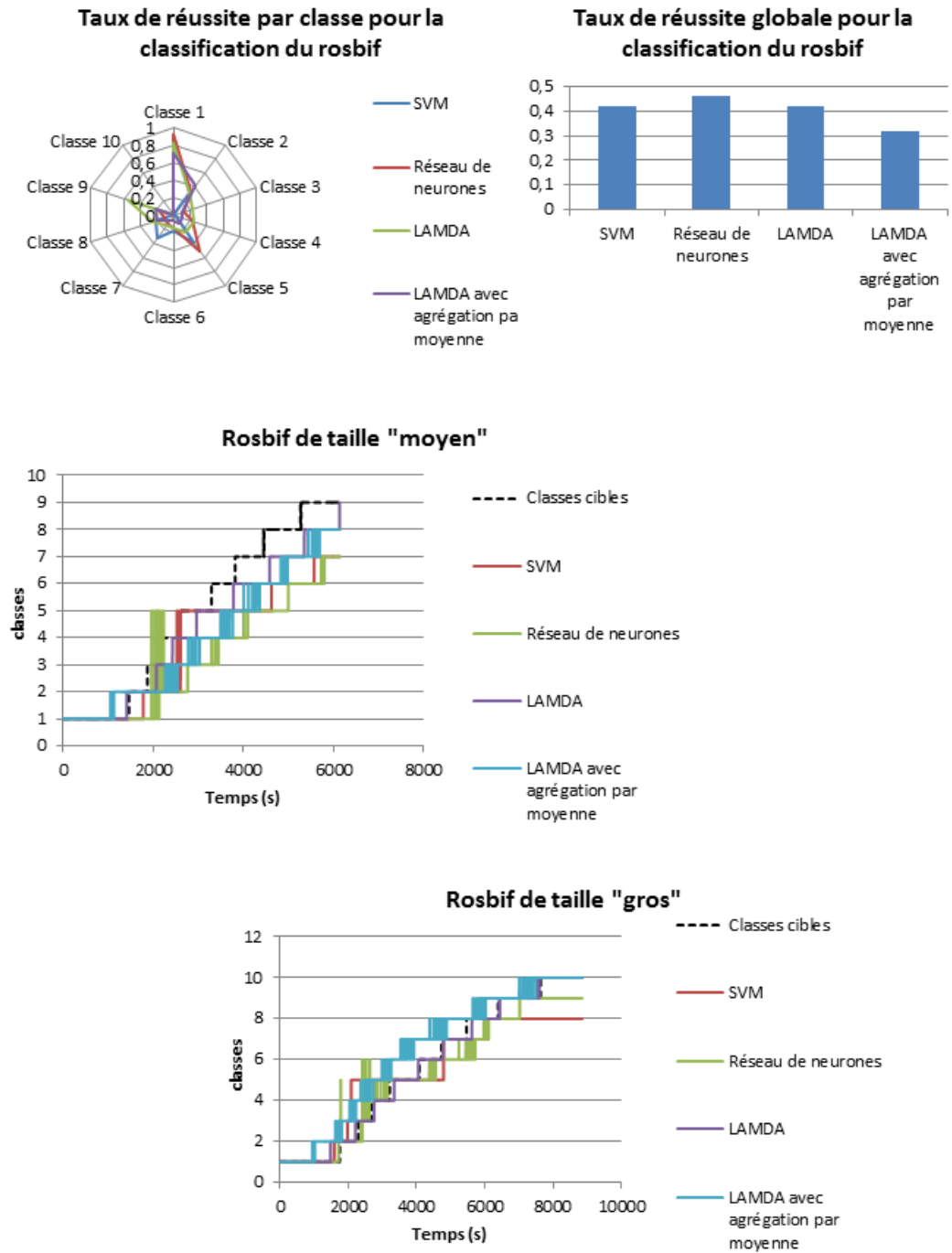
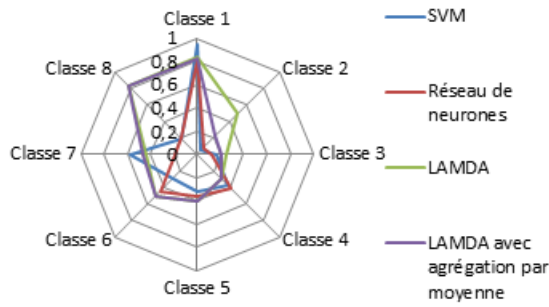
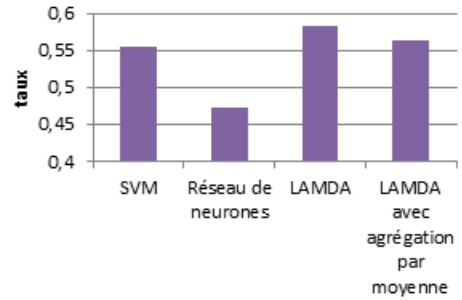


FIGURE 5.25 – Résumé d’informations sur la classification dans le prototype v2 pour du rosbig

