



République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique



UNIVERSITÉ DE TEBESSA
Faculté des Sciences et de la Technologie
Département de Génie Électrique

MÉMOIRE

Pour obtenir le diplôme de

Magister en Automatique

Option : Automatique et Traitement du signal

Présentée et soutenue publiquement

Par :

M. Riadh DJABRI

Le : 29 Mai 2013

Régression et identification floues

Encadré par: Dr. Noureddine GUERFI
Dr. Fayçal MEGRI

Devant le jury :

Président :	Pr. Salah CHENIKHER	Professeur	Université de TEBESSA
Rapporteur :	Dr. Noureddine GUERFI	MCA	Université de MILA
Examineur :	Pr. Mohamed Faouzi HARKAT	Professeur	Université d'ANNABA
Examineur :	Dr. Djalel DIB	MCA	Université de TEBESSA
Invité :	Dr. Fayçal MEGRI	MCB	Université de TEBESSA

Remerciements

En premier lieu je remercie Dieu tout puissant.

Je tiens tout d'abord à remercier les membres de mon jury qui ont accepté de juger ce travail de mémoire :

- Salah CHENIKHER, Professeur à l'université de TEBESSA, président de ce jury.
- Mohamed Faouzi HARKAT, Professeur à l'université d'ANNABA, examinateur.
- Djalel DIB, Maître de Conférence classe A à l'université de TEBESSA, examinateur.

Un merci particulier à Nouredine GUERFI, Maître de Conférence classe A à l'université de MILA qui a été mon encadreur principal de mémoire. Merci pour son aide et pour ses précieux conseils et sa disponibilité.

Mes remerciements s'adressent à Fayçal MEGRI, Maître de Conférence classe B à l'université de TEBESSA qui a été mon co-directeur de mémoire et encadrant durant mon travail. Merci pour ses conseils et encouragements. Je lui suis reconnaissant de m'avoir confié ce projet innovant et passionnant.

Et vous mes amis Sami, Atef, et Mohcene, je vous remercie de m'avoir écouté et encouragé tout au long de ce mémoire, et notamment d'avoir été présents le jour de ma soutenance.

Enfin, j'adresse ma reconnaissance et témoigne toute mon affection à mes parents.

« Une personne qui n'a jamais commis d'erreur ...
... n'a jamais tenté d'innover »
Albert Einstein

Table des matières

Introduction	1
I La régression floue	3
I.1 Introduction	4
I.2 Les différentes approches de la régression simple	4
I.2.1 Régression paramétrique et non paramétrique	4
I.2.2 Régression linéaire et non linéaire	5
I.3 Notations sur la logique floue	6
I.3.1 <i>Les ensembles flous</i>	7
I.3.2 <i>Les nombres flous</i>	7
I.3.3 <i>Les nombres flous L-R au sens de Dubois et Prade</i>	8
I.3.4 Intervalles flous et arithmétique associée	10
I.4 Les fondements de la régression possibiliste	10
I.5 Présentation des contraintes	14
I.6 Régression floue et la programmation quadratique (QP)	14
I.6.1 <i>Le critère de base de Tanaka</i>	14
I.6.2 <i>Le deuxième critère de Tanaka</i>	16
I.6.3 <i>La distance de Diamond</i>	17
I.7 Exemple illustratif	18
I.7.1 Les indicateurs de performance	18
I.7.2 Les données et les modèles identifiés	19
I.7.3 Un nouveau modèle linéaire	24
I.8 Conclusion	30
II La régression floue par les SVMs	31
II.1 Introduction	32

II.2	Les SVMs	33
II.2.1	<i>Principe des SVMs</i>	33
II.2.2	<i>Cas linéairement non séparable</i>	35
II.2.3	<i>Introduction des fonctions noyaux</i>	35
II.3	Régression à vecteurs de support (SVR)	36
II.3.1	Régression Linéaire	37
II.3.1.1	<i>La fonction de perte ϵ-insensitive</i>	37
II.3.1.2	<i>La fonction de perte quadratique</i>	38
II.3.1.3	<i>Exemple illustratif</i>	39
II.3.2	Régression non linéaire	40
II.3.2.1	<i>Exemple illustratif</i>	41
II.4	Régression floue à vecteur de support (FSVR)	43
II.4.1	La programmation quadratique (QP)	43
II.4.2	Extension au modèle de régression non linéaire floue	49
II.4.3	<i>Exemple illustratif</i>	50
II.5	Une nouvelle approche	53
II.5.1	La régression floue par SVM de Hong	53
II.5.1.1	Régression linéaire floue	54
II.5.1.2	Modèle de régression floue non linéaire	56
II.5.2	PSO et UPSO pour l'optimisation des hyper-paramètres	57
II.5.2.1	Algorithme standard d'optimisation par essaim de particules (PSO)	57
II.5.2.2	L'optimisation par essaim particulaire unifiée (UPSO)	59
II.5.2.3	La sélection des hyper-paramètres par standard PSO et UPSO	60
II.5.3	L'algorithme proposé et les étapes de l'identification	60
II.5.4	Exemples illustratifs	62
II.6	Conclusion	66
 III Régression floue : Application		 67
III.1	Introduction	68
III.2	Identification de modèles multi-entrées	68
 Conclusions et perspectives		 76
 A Le jeu de données multivariables		 77

Bibliographie	81
---------------	----

Liste des tableaux

I.1	Le jeu de données observées	20
I.2	Paramètres des modèles triangulaire obtenus et indicateur de performance . .	20
I.3	Paramètres des modèles trapézoïdaux obtenus et indicateur de performance .	20
I.4	Paramètres des modèles triangulaire identifié et indicateurs de performance associés	22
I.5	Paramètres des modèles obtenus et indicateur de performance	23
I.6	Le jeu de données observées	25
I.7	Valeur du critère d'optimisation selon l'ordre N de modèles identifiés	28
I.8	Paramètres des modèles d'ordre 4 obtenus pour $\text{shift} = 30$ et $\text{shift} = 1$	29
II.1	Le jeu de données observées	39
II.2	Le jeu de données observées	62
II.3	Performances d'identification et les hyper-paramètres optimaux pour les mo- dèles des deux exemples	63
III.1	Test sur les données de validation pour le modèle1 par l'approche proposée UPSO-FSVR	70
III.2	Test sur les données de validation pour le modèle1 par la technique linéaire .	71
III.3	Test sur les données de validation pour le deuxième modèle par l'approche proposée UPSO-FSVR	73
III.4	Test sur les données de validation pour le deuxième modèle	74
III.5	Performances d'identification et les hyper-paramètres optimaux pour les deux modèle multi-entrées	74
III.6	Performances d'identification et la somme des résidus pour les deux modèle multi-entrées	75

Table des figures

I.1	Représentation d'un nombre flou et d'une h -coupe	9
I.2	Modèle flou triangulaire identifié	21
I.3	Modèle flou trapézoïdale identifié	21
I.4	Modèle flou triangulaire identifié ($2^{\text{ème}}$ ordre)	23
I.5	Modèle flou trapézoïdale identifié ($2^{\text{ème}}$ ordre)	24
I.6	Modèle identifié d'ordre 1	26
I.7	Mdèle identifié d'ordre 2	26
I.8	Mdèle identifié d'ordre 3	27
I.9	Modèle identifié d'ordre 4	27
I.10	Modèle identifié d'ordre 5	28
I.11	Représentation du modèle identifié d'ordre 4 pour une valeur erronée de décalage	29
II.1	Hyperplan avec 3 vecteurs de support	33
II.2	Les fonctions de perte, [13]	36
II.3	Régression linéaire avec les SVMs	40
II.4	Régression non linéaire polynomiale de degré 2	42
II.5	Régression non linéaire avec RBF noyau	42
II.6	Régression non linéaire avec exp RBF noyau	43
II.7	Explication du modèle avec h -coupe, [14]	45
II.8	Explication du modèle de régression linéaire floue, [14]	46
II.9	Représentation du modèle linéaire identifié par SVM floue	51
II.10	Représentation du modèle linéaire identifié par SVM simple	51
II.11	Modèle non linéaire identifié par SVM floue (noyau polynomial de degré 3) .	53
II.12	Schéma de principe du déplacement d'une particule, [22]	58
II.13	Diagramme de l'algorithme proposé	61

II.14 Modèle de régression floue non linéaire obtenue par UPSO-FSVR pour l'exemple1	64
II.15 Modèle de régression floue non linéaire obtenue par standard PSO-FSVR pour l'exemple1	65
II.16 Modèle de régression floue non linéaire obtenue par UPSO-FSVR pour l'exemple2	65
II.17 Modèle de régression floue non linéaire obtenue par standard PSO-FSVR pour l'exemple2	66
III.1 Modèle multivariable obtenu par l'approche proposée (Modèle1)	69
III.2 Modèle obtenu par l'approche linéaire (Modèle1)	70
III.3 Modèle multi entrées obtenu par l'approche proposée (Mdèle2)	72
III.4 Modèle multi entrées obtenu par la technique linéaire (Mdèle2)	73

Introduction

Depuis environ 30 ans, les travaux de recherche sur la régression floue (modèles régressifs avec des paramètres flous) se sont multipliés et il est maintenant très difficile d'établir une liste exhaustive de toutes les stratégies développées.

A notre connaissance, les premiers développements ont été initiés par Tanaka et al. [30]. L'idée de base consistait à exploiter des concepts possibilistes pour identifier les paramètres d'un modèle linéaire en minimisant la dispersion des coefficients. Un programme d'optimisation linéaire sous contraintes traduit alors la stratégie mise en œuvre pour réaliser cette identification. Les contraintes sont dictées par la réalisation d'une inclusion des sorties observées dans celles prédites par le modèle (modèle dit de possibilité).

Parallèlement à ces travaux, l'idée d'une identification basée sur la minimisation d'un critère quadratique (travaux de Diamond [10],[11]) se concrétise avec l'apparition des moindres carrés flous. De par sa très grande capacité d'adaptation et d'intégration avec la majorité des méthodes d'identification précises, il nous paraît tout à fait naturel d'exploiter l'idée des moindres carrés dans un contexte incertain. En effet, l'idée directrice sous-jacente réside dans la minimisation de l'erreur quadratique de la sortie, exprimée en termes de distances entre intervalles flous (sorties observées et sorties prédites). Dans ce contexte, diverses distances ont été exploitées en tant que critère à optimiser (distance de Diamond [10], distance de Ming,...).

Cependant, si cette méthode donne souvent des résultats pertinents, de par son principe elle est souvent sujette à la violation de la contrainte d'inclusion. Dans ce contexte, afin de remédier à ces problèmes, des contraintes ont été introduites dans le problème d'optimisation [31], [32] pour aboutir à une méthodologie de programmation quadratique sous contraintes.

Notre travail consiste à effectuer une analyse à caractère expérimental des différences qui existent entre les approches conventionnelles en termes de robustesse. Pour cela on a choisi d'aborder le problème de manière pratique en s'appuyant sur des simulations permettant de mettre en évidence la sensibilité de chaque méthode par rapport à toute modification des données observées, ou à un changement de domaine d'apprentissage.

Nous avons organisé la présentation de ce mémoire en trois chapitres. Dans le premier chapitre, nous avons commencé par donner des supports théoriques sur la logique floue et la régression floue. Dans le deuxième chapitre, nous présentons la régression floue par les machines à vecteur de support (les SVMs), ainsi que notre approche proposée, et quelques exemples illustratifs montrant la robustesse. Une application multi-entrées bruitées est détaillée dans le troisième chapitre. A la fin, on terminera par une conclusion générale ou on récapitulera tous les résultats obtenus dans ce travail.

Chapitre I

La régression floue

I.1 Introduction

L'origine du mot régression vient de Sir Francis Galton. En 1885, travaillant sur l'hérédité, il chercha à expliquer la taille des fils en fonction de celle des pères. Il constata que lorsque le père était plus grand que la moyenne, "taller than mediocrity", son fils avait tendance à être plus petit que lui et, a contrario, que lorsque le père était plus petit que la moyenne, "shorter than mediocrity", son fils avait tendance à être plus grand que lui. Ces résultats l'ont conduit à considérer sa théorie de "regression toward mediocrity". Cependant l'analyse de causalité entre plusieurs variables est plus ancienne et remonte au milieu du XVIII^e siècle. En 1757, R. Boscovich, né à Ragussa, l'actuelle Dubrovnik, proposa une méthode minimisant la somme des valeurs absolues entre un modèle de causalité et les observations. Ensuite Legendre dans son célèbre article de 1805, « Nouvelles méthodes pour la détermination des orbites des comètes », introduit la méthode d'estimation par moindres carrés des coefficients d'un modèle de causalité et donna le nom à la méthode. Parallèlement, Gauss publia en 1809 un travail sur le mouvement des corps célestes qui contenait un développement de la méthode des moindres carrés, qu'il affirmait utiliser depuis 1795 [8].

Dans ce chapitre, nous allons analyser les différentes techniques de la régression floue linéaire et non linéaire : nous verrons que c'est une technique statistique permettant de modéliser la relation entre une ou plusieurs variables explicatives et une variable à expliquer. Cette présentation va nous permettre d'exposer la régression dans un cas simple afin de bien comprendre les enjeux de cette méthode, les problèmes posés et les réponses apportées.

I.2 Les différentes approches de la régression simple

I.2.1 Régression paramétrique et non paramétrique

Dans une approche paramétrique [2], l'utilisateur doit définir la forme mathématique du modèle à identifier. Ainsi, cela lui permet d'avoir une action de contrôle sur le nombre de paramètres de ce modèle, paramètres qui seront donc les inconnues à déterminer à l'aide de la technique régressive. Il lui sera donc possible de chercher à réduire le nombre de ces paramètres, en vue de simplifier le modèle, et donc de faciliter son exploitation une fois la régression réalisée. Cette action de contrôle de la complexité du modèle, donc du nombre des paramètres n'est pas possible dans une approche non paramétrique, puisque c'est la meilleure représentation des données qui va être recherchée, quelque soit la complexité du modèle ainsi

obtenu. Ainsi, les approches non paramétriques fournissent un modèle qui ne peut pas être décrit par un petit nombre de paramètres. La seule interprétation possible de ce modèle par l'utilisateur ne pourra donc se faire qu'au travers d'une représentation graphique.

Bien évidemment, le fait que l'utilisateur doit déterminer la forme mathématique du modèle à identifier dans une approche paramétrique peut aussi être vu comme un inconvénient. En effet, le modèle spécifié par l'utilisateur peut être inapproprié pour représenter au mieux les données, quels que soient les paramètres optimaux fournis par la technique régressive, aussi performante soit-elle. Cela peut notamment être le cas si les connaissances à priori de l'utilisateur sur les données à analyser sont insuffisantes pour proposer une structure de modèle adéquate. Cependant on remarquera que le choix d'une approche non paramétrique ne doit se faire que si les données à analyser sont suffisamment riches en informations pour permettre de déterminer et la structure du modèle, et les nombreux paramètres correspondants lors de la phase de régression.

Ces différents avantages et inconvénients entre approches paramétrique et non paramétrique ont été soulignés dans des études menées dans le cadre conventionnel. Dans le contexte d'imprécision qui nous intéresse, plusieurs approches non paramétriques peuvent être distinguées. Certains travaux concernent l'utilisation des Support Vector Machines dans un cadre de régression non paramétrique. Citons par exemple les travaux de Hao, Hwang. Il s'agit dans ces approches de déterminer un modèle imprécis, soit sur un jeu de données précises, soit sur des données elles-mêmes imprécises, ce qui correspond donc à notre définition contextuelle de la régression floue.

Un des avantages de ces méthodes, mis en avant dans les travaux cités précédemment, concerne là encore la non-nécessité d'avoir une connaissance à priori de la structure du modèle à identifier. Malheureusement, on retrouve dans ces approches non paramétriques imprécises le fait que les modèles obtenus ne peuvent être interprétés qu'au travers de la représentation graphique de l'évolution des données. Ce dernier point est particulièrement vrai dans le cas de méthodes régressives robustes aux éventuels points aberrants, ou encore dans des approches appliquées sur des données ayant été classifiées, afin d'en dégager différentes tendances distinctes lors d'une phase de clustering initiale.

I.2.2 Régression linéaire et non linéaire

Les techniques régressives paramétriques [2] permettent de modéliser par une expression analytique, une relation existant entre des données numériques. Celles-ci, collectées par un utilisateur en vue de leur analyse, sont composées de manière générale :

- d’une variable dépendante, également nommée réponse, sortie, ou bien encore mesure, et notée y .
- d’une ou plusieurs variables indépendantes, également nommées entrées, et notées $\mathbf{x} = [x_i, i = 1, \dots, N]$, ou \mathbf{x} est le vecteur d’entrées de N composantes.

Un modèle non linéaire pouvant avoir une forme mathématique quelconque, il est évident qu’il présente un meilleur potentiel de représentativité des données observées qu’un modèle linéaire en ses paramètres, pour lequel la contrainte de linéarité limite le choix de sa structure mathématique. Cependant, cette linéarité en les paramètres imposée dans le choix de la forme du modèle dans un cadre de régression paramétrique linéaire permet de garantir l’obtention d’un modèle représentatif de forme analytique simple, et donc facilement exploitable par l’utilisateur.

Il est également possible de linéariser un modèle non linéaire, afin de faciliter l’identification de ses paramètres (l’exemple le plus simple concerne les modèles exponentiels, qui peuvent être linéarisés en combinant la fonction les définissant avec un logarithme népérien). Ainsi, il est envisageable dans certains cas de pouvoir appliquer une technique régressive paramétrique linéaire bien que les données collectées initialement imposeraient le choix d’un modèle non linéaire.

Afin de respecter les contraintes imposées par la définition retenue de l’analyse régressive, on privilégiera l’obtention d’un modèle de forme analytique simple et donc facilement exploitable et inversible le cas échéant par l’utilisateur. Ainsi, dans la suite, nous nous focaliserons sur les techniques régressives paramétriques linéaires et quadratiques. Afin de prendre en compte le contexte d’imprécision dans lequel nous plaçons nos travaux, nous étudierons donc les techniques de régression paramétriques linéaires et quadratiques adaptées à ce cadre.

Dans le contexte imprécis qui nous intéresse particulièrement ici, il faut connaître les notations de base de la logique floue [2].

I.3 Notations sur la logique floue

La théorie des ensembles flous a été introduite par Zadeh[39] en 1965. Elle a longtemps concerné un nombre restreint d’adeptes parmi les chercheurs et les enseignants. Utilisée en priorité en mathématique, elle s’est révélée féconde dans de nombreuses applications. Un ensemble flou est un ensemble dont les limites ne sont pas spécifiquement définies, par opposition aux ensembles classiques. Le flou fait correspondre un ensemble à une notion vague du langage (par exemple grand, moyen, petit).

I.3.1 Les ensembles flous

Un ensemble flou A est défini sur un univers U par une fonction d'appartenance notée μ_A qui retourne une valeur comprise entre zéro et un pour tout, élément u de U :

$$0 \leq \mu_A(u) \leq 1$$

Chaque ensemble flou possède plusieurs caractéristiques dont voici les deux principales :

- Le support d'un ensemble flou représente l'ensemble des éléments de U dont le degré d'appartenance est non nul [10] :

$$Supp(A) = \{u \in U; \mu_A(u) \neq 0\}$$

- Le support d'un ensemble à un niveau h , noté support- h , représente l'ensemble des éléments de U dont le degré d'appartenance est supérieur ou égal à h :

$$Supp_A(h) = \{u \in U; \mu_A(u) \geq h\}$$

- La hauteur d'un ensemble flou est représentée par la valeur maximale de la fonction d'appartenance sur U [33] :

$$\alpha(A) = \max(\mu_A(u); u \in U)$$

I.3.2 Les nombres flous

Les nombres flous sont des ensembles flous particuliers qui vérifient certaines propriétés [10]

- U représente l'ensemble des réels.
- La hauteur de l'ensemble est égale à 1.
- La fonction d'appartenance est convexe :

$$\forall a, \forall b, \forall c, \text{ si } a < b < c \text{ alors } \mu_A(b) \geq \text{Min}(\mu_A(a), \mu_A(c))$$

Ils sont généralement représentés de la manière suivante : $A(m, r_L, r_R)_{RL}$, [27].

- m représente le centre de A (L'élément dont la valeur de la fonction d'appartenance est égale à 1) pour les nombres flous triangulaires et un interval pour les nombres flous trapésoïdals, Dans cette formulation des nombres flous, le centre est unique.
- r_L représente le support de A situé à gauche du centre.
- r_R représente le support de A situé à droite du centre.
- L et R représentent les fonctions d'appartenance respectivement à gauche et à droite du centre.

I.3.3 Les nombres flous L - R au sens de Dubois et Prade

Ce sont des cas particuliers des nombres flous dits nombres L - R qui se définissent de la manière suivante :

$$L(x) = 1 - x^p \quad \text{et} \quad R(x) = 1 - x^q \quad (\text{I.1})$$

Avec ' p ' et ' q ' deux réels généralement compris entre 0 et 1. Lorsque $p = q = 1$ le nombre flou défini est un nombre triangulaire et si ' p ' et ' q ' sont supérieurs à 1, le nombre flou obtenu est un nombre trapézoïdal.

Cette définition des nombres L - R est la plus utilisée, puisqu'elle reflète parfaitement les courbes curvilignes des nombres L - R , et propose une infinité de courbes tout en étant fort bien utilisable par des outils informatiques.

La plupart des travaux concernant la régression floue portent sur des nombres flous triangulaires et trapézoïdales symétriques avec alors $L(x) = \max(0, 1 - |x|)$.

Dans ce cas, Considérant la classe L des nombres LR -flous, un nombre flou trapézoïdal A (figure I.1 - a -), est défini par les intervalles correspondant à son support et à son noyau, on le note comme suit :

$$A = (K_A, S_A) = ([K_A^-, K_A^+], [S_A^-, S_A^+]) \quad (\text{I.2})$$

Sa fonction d'appartenance est définie par :

$$\mu_A(x) = \begin{cases} \frac{x - S_A^-}{K_A^- - S_A^-} & \text{pour } (S_A^- < x < K_A^-) \\ 1 & \text{pour } (K_A^- < x < K_A^+) \\ \frac{S_A^+ - x}{S_A^+ - K_A^+} & \text{pour } (K_A^+ < x < S_A^+) \\ 0 & \text{ailleurs} \end{cases} \quad (\text{I.3})$$

Les nombres flous triangulaires symétriques (NFT) sont des cas particulières, définis par leur noyau $K_A = m$, et leur radius $R_A = r$, noté (m, r) .

On a dans ce cas :

$$K_A^+ = K_A^- = K_A, \quad S_A^+ = K_A + R_A, \quad S_A^- = K_A - R_A. \quad (\text{I.4})$$

Et aussi par leur fonction d'appartenance suivante :

$$\mu_A(x) = \begin{cases} 1 - \frac{|m-x|}{r} & \text{pour } (m - r \leq x \leq m + r) \\ 0 & \text{ailleurs} \end{cases} \quad (\text{I.5})$$

Avec (m) la valeur modale et (r) la dispersion (identique à gauche et à droite). La représentation d'un nombre flou de ce type est donné en figure I.1 -b-.

Il est intéressant également ici de définir la notion de h -coupe, h représentant un indice de confiance fourni par un expert, compris entre 0 et 1. Réaliser une h -coupe sur un nombre flou triangulaire symétrique A revient ne considérer que la partie du nombre pour la quelle l'appartenance est supérieur ou égale à h , c'est-à-dire :

$$\begin{aligned} A_h &= \{x \mid \mu_A(x) \geq h\} \\ &= [m - (1 - h)r, m + (1 - h)r] \end{aligned} \tag{I.6}$$

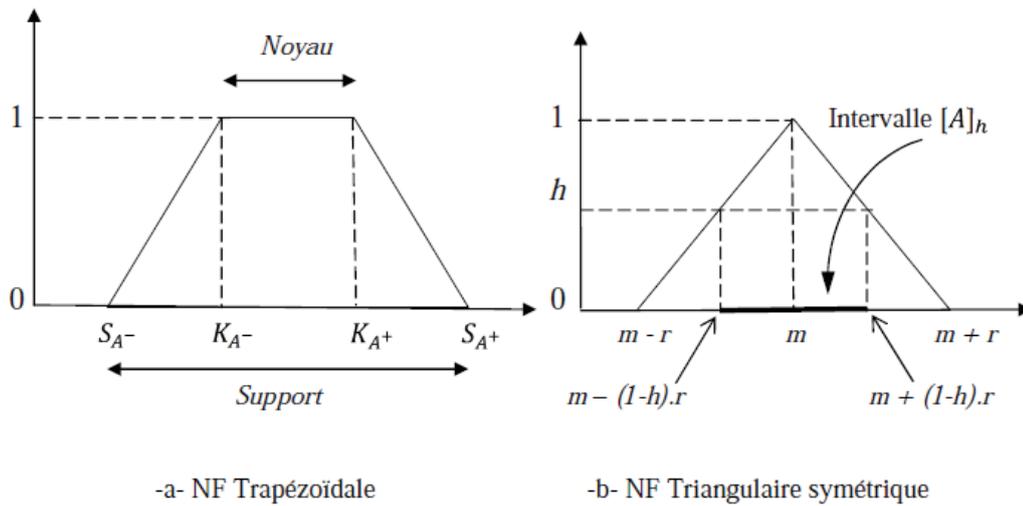


FIG. I.1 – Représentation d'un nombre flou et d'une h -coupe

On voit qu'avec le principe de h -coupe, on travaille finalement sur des intervalles en ne considérant que la partie du nombre flou répondant donc à ce degré de confiance h . Il est utile ici d'exposer les différents types des contraintes (proposées par Tanaka et al. [30]) pouvant être prise en compte concernant des intervalles.

On peut donc vouloir que les intervalles observés soient couvert intégralement par ceux prédit, ce qui correspond à un Min-problème. L'inverse est également possible, c'est le Max-problème, et dans ce cas les intervalles observés doivent contenir la totalité de ceux prédit. Il est également possible de considérer un problème dit conjonctif, pour lequel le recouvrement intégral n'est plus obligatoire. Dans ce cas, la contrainte est que l'intersection des deux intervalles ne doit pas être vide. Il est claire que pour notre étude de régressions linéaires, les intervalles pour lesquels l'une ou l'autre de ces contraintes devra être appliquées seront

les sorties, celles observées d'une part et celles estimées d'autre part. Une fois c'est quelques concepts de bases exposés, il est alors possible de se pencher sur les diverses techniques de régressions dites possibilistes, c'est-à-dire aboutissant au final à la résolution d'un problème de programmation linéaire.

I.3.4 Intervalles flous et arithmétique associée

Si l'on considère un intervalle $A = [a^-, a^+]$ et un réel x , et que l'on définit le Midpoint $M(A)$ et le Radius $R(A)$ de cet intervalle par :

$$\begin{aligned} M(A) &= (a^+ + a^-) / 2 \\ R(A) &= (a^+ - a^-) / 2 \end{aligned} \tag{I.7}$$

rappelle aussi que pour deux intervalles a et b , la relation d'inclusion est donnée par :

$$a \subseteq b \Leftrightarrow \begin{cases} b^- \leq a^- \\ a^+ \leq b^+ \end{cases} \Leftrightarrow \begin{cases} M(b) - R(b) \leq M(a) - R(a) \\ M(a) + R(a) \leq M(b) + R(b) \end{cases} \tag{I.8}$$

$$\Leftrightarrow \begin{cases} M(b) - R(b) \leq M(a) - R(a) \\ M(a) - R(b) \leq M(b) - R(a) \end{cases} . \tag{I.9}$$

Ce qui implique :

$$a \subseteq b \Leftrightarrow |M(a) - R(b)| \leq M(b) - R(a) \tag{I.10}$$

Et on a :

$$[A^-, A^+] . x = \begin{cases} [x.A^-, x.A^+] & \text{si } x \geq 0 \\ [x.A^+, x.A^-] & \text{si } x < 0 \end{cases} \tag{I.11}$$

$$\begin{aligned} &= \begin{cases} [x.(M(A) - R(A)), x.(M(A) + R(A))] & \text{si } x \geq 0 \\ [x.(M(A) - R(A)), x.(M(A) - R(A))] & \text{si } x < 0 \end{cases} \\ &= [x.M(A) - |x|.R(A), x.M(A) + |x|.R(A)] \end{aligned} \tag{I.12}$$

I.4 Les fondements de la régression possibiliste

Le modèle possibiliste développé par Tanaka, Uejima et Asia [30] constitue la première tentative remarquable d'introduction de l'imprécision dans les équations de régression. En effet, dans leur article fondateur publié en 1982, les auteurs y implémentent une technique

de résolution des régressions floues dont l'écriture est proche du modèle linéaire traditionnel, puisqu'ils posent :

$$\tilde{Y}^* = \tilde{A}_0 + \tilde{A}_1 X_1 + \tilde{A}_2 X_2 + \dots + \tilde{A}_k X_k \quad (\text{I.13})$$

Et aussi on peut l'écrire :

$$Y = \sum_{i=0}^k A_i X_i \quad (\text{I.14})$$

On appelle Y la sortie ou variables dépendantes, alors que les X_i , $i : (1 \dots k)$ sont les entrées, ou bien encore variables indépendantes. On observe un échantillon de M données, celles ci étant donc de la forme $(y_j, x_{0j}, x_{1j}, \dots, x_{kj})$, $j = 1, \dots, M$. Les A_i , $i = 0, \dots, k$ sont les coefficients, considérés comme étant des nombres flous triangulaires symétriques de valeur modale m_i et de dispersion r_i

Considérons le cas où les entrées x_{ij} , $i = 0, \dots, k$; $j = 1, \dots, M$, sont précises alors que les sorties sont floues, notées dans ce cas (y_j, e_j)

La méthode de base de Tanaka revient à minimiser la somme des dispersions des coefficients à identifier, alors que la contrainte retenue est celle du Min-problème, c'est-à-dire que les intervalles correspondant aux h -coupes des sorties observées (données mesurées) doivent être également inclus dans ceux obtenus par h -coupe des sorties estimées.