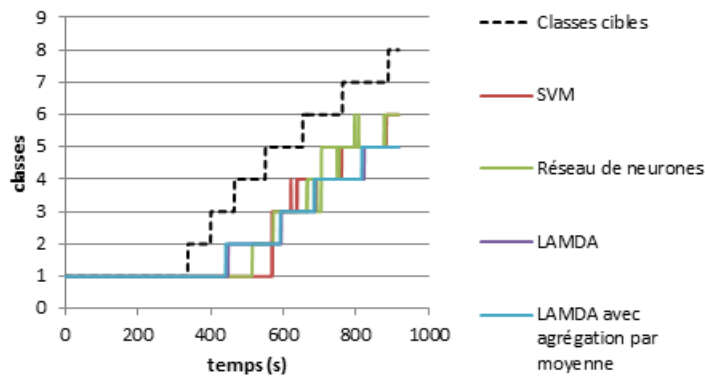
### Taux de réussite par classe pour la classification du saumon



### Taux de réussite globale pour la classification du saumon



### Saumon en faible quantité



### Saumon en grande quantité

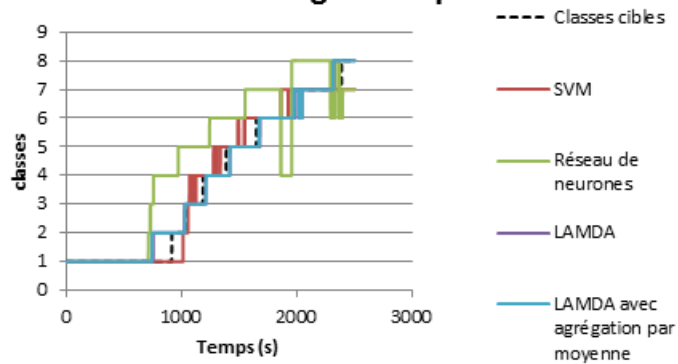


FIGURE 5.26 – Résumé d'informations sur la classification dans le prototype v2 pour du saumon



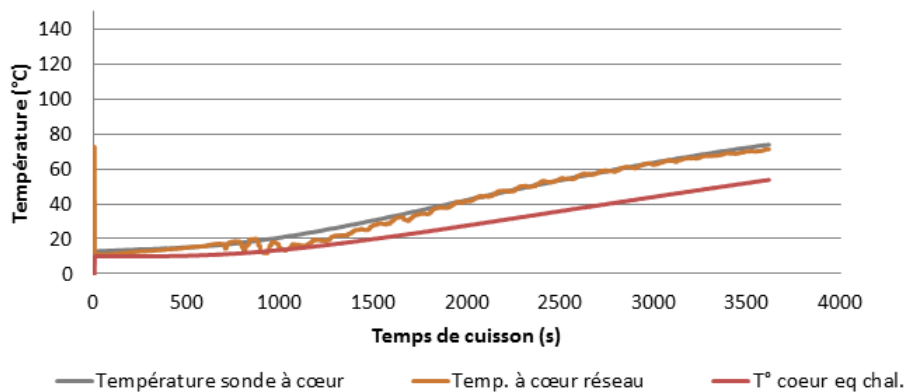


FIGURE 5.27 – Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole

## 5.5 Reconstitution de la température à cœur à l'aide de réseaux de neurones artificiels

En plus de l'équation de la chaleur et des algorithmes de classification, une troisième méthode a été appliquée pour déduire des mesures du four l'état de cuisson du produit. Nous avons effectué une régression de la température au cœur des aliments en utilisant des réseaux de neurones artificiels. L'idée est d'utiliser la capacité des réseaux de neurones à générer des liens non linéaires de manière automatique pour trouver un modèle entre les données mesurées par les capteur du prototype v2 et la température délivrée par la sonde à cœur.

Les entrées du réseaux sont reprises directement des résultats de sélection de la section précédente. Donc dans le cas du rosbif, la cinquième variable intégrée du capteur d'odeur, la température de surface et la valeur cuisatrice. Ce type de réseau a été implémenté sur le microcontrôleur du four avec cinq neurones sur la couche cachée. Ce faible nombre de neurones est choisi pour éviter le sur-apprentissage qui arrive très rapidement sur ces données. Les résultats sur les figures 5.27, 5.28, 5.29, 5.30 et 5.31 ont été obtenus sur des cuissons réelles, la sortie des réseaux de neurones étant calculée par le calculateur du four.

Cette méthode offre des résultats imparfaits, mais toutefois très encourageants. On constate sur les cas où la taille des rosbifs est celle du protocole du projet, que l'on arrive à des erreurs inférieures à 2 °C sur l'intervalle de températures entre 30° C et 70° C. Les résultats sont proches si le rôti est allongé par rapport à la taille standard, ceci était partiellement prévisible du fait que l'on a déjà déterminé dans la section consacrée à l'équation de la chaleur que la variation de longueur n'avait pas beaucoup d'impact sur l'évolution de la température au cœur des aliments. L'ajout d'aromates semble par contre altérer les résultats, les erreurs devenant alors plus grandes (15 °C d'écart entre le modèle et la mesure). Avec un seul exemple, on