L'intervalle obtenue par h -coupe de la $j^{\text{ème}}$ sortie prédite, noté $[Y_j]_h$ est donnée, en fonction des coefficients A_i , par :

$$[Y_j]_h = \sum_{i=0}^{k_{\oplus}} [Y_j]_h x_{ij} = \sum_{i=0}^{k_{\oplus}} [A_i^-, A_i^+]_h x_{ij} \quad (\text{I.15})$$

Où \sum^{\oplus} correspond à l'addition d'intervalles.

Or, on a :

$$[A^-, A^+]_h = [m_i - (1-h)r_i, m_i + (1-h)r_i] \quad (\text{I.16})$$

Avec $Mid([A_i]_h) = m_i$ et $Rad([A_i]_h) = (1-h)r_i$ et avec (I.15) et (I.7) insérées dans (I.13), on obtient :

$$[Y_j]_h = \left[\sum_{i=0}^k x_{ij} m_i - |x_{ij}| (1-h)r_i, \sum_{i=0}^k x_{ij} m_i + |x_{ij}| (1-h)r_i \right] \quad (\text{I.17})$$

Pour respecter les contraintes liées aux Min-problème, on doit avoir $[\hat{Y}_j]_h \subseteq [Y_j]_h$, où $[\hat{Y}_j]_h$ est l'intervalle résultant de h -coupe de la $j^{\text{ème}}$ sortie observée.

On obtient alors comme contraintes à notre programme linéaire :

$$\begin{aligned} \sum_{i=0}^k m_i x_{ij} + (1-h) \sum_{i=0}^k r_i |x_{ij}| &\geq y_j + (1-h)e_j \\ \sum_{i=0}^k m_i x_{ij} - (1-h) \sum_{i=0}^k r_i |x_{ij}| &\leq y_j - (1-h)e_j \end{aligned} \quad (I.18)$$

$$j = 1, \dots, M$$

Dans un premier temps, Tanaka et al. [32] considèrent que l'imprécision du modèle peut être vue comme étant la somme de celle de ses paramètres $A_i, i = 0, \dots, N$. Ceux-ci étant des intervalles flous triangulaires symétriques, leur imprécision est donnée par leur demi-largeur, soit leur Radius. Ainsi, le critère J_1 à optimiser est la somme des Radius des paramètres du modèle, soit :

$$\min_{m_i, r_i} J_1 = \min_{m_i, r_i} \sum_{i=0}^k r_i \quad (I.19)$$

Le principal avantage de cette méthode est quelle est facile à mettre en place car les contraintes considérées découlent directement au Min-problème. De plus, la résolution d'un programme linéaire de ce type est assez aisée.

Pour les inconvénients, on remarqua le grand nombre de contraintes devant être prises en compte. En effet, pour M échantillons de donnée on obtient $2M$ contraintes, ce qui peut être assez problématique si l'on cherche à effectuer une telle régression sur un panel important d'échantillons de mesures.

Aussi, la dispersion totale du modèle est très importante, ce qui ne permet pas de réaliser des prédictions fiables par extrapolation selon Wang et Tsaur [36].

Quelques remarques sur cette méthode ont été apportées par la suite par différentes équipes, en vue de son amélioration [31],[36],.

Tout d'abord, on remarque que beaucoup de coefficients identifiées ont une dispersion nulle ce qui nuit à la qualité du modèle. Pour remédier à cela. Tanaka [31], propose un autre critère à optimiser en lieu et place de celui qui fourni par l'équation (I.19). Il est ainsi plus judicieux d'utiliser un critère prenant en considération non plus la dispersion des coefficients, mais celle des intervalles prédits. Ainsi, (I.19) devient :

$$\min_{m_i, r_i} J_2 = \min_{m_i, r_i} \sum_{j=1}^M \sum_{i=0}^k r_i |x_{ij}| \quad (I.20)$$

On remarqua à ce point que Wang et Tsaur [36] précise que les x_{0j} doivent être fixés à 1 afin d'obtenir un terme constant dans le modèle. Ils ont également suggéré de prendre en compte au sein du critère J_2 de l'équation (I.20) la dispersion des sorties observées, celle-ci

ayant bien entendu une influence sur l'étendue des sorties estimées. Le nouveau critère J_3 est donc le suivant :

$$\min_{m_i, r_i} J_3 = \min_{m_i, r_i} \sum_{j=1}^M \sum_{i=0}^k (r_i |x_{ij}| - e_j) \quad (\text{I.21})$$

Le problème à résoudre est l'optimisation de ce nouveau critère J_3 (I.21), en respectant les contraintes de base exposées en (I.18).

La prédictivité du modèle de Tanaka et al s'en trouve selon eux améliorée, bien qu'aucun résultat chiffré sur un exemple commun n'ait été fourni.

D'autre modification ont également été apportées, non plus au niveau du critère à optimiser, mais sur les contraintes à considérer. En effet, le modèle de base de Tanaka repose sur le principe de Min-problème ce qui fait que les intervalles correspondant aux h -coupes des sorties estimées doivent recouvrir intégralement ceux des sorties collectées. Il est clair que si une des données $(y_j, x_{0j}, x_{1j}, \dots, x_{kj})$, $j = 1, \dots, M$, à une sortie dans la dispersion e_j est très grande, les coefficients estimés verront leur dispersion augmentées fortement pour pallier ce phénomène. On retrouve ce problème ou également si une des données est en fait une valeur singulière (erreur de mesure par exemple), car celle-ci détériorera la précision du modèle de manière forte.

Pour remédier à cela, Tanaka propose de considérer des contraintes issues d'un problème conjonctif. C'est-à-dire de chercher non plus un recouvrement total des h -coupes observées mais à obtenir une intersection non nulle avec celle estimées. Les contraintes exposées en (I.18) deviennent alors :

$$\begin{aligned} \sum_{i=0}^k m_i x_{ij} + (1-h) \sum_{i=0}^k r_i |x_{ij}| &\leq y_j + (1-h)e_j \\ \sum_{i=0}^k m_i x_{ij} - (1-h) \sum_{i=0}^k r_i |x_{ij}| &\geq y_j - (1-h)e_j \end{aligned} \quad (\text{I.22})$$

$$j = 1, \dots, M$$

On voit donc ici que contrairement aux contraintes présentées en (I.18), qui forçaient la borne supérieure (respectivement inférieure) de l'intervalle prédit d'être supérieure (resp. inférieure) à la borne supérieure (resp. inférieure) de l'intervalle observé, les contraintes en (I.22) permettent aux intervalles estimés d'être moins large, car on ne doit plus avoir une intersection non nulle.

Cela permet d'avoir un modèle plus précis, les coefficients le composant ayant une dispersion moins forte.

Cependant certaines limites de cette méthode persistent encore malgré ces différentes modifications. Ainsi seul des nombres flous triangulaires symétriques peuvent être pris en

compte, aucune distinction n'étant faite entre les dispersions à gauche et à droite, qui doivent être identiques. Qui plus est, les méthodes proposées ne concernent que des entrées précises car lorsque l'on cherche des coefficients sous forme de nombres flous triangulaires symétriques il faudrait alors pouvoir multiplier deux nombres de ce type (cf. équation (I.14)) ce qui n'est pas facilement réalisable.

I.5 Présentation des contraintes

Sachant que les sorties observées Y_j sont des intervalles flous triangulaires symétriques et les sorties du modèles sont supposées trapézoïdales, le problème d'inclusion des intervalles flous peut se réduire à une inclusion des noyaux et des supports. Ainsi, les contraintes considérées sont les suivantes :

– *Inclusion des noyaux :*

Le noyau de chaque sortie observée doit être à l'intérieur du noyau du modèle prédit.

– *Inclusion des supports/*

Le support de chaque sortie observée doit être à l'intérieur du support du modèle.

– *Inclusion du noyau dans le support/*

Afin d'assurer que la sortie du modèle est toujours représentable par un intervalle flou trapézoïdale, il faut garantir l'inclusion des noyaux des sorties prédites dans leurs supports.

– *Positivité des radius :*

Bien évidemment, les radius des intervalles flous identifiés doivent être positifs ou nuls.

Ainsi, le nombres de contraintes à considérer pour identifier un modèle régressif à N entrées sur un panel de M données observées en respectant l'inclusion est $6M + 2N$.

I.6 Régression floue et la programmation quadratique (QP)

I.6.1 *Le critère de base de Tanaka*

Dans cette partie, nous introduisons une formule de base de la programmation quadratique, dans l'intervalle correspondant à l'analyse de régression de notre ancienne programmation linéaire basée sur des études de régression.

Une approche de programmation quadratique [31] est un problème d'optimisation, qui consiste à minimiser une fonction objectif quadratique sous réserve de contraintes linéaires.

Pour formuler des intervalles de régression par la programmation quadratique, les éléments suivants devraient être pris en charge :

- les entrées-sorties sont données $:(X_i, Y_j) = (1, x_{1j}, \dots, x_{kj}, y_j), j = 1, \dots, M$.
- les données peuvent être représentées par le modèle linéaire (I.14).
- la sortie donnée Y_j devrait être inclus dans l'intervalle de la sortie estimée $Y(x_j)$, qui est satisfait :

$$Y_j \in Y(x_j), \quad j = 1, \dots, M \quad (\text{I.23})$$

- la fonction objective est définie comme suit :

$$J_4 = \sum_{j=1}^M (r_j |x_j|)^2 = r^t \left(\sum_{j=1}^M (|x_j| |x_j|^t) \right) r \quad (\text{I.24})$$

Qui est la somme des carrées des demi-amplitudes floues (dispersions) des sorties estimées.

Et la somme $\sum_{j=1}^M (|x_j| |x_j|^t)$ est une matrice carrée symétrique $(M + 1) \times (M + 1)$, définie positive .

L'hypothèse (I.23) peut être réécrite comme :

$$Y_j \in Y(x_j) \Leftrightarrow \begin{cases} m^t x_j - r^t |x_j| \leq y_j \\ y_j \leq m^t x_j + r^t |x_j| \end{cases} \quad (\text{I.25})$$

D'après les hypothèses ci-dessus, la problématique de la régression sur intervalle par la programmation quadratique est de déterminer l'intervalle optimal des coefficients $A_i = (m_i, r_i)$, $i = 0, \dots, k$, qui minimise la fonction objectif J dans (I.24) sous les contraintes (I.25). Ainsi la formulation de base est exprimée comme le problème de programmation quadratique suivant :

$$\begin{aligned} \min_{m_i, r_i} \quad & r^t \left(\sum_{j=1}^M (|x_j| |x_j|^t) \right) r + \xi m^t m \\ \text{sujet à :} \quad & m^t x_j - r^t |x_j| \leq y_j \\ & m^t x_j + r^t |x_j| \geq y_j, \quad j = 1, \dots, M \\ & r_i \geq 0, \quad i = 0, \dots, k \end{aligned} \quad (\text{I.26})$$

Ou ξ est un très petit nombre positif, on se trouve ici qu'il est nécessaire d'ajouter le terme $\xi m^t m$ à (I.24), pour que (I.26) devient un problème de programmation quadratique en ce qui concerne les variables de décision m et r . Le terme $\xi m^t m$ est négligeable devant la fonction objectif, le problème de la programmation quadratique (I.26) [31] correspond au problème linéaire dans (I.20).

Il devrait être noté que la fonction objective quadratique donne plusieurs demi-amplitudes floues (dispersions) que la fonction objective linéaire, on peut aussi dire que l'estimation par

la programmation quadratique donne plus de vecteur des demi-amplitudes non crisper (non nettes) $r = (r_0, \dots, r_k)^t$, que l'estimation par la programmation linéaire.

I.6.2 Le deuxième critère de Tanaka

Ici Tanaka propose un problème de programmation quadratique intégrant la tendance centrale des moindres carrés et le modèle possibiliste de la régression floue [32]. Dans cette méthode on minimise la somme des carrés de distances entre les centres de sortie estimée et les sorties observées.

$$\sum_{j=1}^M (y_j - m^t x_j)^2 \quad (\text{I.27})$$

correspondant aux principes de moindres carrés. Ainsi on aboutit à une nouvelle fonction objective à partir de (I.24) et (I.27) :

$$J_5 = k_1 \sum_{j=1}^M (y_j - m^t x_j)^2 + k_2 \sum_{j=1}^M r^t |x_j| |x_j|^t r \quad (\text{I.28})$$

$$\Rightarrow J_5 = k_1 \left(m^t \left(\sum_{j=1}^M x_j x_j^t \right) m - 2 \sum_{j=1}^M y_j x_j^t m + \sum_{j=1}^M y_j^2 \right) + k_2 \sum_{j=1}^M r^t |x_j| |x_j|^t r \quad (\text{I.29})$$

Ou k_1 et k_2 sont les coefficients de poids. L'objectif de l'utilisation de ce nouvel critère est de déterminer l'intervalle optimal des coefficients $A_i = (m_i, r_i)$, $i = 0, \dots, k$, qui minimise la fonction objective J dans (I.29) sous les mêmes contraintes que dans (I.25). Le problème de la programmation quadratique s'écrit alors comme suit :

$$\begin{aligned} \min_{m_i, r_i} \quad & k_1 \sum_{j=1}^M (y_j - m^t x_j)^2 + k_2 \sum_{j=1}^M r^t |x_j| |x_j|^t r \\ \text{subject à :} \quad & m^t x_j - r^t |x_j| \leq y_j \\ & m^t x_j + r^t |x_j| \geq y_j, \quad j = 1, \dots, M \\ & r_i \geq 0, \quad i = 0, \dots, k \end{aligned} \quad (\text{I.30})$$

Les coefficients de poids k_1 et k_2 jouent un rôle important dans la formulation des modèles de régression floue. Par exemple, si on prend une valeur pour k_1 plus grande que k_2 , la ligne centrale de la régression obtenue ici est tend vers la ligne obtenue par la méthode de moindres carrés. Au contraire, si on prend une grande valeur de k_2 par à port à k_1 , maintenant on concentre sur réduire le manque de netteté du modèle, et (I.30) devient tout à fait semblable à (I.26). Ainsi, un analyste peut avec souplesse assigner les coefficients de poids k_1 et k_2 , en considérant la différence entre $\sum_{j=1}^M (y_j - m^t x_j)^2$ et $\sum_{j=1}^M r^t |x_j| |x_j|^t r$.