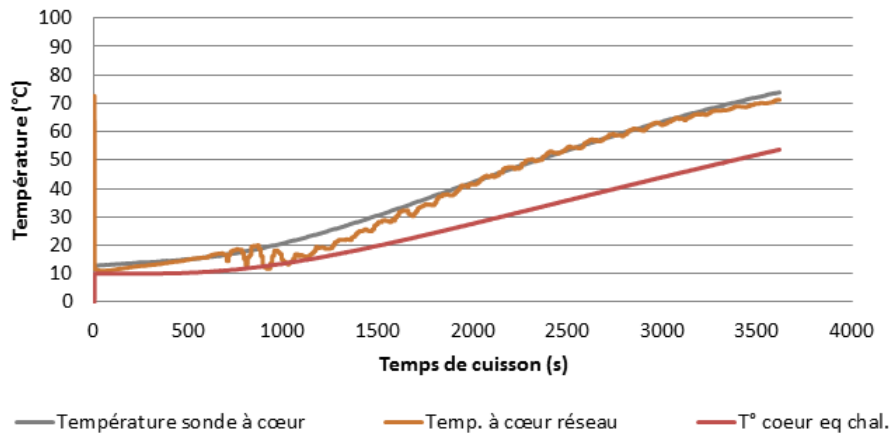


FIGURE 5.28 – Deuxième résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole

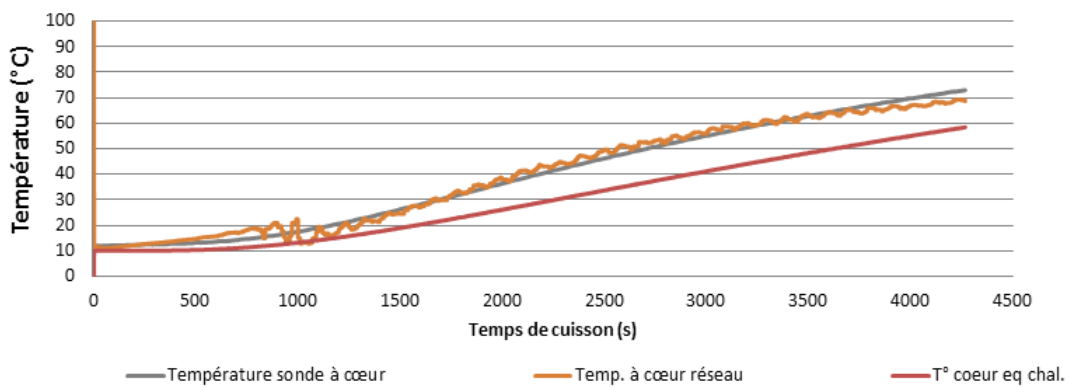


FIGURE 5.29 – Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif plus long que celui décrit par le protocole

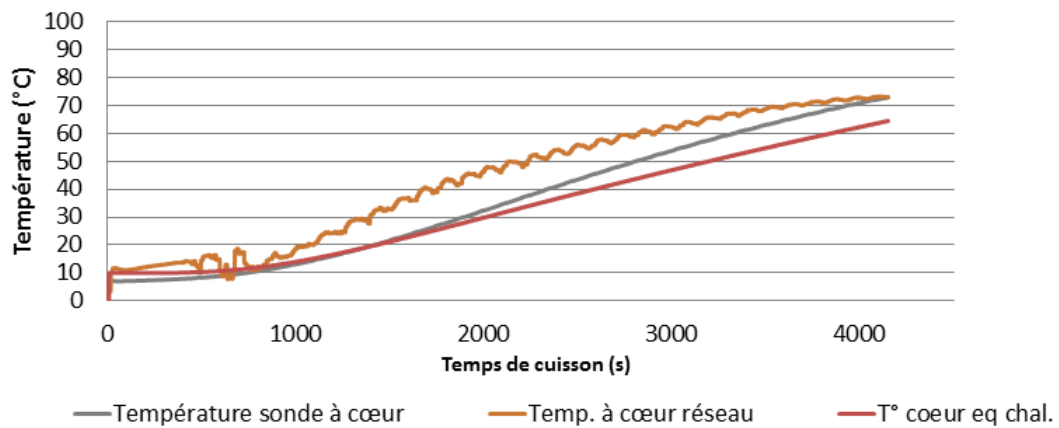


FIGURE 5.30 – Résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif de taille en accord avec le protocole et en présence d’aromates

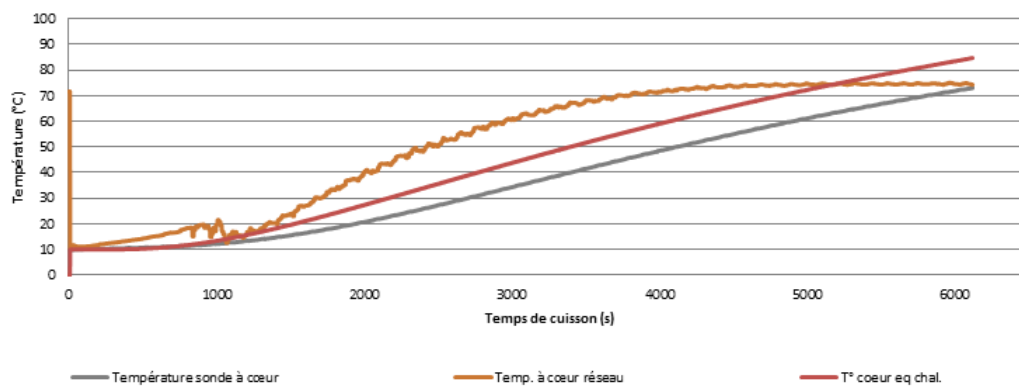


FIGURE 5.31 – Deuxième résultats du calcul de la température à cœur par un réseau de neurones dans le prototype v2 sur du rosbif plus long et plus épais que celui décrit par le protocole