I.6.3 La distance de Diamond

Toutefois, malgré l'intérêt indéniable de la technique de résolution suggérée par Savic et al.,[28] c'est véritablement à Diamond que nous devons la première vraie méthode de régression floue par les moindres carrés dans laquelle nous retrouvons l'objectif de minimisation du carré des écarts entre valeurs observées et valeurs estimées par le modèle (Diamond, 1988,[11]). Or, renouer avec cette technique d'ajustement exigeait au préalable de définir de façon opérationnelle la notion de distance entre les nombres flous puisque les déviations sont considérées entre NFTs. Dans ce cas, Diamond a formulé une "L₂-métrique" entre deux quantités floues (Idri.[19]) qui n'est rien moins qu'une distance euclidienne appliquée aux nombres flous triangulaires mais pas nécessairement symétriques (les nombres $A(m, r_L, r_R)_{RL}$,sont présentés dans la page 15 "Les nombres flous").

La distance de Diamond s'écrit comme suit :

$$d(\langle m_1, r_{L1}, r_{R1} \rangle, \langle m_2, r_{L2}, r_{R2} \rangle)^2 = (m_1 - m_2)^2 + [(m_2 - r_{L2}) - (m_1 + r_{L1})]^2 + [(m_1 + r_{R1}) - (m_2 + r_{R2})]^2 \quad (I.31)$$

Une fois munis d'une telle métrique, il devient désormais possible de recourir à une estimation des paramètres flous de l'équation de régression en minimisant le carré des déviations constatées entre observations et résultantes du modèle. C'est précisément ce que Diamond se propose de réaliser lorsqu'il présente un modèle visant à rendre minimale la somme des L₂métriques entre NFT expérimentaux et estimés (Idri.[19]). Plus généralement, le problème d'optimisation, formulé par l'auteur, peut alors s'écrire par analogie avec la régression statistique classique.

La fonction objective à minimiser :

$$\min \sum_{j=1}^M d\left(\tilde{A}_0 + \tilde{A}_1 x_{1j} + \dots + \tilde{A}_k x_{kj}, \tilde{Y}_j\right)^2 \quad (I.32)$$

Où, $\tilde{A}_i = (K_{\tilde{A}_i}, S_{\tilde{A}_i})$, $i = 0, \dots, k$.

Au total, dans le modèle de Diamond, et à l'instar de l'évaluation du vecteur des centres des paramètres du modèle de régression $(m_0^*, m_1^*, \dots, m_k^*)$, l'estimation du vecteur des amplitudes des paramètres $(r_0^*, r_1^*, \dots, r_k^*)$, s'obtient, elle aussi, par les moindres carrés mais appliqués, cette fois, aux écarts (ξ_i) constatés sur les variables observées. En ce sens, ce modèle permet d'obtenir des paramètres dont la justification statistique est directe puisque, à notre avis, il est possible de considérer que les amplitudes obtenues de la sorte constituent

des estimateurs sans biais ayant les mêmes qualités que les valeurs modales estimées par les moindres carrés ordinaires.

Ainsi, la résolution du problème en considérant l'inclusion revient à résoudre un problème d'optimisation quadratique avec des contraintes linéaires.

I.7 Exemple illustratif

L'objectif de cette section est d'illustrer sur un exemple simple les différentes techniques linéaire et quadratique de la régression floue.

I.7.1 Les indicateurs de performance

Pour comparer différents modèles flous, il faut introduire des indicateurs de performance permettant d'évaluer la qualité des modèles identifiés.

L'objectif essentiel de la régression floue est la minimisation de l'imprécision de la sortie du modèle, qui devra donc être considérée comme une grandeur importante pour la comparaison de modèles linéaires flous. Pour ce faire, l'indicateur retenu est l'expression de l'imprécision de la sortie du modèle défini au préalable comme critère d'optimisation. Sachant que l'on cherche à caractériser aussi bien des modèles flous triangulaires que trapézoïdaux, il faut considérer l'indicateur le plus général, c'est-à-dire J_3 (équation (I.21)).

Nous considérons l'indicateur *volume* défini dans [1].

L'aire de la sortie trapézoïdale prédite pour l'entrée x_j est donnée dans l'espace des bornes par :

$$aire(\widehat{Y}(x_j)) = \frac{K_{\widehat{Y}_j}^+ + S_{\widehat{Y}_j}^+}{2} - \frac{K_{\widehat{Y}_j}^- + S_{\widehat{Y}_j}^-}{2} \quad (I.33)$$

Il est également possible de l'exprimer dans l'espace Midpoint / Radius par :

$$aire(\widehat{Y}(x_j)) = r_{K_{\widehat{Y}_j}} + r_{S_{\widehat{Y}_j}} \quad (I.34)$$

Le *volume* de la sortie trapézoïdale prédite pour l'entrée x_j est donnée dans l'espace des bornes par :

$$volume = \int_{x_{\min}}^{x_{\max}} aire(\widehat{Y}(x)) dx \quad (I.35)$$

Soit, selon l'expression (I.34) :

$$volume = \int_{x_{\min}}^{x_{\max}} (r_{K_{\widehat{Y}_j}} + r_{S_{\widehat{Y}_j}}) dx \quad (I.36)$$

c'est-à-dire :

$$volume = \int_{x_{\min}}^{x_{\max}} (r_{K_{A_0}} + r_{S_{A_0}}) + \sum_{i=1}^N (r_{K_{A_i}} + r_{S_{A_i}}) x_i \cdot signe(x_i) dx \quad (I.37)$$

Or, le signe de la variable x_j est constant sur le domaine décalé $[x_{\min}, x_{\max}]$, car les deux valeurs de décalage admissibles imposent le zéro sur une des bornes du domaine de définition initial. Par conséquent, il est possible d'écrire :

$$volume = (r_{K_{A_0}} + r_{S_{A_0}}) + \sum_{i=1}^N (r_{K_{A_i}} + r_{S_{A_i}}) \cdot \left(\frac{x_{\max_i} + x_{\min_i}}{2} \right) \cdot \Delta_i \quad (I.38)$$

Où :

$$\Delta_i = signe \left(\frac{x_{\max_i} + x_{\min_i}}{2} \right) \quad (I.39)$$

avec N représentant le degré du modèle identifié.

l'équation (I.38) représente la formule de volume d'un modèle flou trapézoïdal.

Le deuxième indicateur est l'erreur quadratique entre les valeurs modales des observations et des prédictions. Dans le cas de l'approche triangulaire, elle est définie par :

$$err_quad = \sum_{j=1}^M (K_{Y_j} - K_{\hat{Y}_j})^2 \quad (I.40)$$

avec, $K_{Y_j}, K_{\hat{Y}_j}$ sont les midpoints de la sortie observée et prédite respectivement.

Maintenant dans le cas de l'approche trapézoïdale l'erreur est définie en ne conservent que les midpoints des intervalles noyaux ($m_{K_{\hat{Y}_j}}$) :

$$err_quad = \sum_{j=1}^M (K_{Y_j} - m_{K_{\hat{Y}_j}})^2 \quad (I.41)$$

I.7.2 Les données et les modèles identifiés

Le jeu de données utilisé ici est simple et fréquemment rencontré dans la littérature [31], et présenté dans le tableau I.1. Les données sont composées d'une entrée précise (crispe) et d'une sortie floue observée sous la forme d'un nombre flou triangulaire symétrique (NFTS), celui-ci est défini par la valeur de son noyau (Midpoint), et son intervalle de support (Radius). Dans le cas de notre étude, l'objectif de la technique de régression floue est d'identifier un modèle d'imprécision (demie amplitude flou) minimale englobant l'ensemble des observations.

j	x_j	Y_j
1	0.1	(2.25,0.75)
2	0.2	(2.875,0.875)
3	0.3	(2.5,1.0)
4	0.4	(4.25,1.75)
5	0.5	(4.0,1.5)
6	0.6	(5.25,1.25)
7	0.7	(7.5,2.0)
8	0.8	(8.5,1.5)

TAB. I.1 – Le jeu de données observées

Les modèles triangulaires et trapézoïdaux sont identifiés en minimisant les critères J_3 , J_4 et de Diamond, sous l'ensemble des contraintes adéquates. L'identification du modèle triangulaire se fait sur les h -coupes de niveau $h = 0$ des sorties observées.

– Premièrement le modèle de premier ordre à identifier est de la forme :

$$\widehat{Y}(x) = A_0 \oplus A_1 \cdot x \quad (\text{I.42})$$

On applique sur ce modèle, la technique linéaire 'LP', et les deux méthodes quadratiques 'QP' de Diamond et de Tanaka.

	LP triangulaire	QP Tanaka	QP Diamond
A_0	(0.9583, [0, 1.9167])	(0.6079, [-0.7008, 1.9167])	(0.5786, [-0.7595, 1.9167])
A_1	(7.9167, [5, 10.8333])	(8.6175, [6.4016, 10.8333])	(8.6761, [6.5189, 10.8333])
<i>volume</i>	18.1667	18.4470	18.4705
<i>err_quad</i>	4.5816	4.2779	4.2721

TAB. I.2 – Paramètres des modèles triangulaire obtenus et indicateur de performance

	LP trapez	QP trapez Diamond
A_0	([0.25, 1.3571], [0, 1.9167])	([0.0369, 1.3571], [-0.7386, 1.9167])
A_1	([7.5, 8.9286], [5, 10.8333])	([7.9261, 8.9286], [6.4773, 10.8333])
<i>volume</i>	25.1667	25.5473
<i>err_quad</i>	4.4129	4.3317

TAB. I.3 – Paramètres des modèles trapézoïdaux obtenus et indicateur de performance

Les paramètres obtenus pour chacun des modèles sont présentés dans les tableaux (I.2, et I.3) , selon qu'une représentation dans l'espace des bornes (cf. l'équation (I.2)) est adoptée. Une représentation des modèles flous triangulaires et trapézoïdaux identifiés est illustrée dans les figures I.2 et I.3 respectivement. Ces figures montrent les sorties observées floues (les midpoint ; cercles rouges, et les radius ; petites lignes verticales bleues) ainsi que les modèles identifiés (le support ; traits pleins, et le noyau ; traits discontinus).

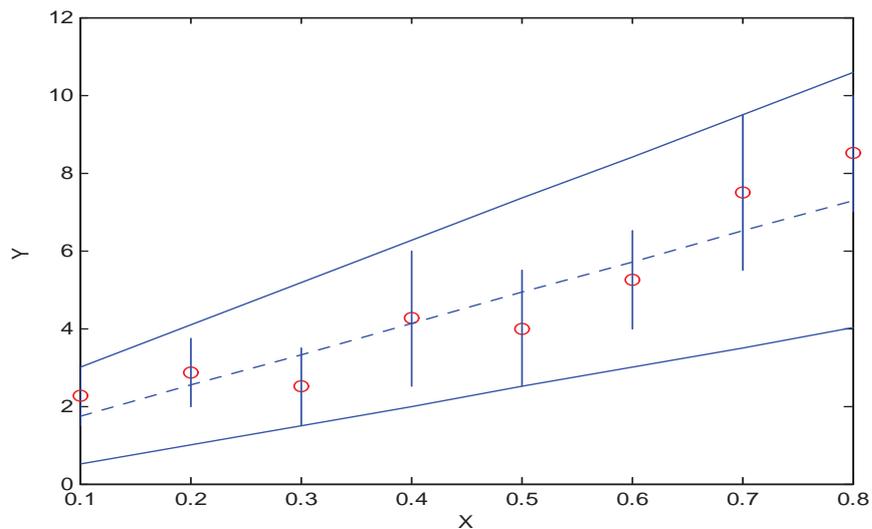


FIG. I.2 – Modèle flou triangulaire identifié

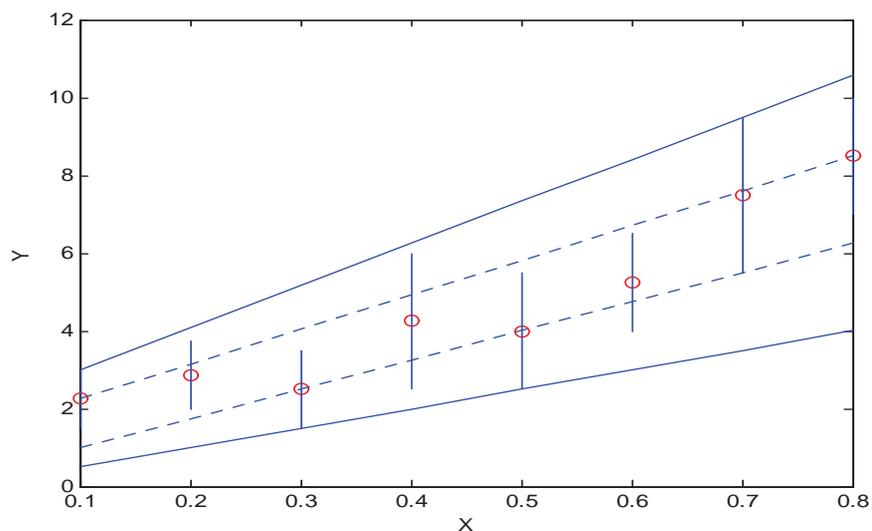


FIG. I.3 – Modèle flou trapézoïdale identifié

	LP triangulaire ($h = 0.8$)	LP triangulaire ($h = 0.9$)
A_0	(0.8417, [-2.3667, 4.05])	(0.8232, [-5.1589, 6.8054])
A_1	(8.0833, [2.6667, 13.5])	(8.1429, [-0.7857, 17.0714])
<i>volume</i>	45.1667	80

TAB. I.4 – Paramètres des modèles triangulaire identifié et indicateurs de performance associés

On constate l'égalité des supports pour les deux types des modèles identifiés, triangulaire (LP et QP) et trapézoïdaux (LP et QP), (cf. tableaux I.2 et I.3) ainsi que l'inclusion des observations dans les prédictions au niveau des supports (garantie par contraintes dans les deux cas).