ne peut toutefois pas généraliser ce résultat pour savoir si cela vient vraiment des aromates, ou d'un autre facteur. L'augmentation du diamètre détériore également les résultats, le réseau n'étant pas capable de prendre en compte le changement de taille qui entraîne une évolution plus lente de la température. Sur ces exemples, on peut aussi observer l'évolution du calcul de l'équation de la chaleur, dont le paramètre  $r$  était fixé à sa valeur moyenne, ce qui rend le modèle incapable de suivre l'évolution réelle dans la plupart des cas.

Il est à noter que par la suite un filtre numérique passe-bas a été ajouté dans le logiciel du four pour supprimer les oscillations produites sur les résultats du réseau de neurones.



# Conclusion

Durant cette thèse, plusieurs étapes ont été effectuées de manière chronologique pour permettre le suivi de cuisson en temps réel d'aliments dans un four. Ceci passe par l'analyse à base de sélection d'attributs d'expériences réalisées en laboratoire, afin de trouver quelles caractéristiques physiques étaient les plus pertinentes pour permettre de retrouver l'état de cuisson. Cela a mené à la construction de capteurs spécifiques et à la réalisation de fours prototypes embarquant ces capteurs et un logiciel de traitement des données. Le logiciel intégré dans le four est issu de mes travaux et devait répondre à plusieurs objectifs et contraintes émanant de l'industriel porteur du projet en particulier. Au final, en plus d'intégrer un algorithme de classification (ce qui était l'objectif initial), il s'appuie sur deux algorithmes supplémentaires qui sont d'une part la résolution de l'équation de la chaleur et la reconstruction par un réseau de neurones artificiels, tout ceci en temps réel bien-sûr.

L'étape de sélection a permis d'isoler plusieurs caractéristiques, qu'elles soient au niveau olfactif ou au niveau du rayonnement par réflexion sous lumière blanche ou par excitation fluorescente. Certaines molécules ont pu être rattachées à des odeurs liées au monde de la cuisine ce qui apporte une certaine cohérence. Un protocole précis concernant les cuissons étant défini, il est possible de vérifier si l'une ou l'autre des méthodes employées apporte plus d'informations sur un même produit, par contre il est impossible de vérifier si les caractéristiques extraites des expériences sont robustes à des variations physiques (masse, origine de la viande, ...). De plus, les coûts et le temps nécessaire à la réalisation des cuissons n'a pas permis d'effectuer un très grand nombre d'expériences pour rendre les résultats fiables sur un plan statistique. Toutefois les résultats obtenus à l'aide de l'algorithme de sélection MEMBAS offrent de bonnes pistes pour des études plus poussées sur la physique et la chimie des cuissons de viandes et poissons.

Les aléas du projet ont fait qu'il a été très compliqué d'obtenir des prototypes fiables dans des délais qui auraient permis des études complémentaires sur les résultats obtenus par l'analyse des données des cuissons en laboratoire. Ces prototypes ont néanmoins servi à tester et mettre en place des algorithmes pour le traitement des données avec les capteurs dans leur version définitive. Plusieurs algorithmes de traitement des données du four ont été développés autour de trois types de méthodes : la résolution d'une équation physique, l'équation de la chaleur, des algorithmes de classification de données et une régression sur la température mesurée au cœur des aliments à l'aide de réseaux de neurones.

L'équation de la chaleur telle qu'elle a été utilisée dans le cadre de cette thèse permet de modéliser assez précisément le comportement de la température au centre des aliments en mesurant uniquement la température de surface. Cette équation a tout de même besoin d'un paramètre qui exprime la capacité à conduire la chaleur depuis la surface vers le centre et ce dernier reste extrêmement compliqué à déterminer en utilisant uniquement les données capteur du four. Cette méthode

offre donc la garantie d'un résultat fiable, à condition de connaître ce paramètre de conduction, ce qui n'est pas le cas actuellement.

Dans le cas de la classification, il a été développé au cours de cette thèse des compléments pour essayer de modéliser des incertitudes dans l'utilisation de la méthode de classification LAMDA. Ceci passe par la transformation de données numériques simples en intervalles flous de formes triangulaires pour approximer des erreurs gaussiennes et la mise en place d'une nouvelle méthode d'agrégation par moyenne pondérée, les coefficients de pondération offrant une mesure de la fiabilité de chaque variable utilisée. Les intervalles flous ont montré une amélioration des résultats par rapport à l'utilisation de données scalaires sur certaines variables, ceci étant démontré par des scores sur l'algorithme de sélection MEMBAS plus élevés. La méthode d'agrégation floue a montré moins de succès puisque sur les exemples utilisés, elle offre de moins bons résultats généraux que les méthodes d'agrégation classiques.