La différence notable entre les différents modèles se situe au niveau des noyaux. Dans le cas du modèle triangulaire (cf. figure I.2), le noyau est précis. Qui, ne permet bien évidemment pas l'inclusion. A contrario, le noyau de la sortie du modèle trapézoïdal (cf. figure I.3) est un intervalle, respectant l'inclusion de tous les noyaux des observations.

Pour une comparaison réaliste en termes d'imprécision des modèles triangulaire et trapézoïdaux, il est donc nécessaire de pouvoir garantir un niveau minimum d'inclusion similaire pour les deux types de modèles. Malheureusement, il n'est généralement pas possible de garantir une inclusion au niveau $h = 1$ avec un modèle triangulaire, comme c'est le cas avec un modèle trapézoïdale. Par contre, le choix d'un h élevé (mais inférieur strictement à 1) dans le problème minimal garantit une inclusion à ce niveau h . Pour augmenter le niveau d'inclusion garanti du modèle triangulaire, celui-ci est maintenant identifié avec des contraintes d'inclusion sur les h -coupes au niveau $h = 0.9$, et non plus au niveau $h = 0$ considéré précédemment. Les paramètres du modèle obtenu (représentation dans l'espace des bornes), ainsi que les indicateurs de performance qui lui sont associés sont disponibles dans le tableau I.4.

- Deuxièmement, on choisit le modèle du deuxième ordre. Le modèle à identifier est de la forme :

$$\widehat{Y}(x) = A_0 \oplus A_1.x \oplus A_2.x^2 \tag{I.43}$$

On applique ici les techniques quadratiques sur le même jeux de données (tableau I.1).

Les paramètres identifiés sont représentés par l'espace des bornes dans le tableau I.5.

	QP triangulaire	QP trapézoïdal
A_0	(2.4733, [1.0717, 3.8749])	([1.8399, 2.8844] , [1.1403, 4.3011])
A_1	(0, [-1.6819, 1.6819])	([-0.9793, -0.8257] , [-2.5801, 0.0076])
A_2	9.077	10.5991
err_quad	2.0048	1.4934
$volume$	17.2677	21.7559

TAB. I.5 – Paramètres des modèles obtenus et indicateur de performance

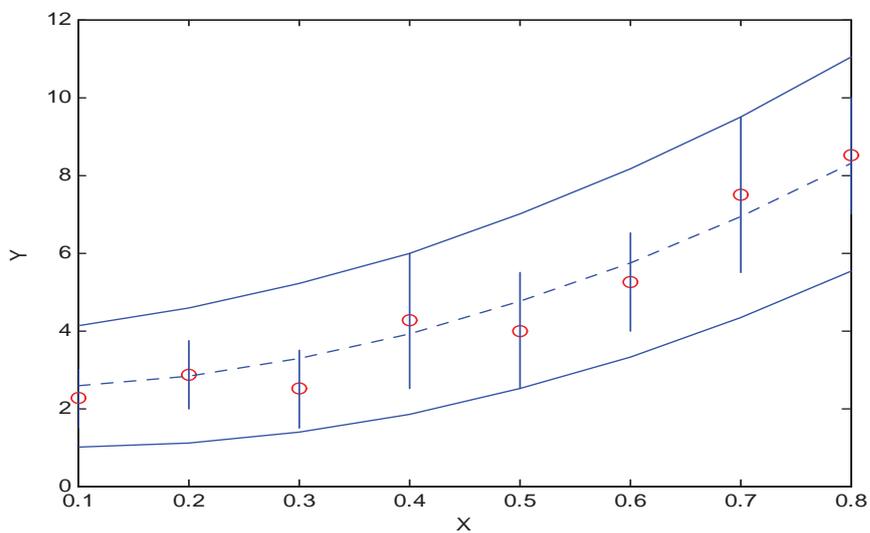


FIG. I.4 – Modèle flou triangulaire identifié (2^{ème} ordre)

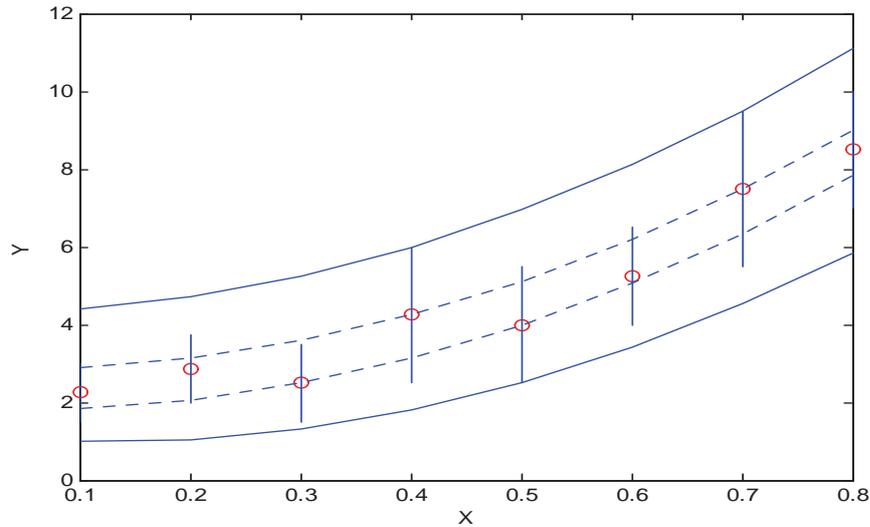


FIG. I.5 – Modèle flou trapézoïdale identifié (2^{ème} ordre)

D’après les résultats du tableau I.5, on peut remarquer qu’il y a un manque de représentation de l’imprécision dans ces modèles identifiés. Par contre l’inclusion est totalement garantie pour les modèles trapézoïdaux.

I.7.3 Un nouveau modèle linéaire

Notre objectif est de trouver un modèle minimal qui englobe toute les données observées. Bissier et al. [1], proposent un nouveau modèle linéaire, ils ont ajouté un paramètre de réglage (*shift*) au modèle identifié de façon à pouvoir ajuster le signe du coefficient pondérant le paramètre A_i pour $i = 1, N$, quel que soit le signe de la variable d’entrée. Pour ce faire, il est proposé à l’utilisateur de chercher à identifier un modèle de la forme :

$$\hat{Y}(x) = A_0 \oplus \sum_{i=1}^N A_i \cdot (x^i - shift^i) \tag{I.44}$$

Le paramètre *shift* étant un scalaire permettant de modifier à volonté le signe de la pondération de A_i .

Il est important ici de rappeler que le signe de la variable d’entrée originelle x du modèle est conditionné par les observations. Dans un premier temps, le jeu de données considéré est celui introduit dans [2] et présenté dans le tableau (I.6), sont précises, tandis que les sorties sont données sous forme d’intervalles flous triangulaires symétriques, définis par $Y_j = (m_{Y_j}, r_{Y_j})$.

j	x_j	Y_j	j	x_j	Y_j
1	1	(9.000, 5.000)	16	16	(9.500, 3.500)
2	2	(6.500, 4.500)	17	17	(13.000, 7.000)
3	3	(9.500, 6.500)	18	18	(16.000, 6.000)
4	4	(10.000, 9.000)	19	19	(20.500, 6.500)
5	5	(12.500, 8.500)	20	20	(29.000, 8.000)
6	6	(20.000, 6.000)	21	21	(29.500, 8.500)
7	7	(18.500, 6.500)	22	22	(31.500, 5.500)
8	8	(21.000, 7.000)	23	23	(35.500, 2.500)
9	9	(26.000, 6.000)	24	24	(45.000, 4.000)
10	10	(28.000, 7.000)	25	25	(45.000, 6.000)
11	11	(25.500, 6.500)	26	26	(42.500, 4.500)
12	12	(26.000, 5.000)	27	27	(50.000, 6.000)
13	13	(5.000, 3.000)	28	28	(48.500, 4.500)
14	14	(7.000, 4.000)	29	29	(49.000, 5.000)
15	15	(9.500, 7.500)	30	30	(48.500, 4.500)

TAB. I.6 – Le jeu de données observées

Le domaine de variation de l'entrée du modèle est définie comme étant l'intervalle $D = [1, 30]$. Si l'on considère les $k = 5$ premières et dernières données, il est possible d'obtenir l'imprécision moyenne des sorties initiale R_{init} et finale R_{fin} . Les valeurs obtenues sont $R_{init} = 6.7$ et $R_{fin} = 5.08$, ce qui implique donc $R_{init} > R_{fin}$. Par conséquent, l'imprécision des sorties sur le domaine est globalement croissante, la valeur de décalage appropriée est donc fixée à un $shift = 30$.

L'ordre adéquat du modèle le mieux placé pour représenter la relation entre les entrées et la sortie de ce jeu de données est inconnu. Par conséquent, des modèles d'ordre allant de $N = 1$ à $N = 5$ sont identifiés successivement. Chacun d'entre eux est représenté sur les figures (I.6, I.7, I.8, I.9, I.10). Les valeurs optimales du critère d'identification (*volume*) sont regroupées dans le tableau I.7.

Le modèle finalement retenu est donc d'ordre 4, la valeur de décalage ayant été fixée à $shift = 30$. Ses paramètres ainsi que la valeur du critère (*volume*) sont synthétisés dans le tableau I.8