Pour classer les données des cuissons dans les prototypes de four, plusieurs algorithmes de classification ont été appliqués sur du rosbif et du saumon. Dans les deux cas, les variations de taille des morceaux posent problème, les algorithmes ne permettant pas de suivre de manière fiable l'état au cœur des aliments, particulièrement sur les zones critiques entre « saignant » et « bien-cuit ».

Enfin une régression à l'aide de réseaux de neurones a été testée avec des résultats dans un cas de cuissons réelles. Cette méthode retourne des températures très précises pour des morceaux dont la taille correspond aux standards du projet, mais présente de grands écarts si l'épaisseur des aliments augmente.

Au final, toutes les méthodes testées ont le potentiel de fonctionner avec de bons niveaux de précision si un problème est résolu : la connaissance de la taille de l'aliment. En effet, on constate pour toutes les méthodes appliquées que c'est le changement de diamètre ou, dans un cas plus général, de masse qui vient perturber la méthode. Néanmoins avec la technologie présente, il est compliqué de retrouver cette information. Pour pallier cela, on peut envisager soit de demander à l'utilisateur du four de fournir une valeur ou un indicateur qui renseigne sur la taille de l'élément à cuire, soit d'utiliser des capteurs capables de fournir cette information. De même, il est possible que théoriquement des réseaux de type « Deep learning » soient capables de résoudre ce problème mais le nombre de cuissons à réaliser et le coût calculatoire pour appliquer ce type de réseau dans le four rendent impossible la mise en place en pratique d'une telle solution.

Malgré tout, la méthodologie appliquée reste valide et pourrait être mise en œuvre sur d'autres projet industriels. De plus, si la taille des aliments pouvait être connue, le projet sur lequel nous avons travaillé durant cette thèse pourrait connaître des débouchés industriels fiables et certains.

# Formulation de l'algorithme MEMBAS

---

Soit le problème d'optimisation à résoudre :

$$\max(WM^c - WM^{nc})$$

avec les contraintes :  $\sum_{n=1}^N w_n^2 = 1$  et  $\forall n \in [1, \dots, N]$ ,  $w_n > 0$

Sa résolution peut être effectuée en recherchant le minimum du Lagrangien associé, qui s'écrit :

$$L = -W's + \lambda(\|W\|_2^2 - 1) + \sum_{n=1}^N \xi_n(-W) \quad (\text{A.1})$$

où  $\lambda$  et  $\xi$  sont respectivement les multiplicateurs de Lagrange et de Kuhn-Tucker. En appliquant la condition du premier ordre pour avoir un optimum, c'est-à-dire le gradient de L par rapport aux poids  $W$  égal à zéro, on obtient la solution :

$$\frac{\partial L}{\partial W} = -s + 2\lambda W - \xi = 0 \quad (\text{A.2})$$

Cette condition est nécessaire et suffisante pour que cet optimum soit un minimum de par la propriété de la convexité du critère et des contraintes. En appliquant les conditions sur les multiplicateurs de Kuhn-Tucker c'est-à-dire :  $\sum_n \xi_n w_n = 0$ . Trois cas doivent être considérés :

1.  $s_n = 0 \Rightarrow \xi_n = 0$  et  $w_n = 0$
2.  $s_n > 0 \Rightarrow s_n + \xi_n > 0 \Rightarrow w_n > 0 \Rightarrow \xi_n = 0$
3.  $s_n < 0 \Rightarrow \xi_n > 0 \Rightarrow w_n = 0 \Rightarrow s_n = -\xi_n$

On peut donc résumer ces trois conditions sous la forme :

$$w_n = \begin{cases} 0 & \text{si } s_n \leq 0 \\ \frac{1}{2\lambda_n s_n} & \text{si } s_n > 0 \end{cases} \quad (\text{A.3})$$