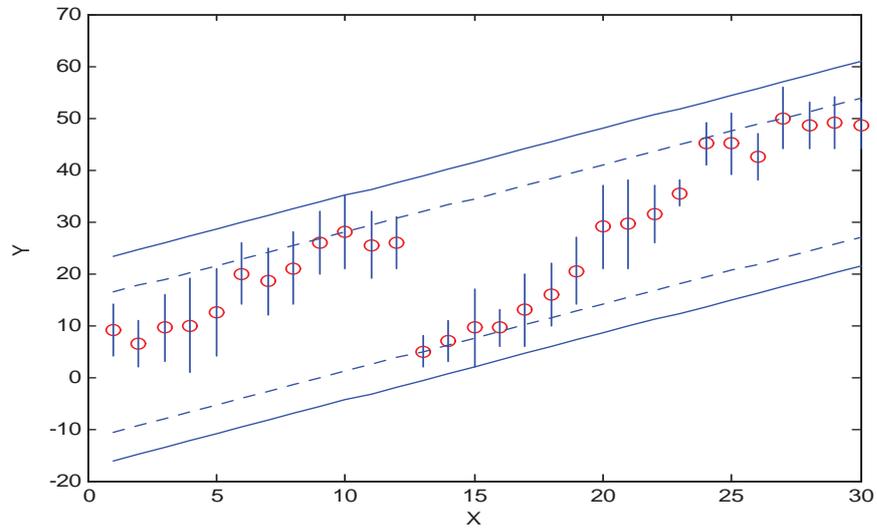


FIG. I.6 – Modèle identifié d'ordre 1

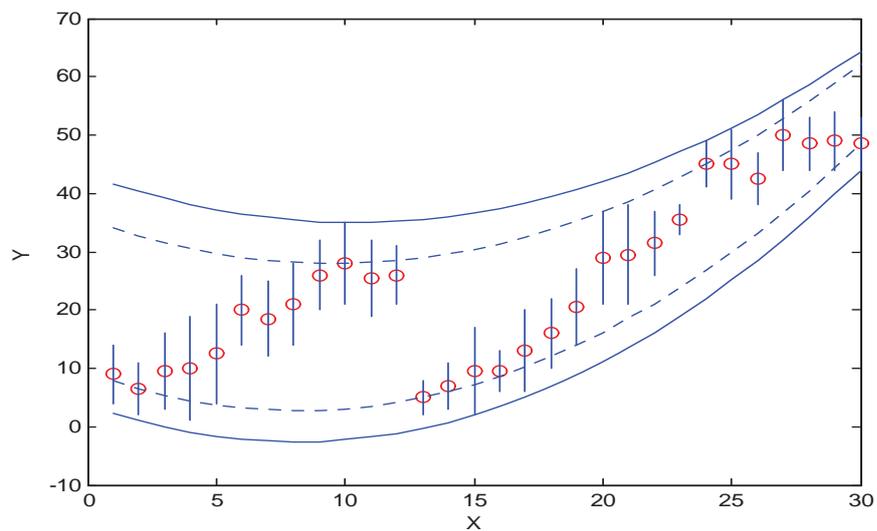


FIG. I.7 – Mdèle identifié d'ordre 2

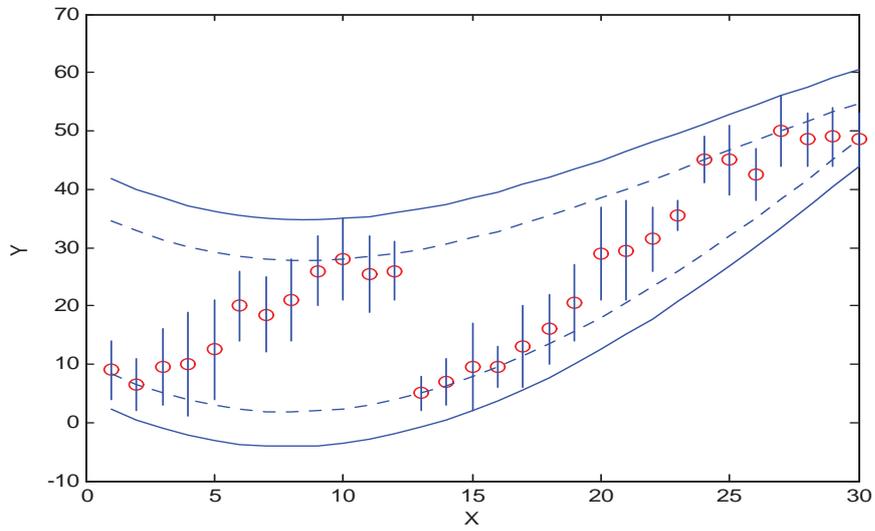


FIG. I.8 – Modèle identifié d'ordre 3

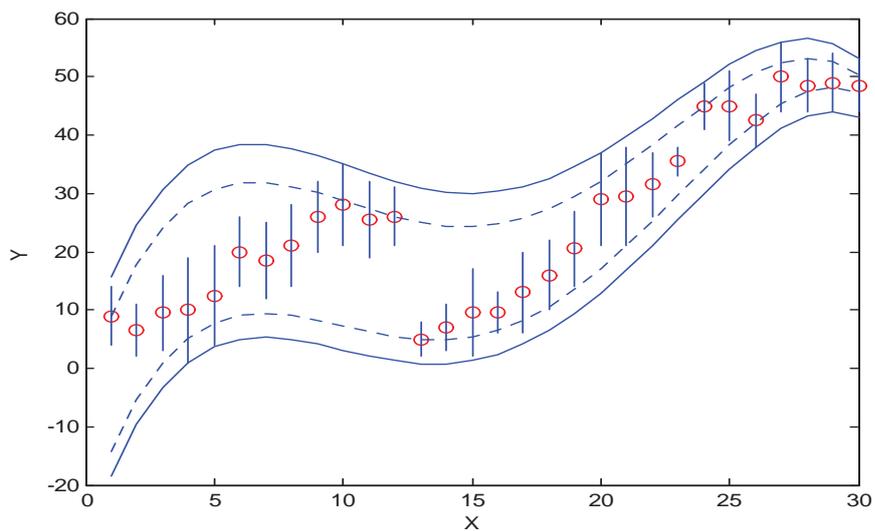


FIG. I.9 – Modèle identifié d'ordre 4

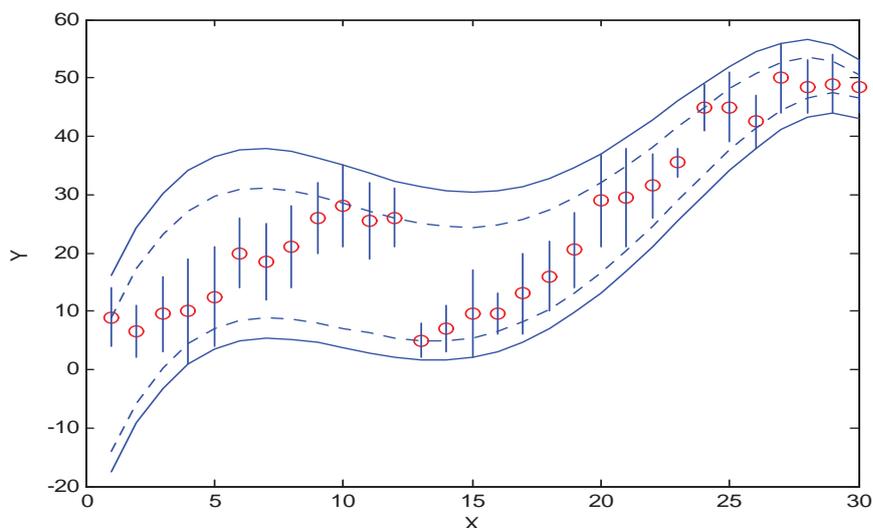


FIG. I.10 – Modèle identifié d’ordre 5

Ordre N du modèle	<i>Volume</i>
$N = 1$	33.1765
$N = 2$	24.8958
$N = 3$	22.1417
$N = 4$	17.5879
$N = 5$	17.6411

TAB. I.7 – Valeur du critère d’optimisation selon l’ordre N de modèles identifiés

A titre d’information, le modèle d’ordre optimal $N = 4$ mais de valeur de décalage non appropriée, c’est-à-dire $shift = 1$ est également présenté dans le tableau I.8, tandis qu’une représentation en est fournie sur la figure I.11.

Bien que l’ordre soit optimal, le modèle obtenu n’offre pas une représentation plus efficace des données. Il est donc clair que la combinaison d’un modèle linéaire d’ordre adéquat déterminé à l’aide d’un choix de valeur de décalage des entrées pertinent permet d’obtenir un modèle performant [2].

	Modèle pour $shift = 30$	Modèle pour $shift = 1$
A_0	$([47.2948, 50.2132], [43.1132, 53])$	$([-11.8172, 9], [-14.8172, 16])$
A_1	13.3424	10.1792
A_2	$([-1.7052, -1.6877], [-1.7096, -1.6877])$	-1.1888
A_3	0.0808	0.0523
A_4	-0.0012	-0.00071
$Volume$	17.5879	25.8172

TAB. I.8 – Paramètres des modèles d'ordre 4 obtenus pour $shift = 30$ et $shift = 1$

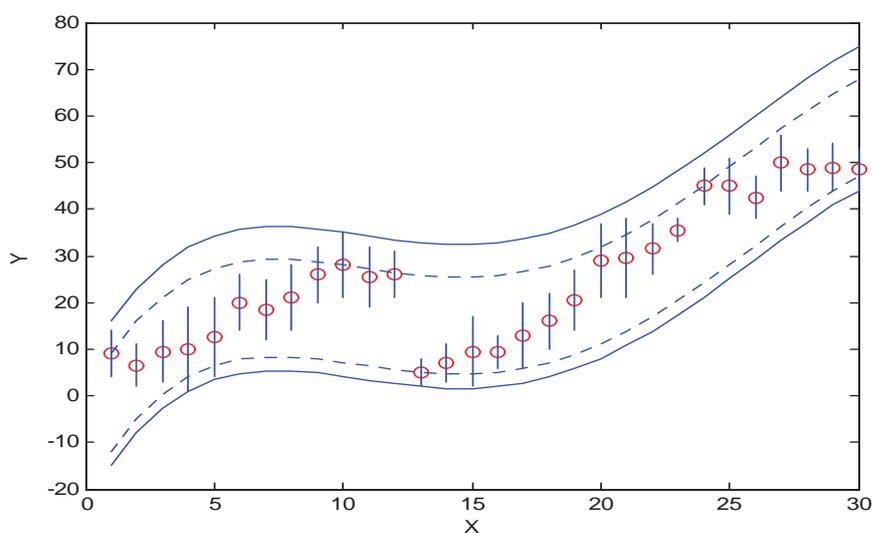


FIG. I.11 – Représentation du modèle identifié d'ordre 4 pour une valeur erronée de décalage

I.8 Conclusion

En résumé, quelques méthodes conventionnelles d'estimation de paramètres dans les modèles de régression floue ont été illustrées tout au long de ce premier chapitre. Et, si toutes les techniques introduisent l'imprécision et l'ambiguïté lors du choix des paramètres flous dans l'équation de régression, les manières de procéder pour déterminer ces nombres flous triangulaires ou trapézoïdaux et décrire la relation fonctionnelle existant entre les entrées et la sortie dans les modèles régressifs sont parfois assez éloignées.

Dans ce chapitre, nous avons traité le problème d'identification (construction) des modèles flous à partir de données entrée-sortie. Nous avons expliqué d'abord les concepts de la logique floue, ensuite nous avons étudié les différentes techniques linéaire et quadratique de régression floue, en nous focalisant particulièrement sur les modèles de Tanaka [31], et de Diamond [11], ainsi que le nouveau modèle de Bissierier [1].

Dans la suite de ce travail, nous nous chercherons un modèle libre c'est-à-dire un modèle indépendant des données observées, ce dernier doit être minime et garantir l'inclusion.

Chapitre II

La régression floue par les SVMs

II.1 Introduction

Dans ces dernières années les machines à vecteur de support (SVM) sont devenues un outil populaire dans la littérature pour l'apprentissage, la classification et la reconnaissance des formes. De nombreux scientifiques et chercheurs ont travaillé à la fois sur la théorie et l'application des SVM. Elles ont été appliquées avec succès dans des domaines tels que la reconnaissance des caractères manuscrits, la catégorisation de texte, la vision par ordinateur, et la bioinformatique. Et l'algorithme lui-même a été considérablement amélioré par rapport à l'idée originale de Vapnik et al [35].

La première SVM est un hyperplan classificateur qui sépare les données d'apprentissage dans un espace d'entrée à n dimensions selon un critère séparé. Différents critères mèneront à une certaine variation dans les algorithmes de SVM. Un cas particulier est celui où le critère est choisi pour être une fonction de perte, l'analyse peut être étendue à la régression. Dans ce chapitre nous allons discuter l'utilisation des SVM pour la régression.

Plus tard, les fonctions du noyau (Kernel) seront introduites pour reproduire des espaces caractéristiques dans lesquels un classificateur hyperplan optimal est recherché. Il s'avère que l'algorithme de machines à vecteurs de support du noyau est formellement semblable à l'original. Selon le théorème de Mercer [34], une fonction doit satisfaire plusieurs propriétés pour être une fonction noyau. Chaque fonction noyau détermine implicitement un espace de caractéristiques avec une dimension plus élevée.

L'implémentation des SVM est un problème de programmation quadratique, pour laquelle beaucoup d'algorithmes sont disponibles, telles que l'optimisation séquentiel minimale (SMO), on utilise dans notre travail les fonctions prédéfinis dans le toolbox de MATLAB. La résolution de ce problème de SVM donne uniquement un peu de paramètres, ce qui indique que juste une partie des données d'apprentissages est suffisante pour déterminer le classificateur optimal. Ces points sont appelés vecteurs de support.

Dans ce chapitre, nous présenterons la régression simple à vecteur de support (SVM), en suite la régression floue par les SVM dans un environnement imprécis, avec quelques exemples illustratifs pour l'évaluer.

Finalement, nous proposerons une nouvelle technique de régression floue plus robuste.

II.2 Les SVMs

II.2.1 Principe des SVMs

Plaçons nous dans le cas de deux classes linéairement séparables. Il existe une infinité d'hyperplan capable de séparer parfaitement ces deux classes. Pour toutes les formes x_i de classe u_i de la base d'apprentissage, on a :

$$\begin{cases} w^t x_i + w_0 > 0 \text{ si } u_i = 1 \\ w^t x_i + w_0 < 0 \text{ si } u_i = -1 \end{cases} \quad (\text{II.1})$$

Le principe des SVM est de choisir celui qui va maximiser la marge entre l'hyperplan et les points de la base d'apprentissage.

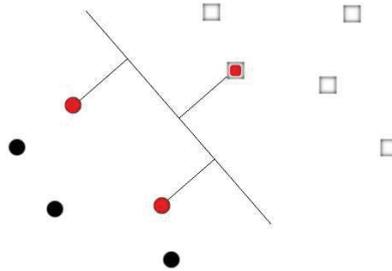


FIG. II.1 – Hyperplan avec 3 vecteurs de support (les vecteurs de support sont en rouge). Leur position maximise la distance entre ces points et leur projeté sur l'hyperplan.[20]

Pour un hyperplan d'équation $h(x) = w^t x_i + w_0$, la distance d'un point x a l'hyperplan est $\frac{h(x)}{\|w\|}$. Si l'on veut obtenir la plus grande marge possible entre l'hyperplan et les vecteurs de support, il faut donc minimiser $\frac{1}{2} \|w\|^2$, sous les contraintes $u_i(w^t x_i + w_0) > 1$ pour tout $i \in 1, \dots, m$ (Bon classement des points de la base d'apprentissage).

D'après la théorie de l'optimisation, et comme l'objectif (minimiser $\frac{1}{2} \|w\|^2$) et les contraintes ($u_i(w^t x_i + w_0) > 1$) sont strictement convexes, ce problème peut se poser sous la forme d'un Lagrangien :

$$L(w; w_0, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^m \alpha_i (u_i \cdot (x_i, w + w_0) - 1) \quad (\text{II.2})$$

dont il faut annuler les dérivées partielles par rapport à w et w_0 . (les α_i étant les multiplicateurs de Lagrange). Dans cette expression, appelée <expression duale>, on voit que

les contraintes de bon classement sont présentes sous la forme de pénalités sur le critère de généralisation (maximisation de la marge).

Le théorème de Kuhn-Tucker prouve que notre problème de minimisation sous contraintes est équivalent aux solutions des équations annulant les dérivées du Lagrangien par rapport à nos variables w, w_0 ,

L'annulation de ces dérivées partielles donne :

$$\begin{cases} Max_{\alpha} \left\{ \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j u_i u_j(x_i, x_j) \right\} \\ \sum_{i=1}^m \alpha_i u_i = 0 \end{cases} \quad (\text{II.3})$$

ainsi que w :

$$w = \sum_{i=1}^m \alpha_i u_i x_i \quad (\text{II.4})$$

Les multiplicateurs de Lagrange qui vérifient (II.3) correspondent aux vecteurs de support : seuls les points sur la marge nous intéressent. Tous les autres points ont un α nul.

Finalement, l'équation de l'hyperplan séparateur est :

$$h(x) = (w^*, x) + w_0 = \sum_{i=1}^m \alpha_i u_i(x_i, x_i) + w_0^* \quad (\text{II.5})$$

où les α_i^* sont les solutions de (II.3) ; et w_0 est trouvé en plaçant les coordonnées d'un « vecteurs de support » dans (II.1) (puisque les vecteurs de support vérifient la contrainte $u_i(w^t x_i + w_0) > 1$). w est simplement une combinaison des vecteurs de support (II.4).

La décision se fait donc très simplement, comme pour un discriminateur linéaire classique. L'ingéniosité de la méthode réside simplement dans le choix judicieux des α , qui constitue l'apprentissage de cette méthode.

Trouver les α revient à choisir les vecteurs de support. On trouve les α en résolvant les équations (II.12) et (II.13), ce qui revient à résoudre un problème de programmation quadratique convexe. Dans la pratique, on utilisera les bibliothèques SVM-Light de Joachim [20], ou la méthode SMO implémentée par Platt [26]. On utilise ici la fonction de la programmation quadratique prédéfinie dans MATLAB.

La réalisation d'un classificateur SVM peut donc être faite le plus simplement : les sources citées précédemment permettent de dégager les vecteurs de support de notre base d'apprentissage, reste à faire la décision grâce à II.5.

II.2.2 Cas linéairement non séparable

Pour le cas de classes non linéairement séparables, on relâche la contrainte de bon classement, initialement :

$$u_i(w^t x_i + w_0) > 1 \quad (\text{II.6})$$

devient :

$$u_i(w^t x_i + w_0) > 1 - \xi_i \quad (\text{II.7})$$

et on ne doit plus minimiser :

$$\frac{1}{2} \|w\|^2 \quad (\text{II.8})$$

mais :

$$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad (\text{II.9})$$

avec $C \geq 0$

II.2.3 Introduction des fonctions noyaux

On a présenté dans la partie précédente une méthode de séparation linéaire, un peu limitée. L'introduction des fonctions noyaux va permettre de s'affranchir de cette limitation. On montre que dans l'équation de l'hyperplan séparateur :

$$h(x) = (w^*, x) + w_0^* = \sum_{i=1}^m \alpha_i^* u_i(x_i; x_i) + w_0^* \quad (\text{II.10})$$

le produit scalaire (x, x_i) peut être remplacé par n'importe quelle fonction noyau $K(x; x_i)$, du moment qu'elle effectue également un produit scalaire. L'équation de l'hyperplan séparateur devient :

$$h(x) = (w^*, x) + w_0^* = \sum_{i=1}^m \alpha_i^* u_i K(x, x_i) + w_0^* \quad (\text{II.11})$$

où les α_i^* sont les solutions de :

$$\text{Max}_{\alpha} \left\{ \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j u_i u_j K(x_i, x_j) \right\} \quad (\text{II.12})$$

et

$$\sum_{i=1}^m \alpha_i u_i = 0 \quad (\text{II.13})$$

Voici un éventail des fonctions noyaux couramment utilisées (respectant donc la condition d'effectuer un produit scalaire)

Il existe plusieurs types de noyau :

- noyau Polynomial : $K(x, y) = (x^T \cdot y + 1)^d$.
 - noyau de la fonction à base radial (RBF) : $K(x, y) = \exp(-\|x - y\|^2 / 2 \cdot \sigma^2)$.
 - noyau exponentielle RBF : $K(x, y) = \exp(-\|x - y\| / 2 \cdot \sigma^2)$.
 - noyau sigmoïde : $K(x, y) = \tanh(x^T \cdot y + 1)$.
- où, $\sigma > 0$ c'est le constant de la fonction RBF.