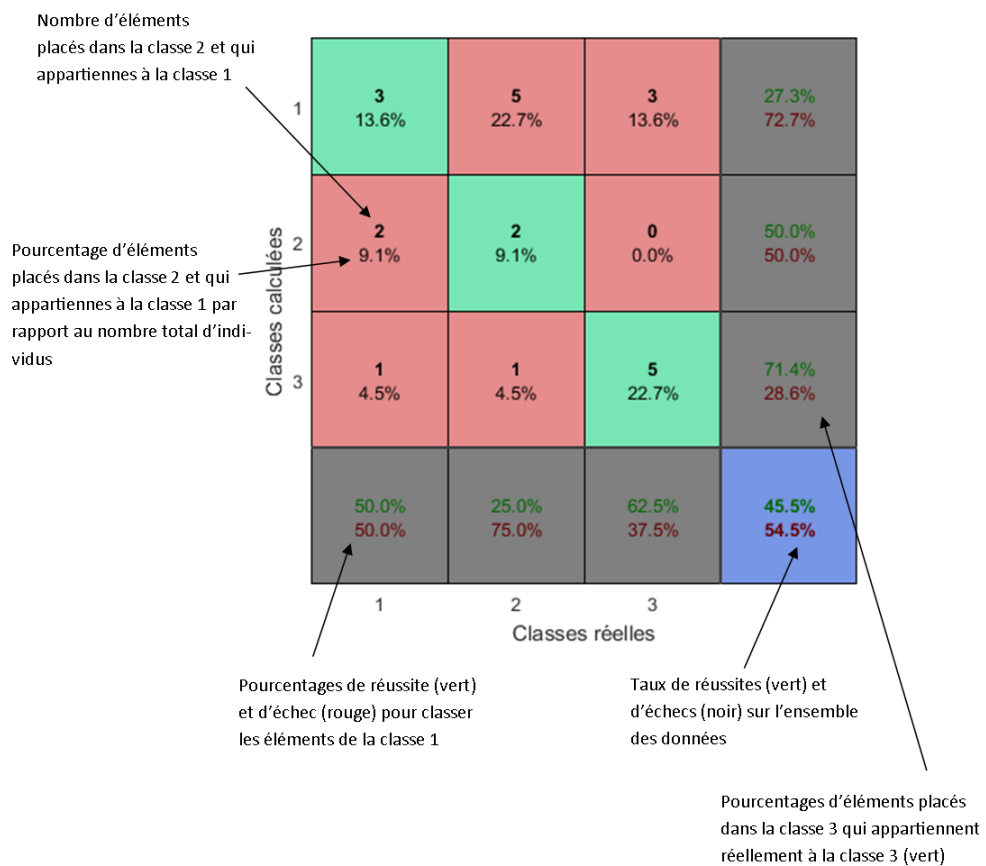
En écrivant sous une forme condensée, on obtient la forme normalisée :  $W^+ = \frac{s^+}{\|s^+\|}$

avec  $s^+ = [\max(s_1, 0), \dots, \max(s_n, 0)]$





# Explications sur la lecture des matrices de confusion





# Exemples de résolutions de l'équation de la chaleur avec paramètre optimal

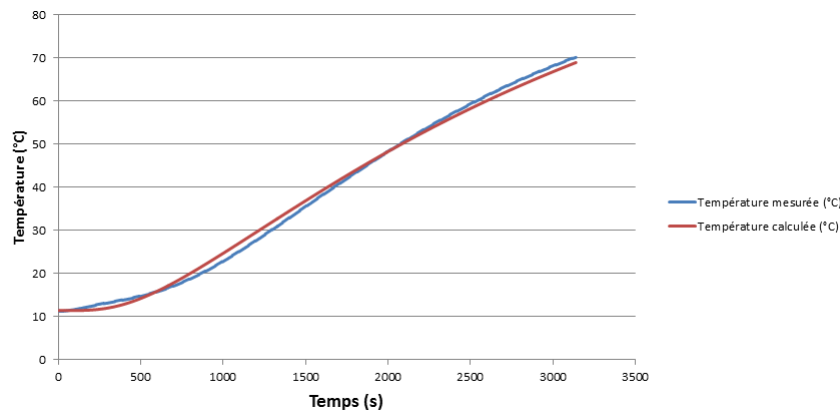


FIGURE C.1 – Exemple de résolution de l'équation de la chaleur pour du rosbif avec un paramètre d'équation optimal

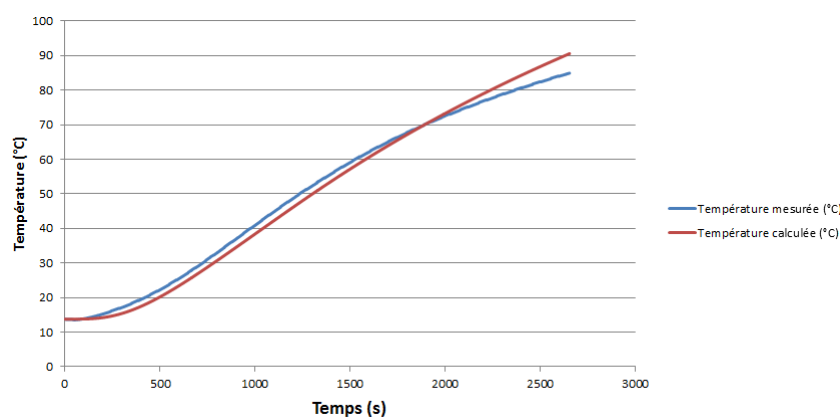


FIGURE C.2 – Exemple de résolution de l'équation de la chaleur pour du saumon avec un paramètre d'équation optimal



## Limites de classes pour la classification du rosbif et du saumon sur le prototype v2

---

Températures (°C)	Classes	Informations
	1	Cru
28		
	2	
38		
	3	Bleu
45		
	4	Saignat
54		
	5	A point
67		
	6	Bien cuit
75		
	7	
83		
	8	
91		
	9	
99		
	10	

TABLE D.1 – Tableau des températures à cœur du rosbif qui délimitent les classes pour la classification dans le prototype v2

Températures (°C)	Classes	Informations
	1	Cru
36		
	2	
42		
	3	
48		
	4	
56		
	5	Rosé
65		
	6	A point
74		
	7	
83		
	8	
91		
	9	
99		
	10	

TABLE D.2 – Tableau des températures à cœur du saumon qui délimitent les classes pour la classification dans le prototype v2