II.3 Régression à vecteurs de support (SVR)

Les SVMs peuvent également être appliquées à des problèmes de régression par l'introduction d'une fonction de perte [29]. La fonction de perte doit être modifiée pour inclure une mesure de distance. La figure II.2 illustre quatre fonctions de perte possible [13].

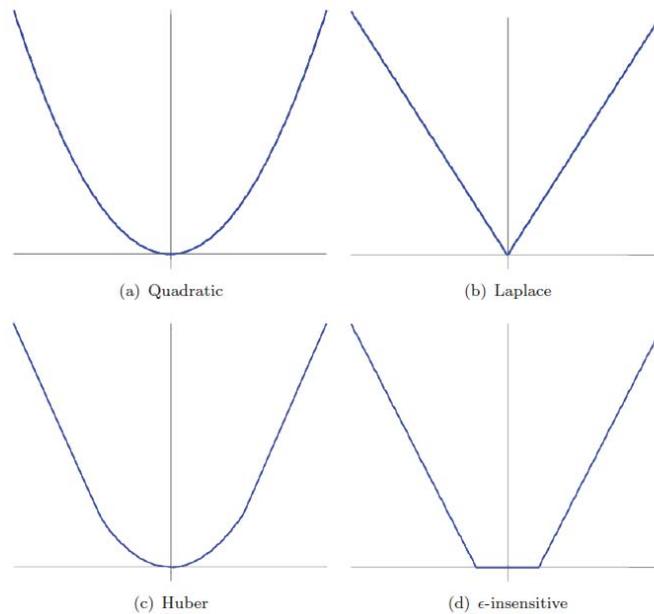


FIG. II.2 – Les fonctions de perte, [13]

La fonction de perte de la figure II.2 (a) correspond au critère d'erreur des moindres carrés classiques. La fonction de perte montrée dans la figure II.2 (b) est une fonction de perte de Laplace qui est moins sensible aux valeurs aberrantes que la fonction de perte quadratique. Huber [13], a proposé la fonction de perte de la figure II.2 (c) comme une fonction de perte robuste qui a des propriétés optimales lorsque la distribution sous-jacente des données est inconnue. Ces trois fonctions de perte ne produiront aucun clairsemé dans les vecteurs de support. Pour résoudre ce problème, la fonction de perte ϵ -insensitive (cf. la

figure II.2 (d)) proposée par Vapnik [35], comme une approximation de la fonction de perte Huber qui permet d'obtenir un ensemble de vecteurs de support clairsemé.

II.3.1 Régression Linéaire

Considérons le problème d'approximation de l'ensemble des données

$$D = \{(x^1, y^1), \dots, (x^l, y^l)\}, x \in \mathbb{R}^n, y \in \mathbb{R} \quad (\text{II.14})$$

avec une fonction linéaire,

$$f(x) = \langle w, x \rangle + b \quad (\text{II.15})$$

la fonction de régression optimale est donnée par le minimum de la fonctionnelle,

$$\Phi(w, \xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m (\xi_i^- + \xi_i^+) \quad (\text{II.16})$$

où, w est le vecteur des poids, C détermine l'échange entre la complexité du $f(x)$ et le montant maximal duquel les écarts supérieurs à ϵ sont tolérés, et ξ_i^-, ξ_i^+ sont les écarts inférieurs et supérieurs représentant les contraintes sur la sortie de système.

II.3.1.1 La fonction de perte ϵ -insensitive

Ici on utilise une fonction de perte ϵ -insensitive, présentée dans la Figure II.2 (d), comme suit :

$$L_\epsilon(y) = \begin{cases} 0 & \text{pour } |f(x) - y| < \epsilon \\ |f(x) - y| - \epsilon & \text{otherwise} \end{cases} \quad (\text{II.17})$$

la solution est donnée par :

$$\max_{\alpha, \alpha^*} F(\alpha, \alpha^*) = \max_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle + \sum_{i=1}^l \alpha_i (y_i - \epsilon) - \alpha_i^* (y_i + \epsilon) \quad (\text{II.18})$$

où alternativement :

$$\bar{\alpha}, \bar{\alpha}^* = \arg \min_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle - \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i + \sum_{i=1}^l (\alpha_i + \alpha_i^*) \epsilon \quad (\text{II.19})$$

avec les contraintes :

$$\begin{aligned} 0 \leq \alpha, \alpha^* \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \end{aligned} \quad (\text{II.20})$$

La résolution de l'équation (II.18) avec les contraintes données dans l'équation (II.19) détermine les multiplicateurs de Lagrange α, α^* , et la fonction de régression est donnée par l'équation (II.15), où :

$$\begin{aligned}\bar{w} &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i \\ \bar{b} &= -\frac{1}{2} \langle \bar{w}, (x_r + x_s) \rangle.\end{aligned}\tag{II.21}$$

Les conditions de Karush-Kuhn-Tucker (KKT) qui sont satisfaites par la solution sont :

$$\bar{\alpha} \bar{\alpha}^* = 0, \quad i = 1, \dots, l.\tag{II.22}$$

Par conséquent, les vecteurs de support sont des points où exactement l'un des multiplicateurs de Lagrange est supérieur à zéro. Lorsque $\epsilon = 0$, nous obtenons la fonction de perte L_1 , et le problème d'optimisation est simplifié :

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_i \beta_j \langle x_i, x_j \rangle - \sum_{i=1}^l \beta_i y_i\tag{II.23}$$

avec les contraintes :

$$\begin{aligned}-C &\leq \beta_i \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l \beta_i &= 0\end{aligned}\tag{II.24}$$

et la fonction de régression est donnée par l'équation (II.15), où le vecteur de poids \bar{w} et le terme de biais b sont donnés comme suit :

$$\begin{aligned}\bar{w} &= \sum_{i=1}^l \beta_i x_i \\ \bar{b} &= -\frac{1}{2} \langle \bar{w}, (x_r + x_s) \rangle.\end{aligned}\tag{II.25}$$

II.3.1.2 La fonction de perte quadratique

On utilise maintenant une fonction de perte quadratique, (cf. Figure II.2 (a)), on obtient :

$$L_{quad}(f(x) - y) = (f(x) - y)^2\tag{II.26}$$

La solution est donnée par :

$$\begin{aligned}\max_{\alpha, \alpha^*} F(\alpha, \alpha^*) &= \max_{\alpha, \alpha^*} -\frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i, x_j \rangle \\ &\quad + \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i - \frac{1}{2C} \sum_{i=1}^l (\alpha_i^2 - (\alpha_i^*)^2)\end{aligned}\tag{II.27}$$

L'optimisation correspondante peut être simplifiée en exploitant les conditions KKT, l'équation (II.22) et en notant que ceux-ci impliquent $\beta_i^* = |\beta_i|$, et le problème d'optimisation est,

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_i \beta_j \langle x_i, x_j \rangle - \sum_{i=1}^l \beta_i y_i + \frac{1}{2C} \sum_{i=1}^l \beta_i^2 \quad (\text{II.28})$$

avec les contraintes :

$$\sum_{i=1}^l \beta_i = 0 \quad (\text{II.29})$$

et la fonction de régression est donnée par les équations (II.15) et (II.25).

II.3.1.3 Exemple illustratif

Considérons par exemple les données figurant dans le tableau II.1 , [13]. La solution SVR pour une fonction de perte ϵ -insensitive (Figure II.2 (d)) sans contrôle de la capacité supplémentaire est illustré à la figure II.3.

x	Y
1.0	-1.6
3.0	-1.8
4.0	-1.0
5.6	1.2
7.8	2.2
10.2	6.8
11.0	10.0
11.5	10.0
12.7	10.0

TAB. II.1 – Le jeu de données observées

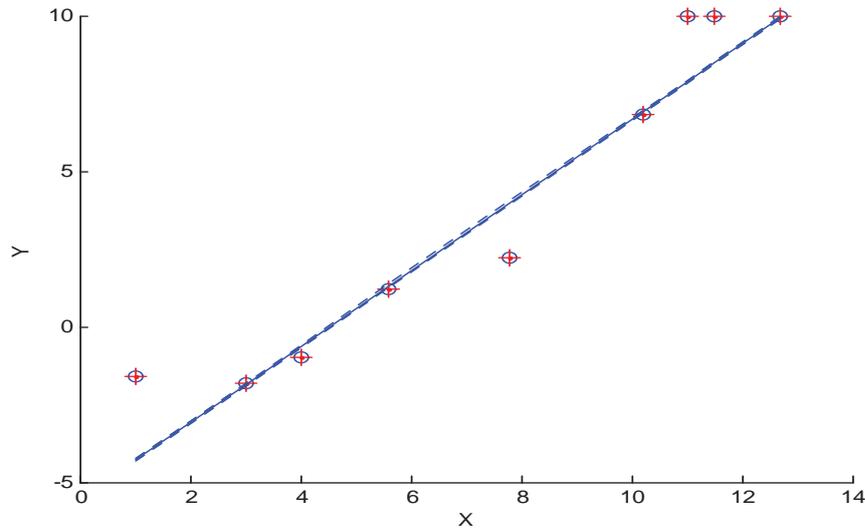


FIG. II.3 – Régression linéaire avec les SVMs

II.3.2 Régression non linéaire

Un modèle non-linéaire est généralement nécessaire pour bien traiter les données du modèle, l'idée des SVM est de changer l'espace des données. Une cartographie non-linéaire peut être utilisée pour cartographier les données dans un espace de grande dimension où la régression linéaire est effectuée. L'approche du noyau est utilisée pour traiter la malédiction de la dimensionnalité.

La solution de la régression non linéaire à vecteur de support, utilisant une fonction de perte ϵ -insensitive (Figure II.2 (d)) est donnée par :

$$\max_{\alpha, \alpha^*} F(\alpha, \alpha^*) = \max_{\alpha, \alpha^*} \sum_{i=1}^l \alpha_i^* (y_i - \epsilon) - \alpha_i (y_i + \epsilon) - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j) \quad (\text{II.30})$$

avec les contraintes :

$$\begin{aligned} 0 \leq \alpha, \alpha^* \leq C, \quad i = 1, \dots, l \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \end{aligned} \quad (\text{II.31})$$

La solution de la programmation quadratique (II.30) sous les contraintes (II.31), détermine les multiplicateurs de Lagrange α_i, α_i^* , et la fonction de régression est donnée par :

$$f(x) = \sum_{SV_s} (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) + b \quad (\text{II.32})$$

Où :

$$\begin{aligned}\langle \bar{w}, x \rangle &= \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_j) \\ \bar{b} &= -\frac{1}{2} \sum_{i=1}^l (\alpha_i - \alpha_i^*) (K(x_i, x_r) + K(x_i, x_s)).\end{aligned}\tag{II.33}$$

La contrainte d'égalité peut être supprimé si le noyau contient un terme de biais, b étant logée dans la fonction du noyau, et la fonction de régression est donnée par :

$$f(x) = \sum_{i=1}^l (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x).\tag{II.34}$$

Les critères d'optimisation pour les autres fonctions de pertes sont également obtenus en remplaçant le produit scalaire par une fonction noyau. La fonction de perte ϵ -insensitive est intéressante, car contrairement aux fonctions de coût quadratique et d' Huber, où tous les points de données sont des vecteurs de support, la solution SV peut être rare. La fonction de perte quadratique donne une solution équivalente à la régression d'arête, ou la régularisation d'ordre zéro, avec le paramètre de régularisation $\lambda = 1/C$.

II.3.2.1 Exemple illustratif

Pour illustrer quelques solutions de la régression non linéaire à vecteur de support, différentes fonctions noyaux (Kernel) ont été employées pour modeler les données de régression dans le tableau (II.1) [13], avec une fonction de perte ϵ -insensitive (*Upsilon* $\epsilon = 0.5$) et sans aucun contrôle de capacité supplémentaire. La figure (II.4) montre la solution de la régression à vecteur de support (SVR) pour un noyau polynomial de degré $d = 2$. Avec les SVs étant entouré avec une croix rouge. Les deux lignes pointillées représentent la région epsilon-insensitive (ou tube epsilon) autour de la solution. Si les points de toutes les données se trouvent dans cette région il n'y aura aucune erreur associée à la fonction de perte. Les résultats montrent qu'il n'ya pas de vecteurs de support au sein de la région ϵ -insensitive.

La figure (II.5) montre la solution de la SVR pour un noyau RBF (radial basis function) avec $\sigma = 1.0$. Dans cet exemple, le modèle est suffisamment souple pour modéliser la fonction avec zéro erreur associée à la fonction de perte : les points de toutes les données se trouvent sur ou à l'intérieur de la zone ϵ -insensitive.

La figure (II.6) montre la solution pour un noyau RBF exponentielle, qui est une spline linéaire par morceaux. Bien que ce modèle ait une capacité élevée, il affiche un comportement sensible dans les régions extrémité.