# Bibliographie

- [Aguilar-Martin 1982] J Aguilar-Martin et R López de Mántaras. *The process of classification and learning the meaning of linguistic descriptors of concepts*. Approximate reasoning in decision analysis, pages 165–175, 1982. (Cité en page 19.)
- [AroChemBase ] AroChemBase. *Module d'analyse Arochembase, analyse sensorielle de l'odeur*. Ressource web. URL : <http://www.alpha-mos.com/fr/instruments-analytiques/module-analyse-odeurs-arochembase.php> - dernière visite le 20/09/2016. (Cité en page 35.)
- [Besse 2013] Philippe Besse. Introduction à la régression multiple. Université de Toulouse, 2013. (Cité en page 28.)
- [Biem 2015] Alain Biem. *Chapter 8 : Neural Networks : A Review, Data Classification : Algorithms and Applications*, 2015. edited by Charu C. AGGARWAL, in Data Mining and Knowledge Discovery Series, Chapman and Hall - CRC, Edition Taylor and Francis Group, LLC. pages 205-246. (Cité en page 17.)
- [Burgess 1998] Christopher JC Burgess. *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery, vol. 2, no. 2, pages 121–167, 1998. (Cité en page 15.)
- [Carraro 2007] Laurent Carraro et Anca Badea. *Notions sur l'analyse discriminante*, novembre 2007. (Cité en page 11.)
- [Cunningham 2007] Padraig Cunningham et Sarah Jane Delany. *k-Nearest neighbour classifiers*. Multiple Classifier Systems, pages 1–17, 2007. (Cité en page 13.)
- [Desroches 1987] Ph Desroches. *Système de classification avec apprentissage et reconnaissance des formes. Manuel d'utilisation*. Rapport de recherche, Centre d'Etudes Avancées de Blanes, 1987. (Cité en page 19.)
- [Enas 1986] Gregory G Enas et Sung C Choi. *Choice of the smoothing parameter and efficiency of k-nearest neighbor classification*. Computers & mathematics with applications, vol. 12, no. 2, pages 235–244, 1986. (Cité en page 13.)
- [Fisher 1936] R.A. Fisher. *{UCI} Machine Learning Repository*, 1936. <http://archive.ics.uci.edu/ml>. (Cité en page 10.)
- [Goñi 2010] Sandro M Goñi et Viviana O Salvadori. *Prediction of cooking times and weight losses during meat roasting*. Journal of Food Engineering, vol. 100, no. 1, pages 1–11, 2010. (Cité en page 4.)
- [Hedjazi 2011] Lyamine Hedjazi. *Outil d'aide au diagnostic du cancer à partir d'extraction d'informations issues de bases de données et d'analyses par biopuces*. PhD thesis, Université Paul Sabatier-Toulouse III, 2011. (Cité en page 26.)



- [Holtz 1986] Eva Holtz et Christina Skjöldebrand. *Simulation of the temperature of a meat loaf during the cooking process*. Journal of Food Engineering, vol. 5, no. 2, pages 109–121, 1986. (Cit  en page 3.)
- [Hsieh 1999] C. H. Hsieh et S. H. Chen. *Similarity of generalized fuzzy numbers with graded mean integration representation*. In Proceedings of the Eighth International Fuzzy Systems Association World Congress, volume vol. 2, pages pp. 551–555, Taipei, Taiwan, Republic of China, 1999. (Cit  en page 66.)
- [Hsu 2002] Chih-Wei Hsu et Chih-Jen Lin. *A comparison of methods for multiclass support vector machines*. IEEE transactions on Neural Networks, vol. 13, no. 2, pages 415–425, 2002. (Cit  en page 16.)
- [Isleroglu 2016] Hilal Isleroglu et Figen Kaymak-Ertekin. *Modelling of heat and mass transfer during cooking in steam-assisted hybrid oven*. Journal of Food Engineering, vol. 181, pages 50–58, 2016. (Cit  en page 4.)
- [June ] June. *June Intelligent Oven - The computer-based oven that thinks like a chef*. Resource web. URL : <https://juneoven.com/> - derni re visite le 22/09/2016. (Cit  en page 5.)
- [LeCun 2015] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. *Deep learning*. Nature, vol. 521, no. 7553, pages 436–444, 2015. (Cit  en page 18.)
- [Lee 2014] Victor E. Lee, Lin Liu et Ruoming Jin. *Data classification : algorithms and applications*. CRC Press, 2014. (Cit  en page 11.)
- [Liu 2000] Yongliang Liu, Yud-Ren Chen et Yukihiko Ozaki. *Two-dimensional visible/near-infrared correlation spectroscopy study of thermal treatment of chicken meats*. Journal of Agricultural and Food Chemistry, vol. 48, no. 3, pages 901–908, 2000. (Cit  en page 54.)
- [Lu 2007] Yijuan Lu, Ira Cohen, Xiang Sean Zhou et Qi Tian. *Feature selection using principal feature analysis*. In Proceedings of the 15th ACM international conference on Multimedia, pages 301–304. ACM, 2007. (Cit  en page 26.)
- [McCulloch 1943] Warren S McCulloch et Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115–133, 1943. (Cit  en page 18.)
- [Monrousseau 2015] Thomas Monrousseau, Louise Trav -Massuy s et Marie-V ronique Le Lann. *Processing measure uncertainty into fuzzy classifier*. In 26th International Workshop on Principles of Diagnosis, 2015. (Cit  en page 64.)
- [Nishii 1995] Kazunari Nishii, Kenji Watanabe, Shigeki Ueda et Motohiko Naka. *Automatic cooking appliance employing a neural network for cooking control*, 1995. US Patent 5,389,764. (Cit  en page 5.)
- [Piera 1989] N Piera, Ph Desroches et J Aguilar-Martin. *LAMDA : An incremental conceptual clustering method*. Rapport technique LAAS-CNRS, no. 89420, 1989. (Cit  en page 19.)

- [Rawat 2014] Seema Rawat. *Challenges and opportunities with big data*. Innovation in IT, vol. 1, no. 1, pages 26–42, 2014. (Cité en page 10.)
- [Rosenblatt 1962] Frank Rosenblatt. *Principles of neurodynamics*. 1962. (Cité en page 18.)
- [Roux 2015] Elisa Roux, Louise Travé-Massuyès et Marie-Véronique Le Lann. *Applied multi-layer clustering to the diagnosis of complex agro-systems*. In 26th International Workshop on Principles of Diagnosis, 2015. (Cité en page 69.)
- [Rumelhart 1986] DE Rumelhart, GE Hinton et RJ Williams. *Learning internal representations by error propagation. Parallel distributed processing : Explorations in the microstructure of cognition, Vol. 1 : Foundations (pp. 318-362)*, 1986. (Cité en page 18.)
- [Sheridan 2002] PS Sheridan et NC Shilton. *Determination of the thermal diffusivity of ground beef patties under infrared radiation oven-shelf cooking*. Journal of Food Engineering, vol. 52, no. 1, pages 39–45, 2002. (Cité en page 4.)
- [Shilton 2002] N Shilton, P Mallikarjunan et P Sheridan. *Modeling of heat transfer and evaporative mass losses during the cooking of beef patties using far-infrared radiation*. Journal of Food Engineering, vol. 55, no. 3, pages 217–222, 2002. (Cité en page 5.)
- [Tomohiro 1999] Teruhiko Tomohiro. *Method of estimating temperature inside material to be cooked and cooking apparatus for effecting same*, 1999. US Patent 5,893,051. (Cité en page 5.)
- [Vapnik 1982] Vladimir Naumovich Vapnik et Samuel Kotz. *Estimation of dependences based on empirical data*, volume 40. Springer-Verlag New York, 1982. (Cité en page 15.)
- [Zadeh 1965] L.A. Zadeh. *Fuzzy Sets*. Information and Control, vol. 8, pages 338–353, 1965. (Cité en page 19.)
- [Zhou 2016] Bo Zhou et Hao Ye. *A study of polynomial fit-based methods for qualitative trend analysis*. Journal of Process Control, vol. 37, pages 21–33, 2016. (Cité en page 29.)
- [Ziouche 2015] Katir Ziouche, Pascale Lejeune, Tuami Lasri et Didier Leclercq. *Measure sans contact de la température de surface d'aliments en cours de cuisson dans un four en mode vapeur et mode chaleur tournante*. Sherbrooke, Québec (Canada), juin 2015. XIIème Colloque Interuniversitaire Franco-Québécois sur la Thermique des Systèmes. (Cité en page 74.)
- [Zuber 2007] François Zuber, Albert Duquenoy et Housseem Abidi. *Traitements thermiques d'appertisation : optimisation de la qualité de surface d'un produit conducteur avec un profil à température variable*. In JITH 2007, page 5. ENSTIMAC, 2007. (Cité en page 86.)



---

**Résumé :** Dans un monde où tous les appareils électroménagers se connectent et deviennent intelligents, il est apparu pour des industriels français le besoin de créer des fours de cuisson innovants capables de suivre l'état de cuisson à cœur de poissons et de viandes sans capteur au contact. Cette thèse se place dans ce contexte et se divise en deux grandes parties. La première est une phase de sélection d'attributs parmi un ensemble de mesures issues de capteurs spécifiques de laboratoire afin de permettre d'appliquer un algorithme de classification supervisée sur trois états de cuisson. Une méthode de sélection basée sur la logique floue a notamment été appliquée pour réduire grandement le nombre de variable à surveiller. La seconde partie concerne la phase de suivi de cuisson en ligne par plusieurs méthodes. Les techniques employées sont une approche par classification sur dix états à cœur, la résolution d'équation de la chaleur discrétisée, ainsi que le développement d'un capteur logiciel basé sur des réseaux de neurones artificiels synthétisés à partir d'expériences de cuisson, pour réaliser la reconstruction du signal de la température au cœur des aliments à partir de mesures disponibles en ligne. Ces algorithmes ont été implantés sur microcontrôleur équipant une version prototype d'un nouveau four afin d'être testés et validés dans le cas d'utilisations réelles.

**Mots clés :** méthodes de classification, apprentissage supervisé, logique floue, optimisation, sélection d'attributs, réseau de neurones

---