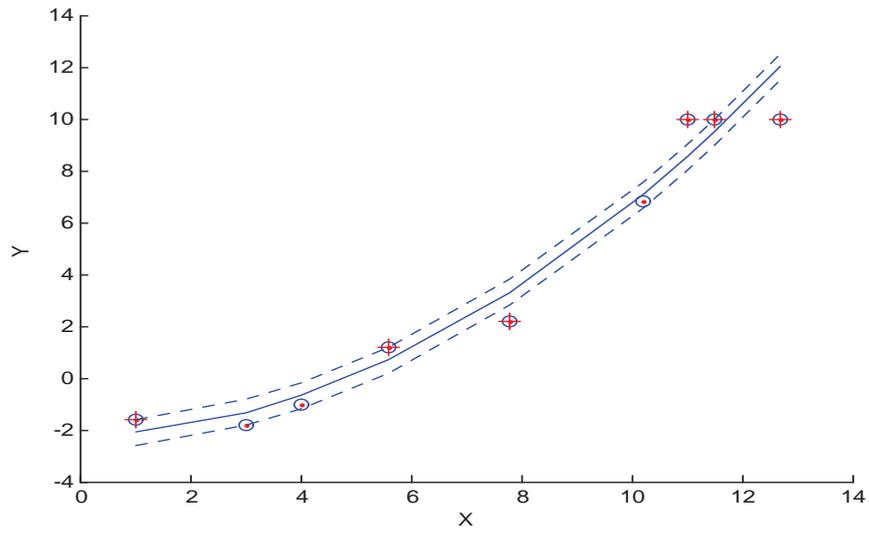


FIG. II.4 – Régression non linéaire polynomiale de degré 2

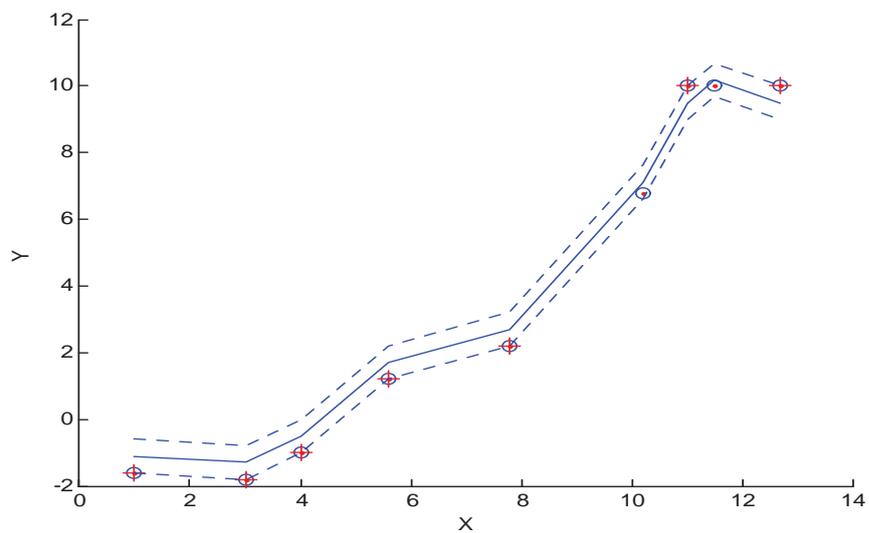


FIG. II.5 – Régression non linéaire avec RBF noyau

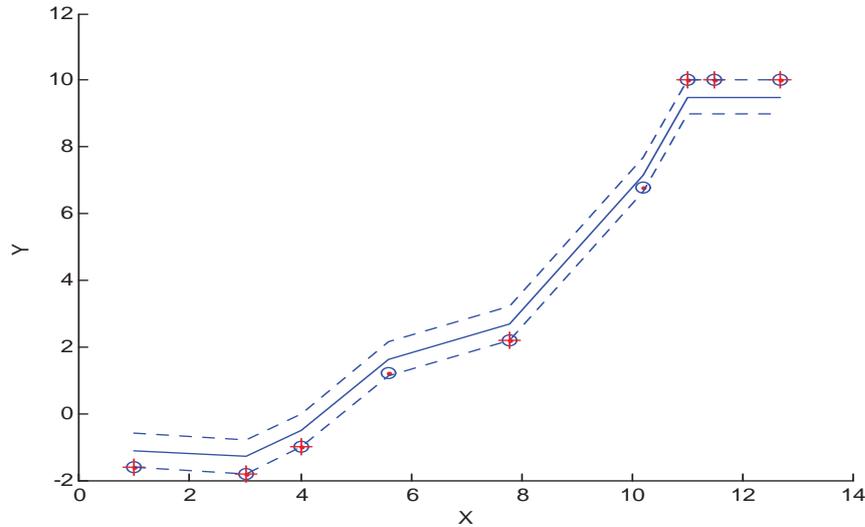


FIG. II.6 – Régression non linéaire avec exp RBF noyau

II.4 Régression floue à vecteur de support (FSVR)

Dans la modélisation des systèmes où l'information disponible est incertaine, nous devons traiter une structure floue du système considéré. Cette structure est représentée comme une fonction floue dont les paramètres sont des intervalles flous. Les fonctions floues sont définies par principe d'extension de Zadeh [39]. Dans cette section, nous intégrons les concepts de la logique floue dans le modèle de la régression avec les SVMs. Les paramètres ont été identifiés dans le modèle de régression à vecteur de support, tels que les composantes du vecteur de poids et le terme de biais, et les résultats désirés sont des nombres flous. Pour la simplification du calcul, nous supposons que les paramètres flous sont des nombres flous triangulaires symétriques (NFTS), [14].

II.4.1 La programmation quadratique (QP)

Tout d'abord, les composants de vecteur de poids et le terme de biais utilisées dans le modèle de la fonction de régression sont des NFTSs. Étant donné le vecteur de poids floue $W_i = (w_i, c_i)$, où w_i le midpoint, et c_i le radius, de même le terme de biais flou $B = (b, d)$, où b le midpoint, et d le radius.

Le modèle est donnée par :

$$Y = W_1x_1 + W_2x_2 + \dots + W_nx_n + B = \langle W.x \rangle + B \quad (\text{II.35})$$

est définie par la fonction d'appartenance suivante :

$$\mu_Y(y) = \begin{cases} 1 - \frac{|y - (\langle W.x \rangle + b)|}{\langle c. |x| \rangle + d}, & \text{pour } x \neq 0 \\ 1, & \text{pour } x = 0 \text{ et } y = 0 \\ 0, & \text{pour } x = 0 \text{ et } y \neq 0 \end{cases} \quad (\text{II.36})$$

Où , $\mu_Y(y) = 0$, quand $\langle c. |x| \rangle + d \leq |y - (\langle W.x \rangle + B)|$

Puis, nous traitons la sortie floue désirée dans la tâche de régression. Les données observées, noté $\tilde{Y}_i = (y_i, e_i)$, sont également des nombres flous, où y_i est le midpointhe, et e_i le radius.

La fonction d'appartenance de \tilde{Y}_i est donnée par :

$$\mu_{\tilde{Y}}(y) = 1 - \frac{|y - y_i|}{e_i} \quad (\text{II.37})$$

Pour formuler un modèle de régression linéaire floue, les points suivants ont été retenus :

1. Les données peuvent être représentées par un modèle linéaire floue $Y_i^* = \langle W^*.x_i \rangle + B^*$, donné x_i, Y_i^* peut être obtenu comme suit

$$\mu_{Y_i^*}(y) = 1 - \frac{|y - (\langle W.x_i \rangle + b)|}{\langle c. |x_i| \rangle + d} \quad (\text{II.38})$$

2. Le degré d'ajustement du modèle linéaire floue estimée.

$$Y_i^* = \langle W^*.x_i \rangle + B^*$$

pour les données observées $\tilde{Y}_i = (y_i, e_i)$, est mesurée par l'indice \bar{h}_i , ce qui maximise h , sujet à $\tilde{Y}_i^h \subset Y_i^{*h}$, où :

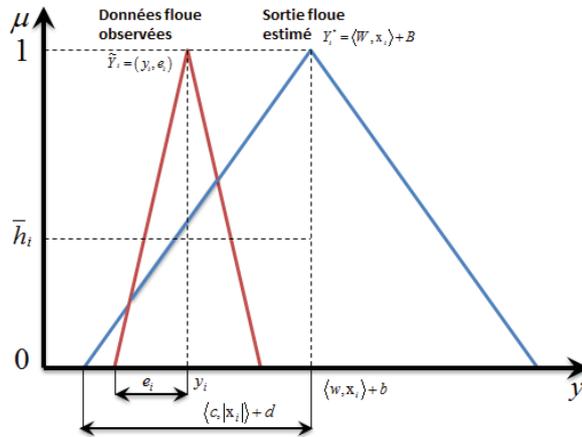
$$\begin{aligned} \tilde{Y}_i^h &= \{y \mid \mu_{\tilde{Y}}(y) \geq h\} \\ Y_i^{*h} &= \{y \mid \mu_{Y_i^*}(y) \geq h\} \end{aligned}$$

Qui sont des ensembles par h -coupe. Le degré d'ajustement optimal pour toutes les données est définie par, $\min_j |\bar{h}_j|$.

3. L'imprécision du modèle flou linéaire est définie par : $\frac{1}{2} \|c\|^2 + d$.

4. Le terme de contrôle de la capacité du modèle linéaire flou est définie par : $\|w\|^2$.

La résolution du problème de régression donne le vecteur de poids flou $W = (w, c)$ et le terme de biais flou $B = (b, d)$, tels que le degré d'ajustement entre la sortie estimée Y_i^{*h} et la sortie désirée Y_i^{*h} , est une constante donnée H pour tous les i , où H est choisi comme


 FIG. II.7 – Explication du modèle avec h -coupe, [14]

le degré de l'ajustement du modèle linéaire floue par l'utilisateur. La valeur \bar{h}_i peuvent être obtenus à partir de :

$$\bar{h}_i = 1 - \frac{|y_i - (\langle W, x_i \rangle + b)|}{(\langle c, |x_i| \rangle + d) - e_i} \quad (\text{II.39})$$

qui est la même que celle obtenue par Tanaka [30], et il est illustré dans la figure (II.7). Notre problème de régression est donc :

$$\begin{aligned} \underset{w, c, b, d, \xi_i}{\text{Minimisez}} \quad & J = \frac{1}{2} \|w\|^2 + K \left(\frac{1}{2} \|c\|^2 + d \right) + P \sum_{i=1}^N \xi_i \\ \text{sujet à :} \quad & \bar{h}_i \geq H \quad \text{pour tout } i = 1, \dots, N \end{aligned} \quad (\text{II.40})$$

où, $\|w\|^2$, est le terme qui caractérise la complexité du modèle, et $\frac{1}{2} \|c\|^2 + d$ est le terme qui caractérise l'imprécision du modèle.

Plus d'imprécision dans le modèle linéaire régressif floue cela signifie plus d'inexactitude dans le résultat de la régression. K est le paramètre de différence choisi par le décideur. La valeur de H détermine la borne inférieure de mise en place du modèle flou linéaire, et \bar{h}_i est le degré d'ajustement du modèle linéaire floue estimée $Y_i^* = \langle W^*, x_i \rangle + B^*$ pour les données floues de la sortie désirée $\tilde{Y}_i = (y_i, e_i)$. $\{\xi_i\}_{i=1, \dots, N}$, sont des ensembles de variables qui mesurent l'excédent de quantité de la variation des contraintes pour chaque point, où P est un paramètre de pénalité fixe choisi par l'utilisateur, une plus grande valeur de P correspond à attribuer une pénalité plus élevée aux erreurs. La figure (II.8) représente graphiquement cette situation [14].

Plus précisément, selon (II.39), notre problème consiste à trouver le vecteur de poids flou $W^* = (w, c)$ et le terme de biais flou $B^* = (b, d)$, qui est solution du problème de

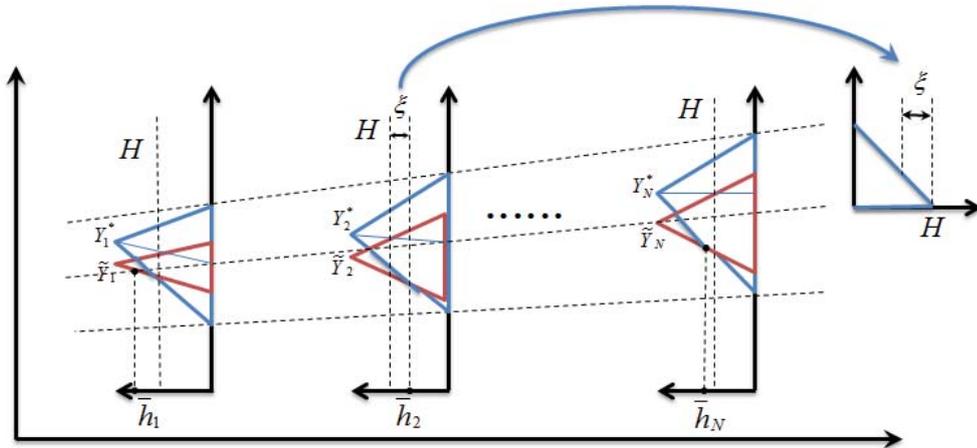


FIG. II.8 – Explication du modèle de régression linéaire floue, [14]

programmation quadratique suivant :

$$\begin{aligned}
 & \underset{w,c,b,d,\xi_i}{\text{Minimisez}} \quad J = \frac{1}{2} \|w\|^2 + K \left(\frac{1}{2} \|c\|^2 + d \right) + P \sum_{i=1}^N (\xi_{1i} + \xi_{2i}) \\
 & \text{sujet à :} \\
 & (\langle w \cdot x_i \rangle + b) + (1 - H) (\langle c \cdot |x_i| \rangle + d) \geq y_i + (1 - H) e_i - \xi_{1i} \\
 & - (\langle w \cdot x_i \rangle + b) + (1 - H) (\langle c \cdot |x_i| \rangle + d) \geq -y_i + (1 - H) e_i - \xi_{2i} \\
 & \text{et} \\
 & d \geq 0, c_j \geq 0, \quad \xi_{1i}, \xi_{2i} \geq 0 \\
 & \text{pour} \quad i = 1, \dots, N, j = 1, \dots, n
 \end{aligned} \tag{II.41}$$

La différence entre le modèle de régression simple avec les SVMs et la technique de la régression floue à vecteur de support reside dans le fait que le premier modèle cherche une fonction linéaire qui a au plus ϵ déviation des cibles réellement obtenues y_i pour toute les données, alors que le deuxième modèle de régression floue avec les SVMs cherche une fonction floue linéaire avec des paramètres flous qui a au moins le degré d'ajustement H des cibles floues désirées de toute les données.

Nous pouvons trouver la solution de ce problème d'optimisation donné par (II.41) dans

les variables duales en trouvant le point selle du Lagrangien.

$$\begin{aligned}
 L = & \frac{1}{2} \|w\|^2 + K \left(\frac{1}{2} \|c\|^2 + d \right) + P \sum_{i=1}^N (\xi_{1i} + \xi_{2i}) \\
 & - \sum_{i=1}^N \alpha_{1i} (\langle W \cdot x_i \rangle + b + (1-H) (\langle c \cdot |x_i| \rangle + d)) \\
 & - y_i - (1-H) e_i + \xi_{1i} \\
 & - \sum_{i=1}^N \alpha_{2i} (-\langle W \cdot x_i \rangle - b + (1-H) (\langle c \cdot |x_i| \rangle + d)) \\
 & + y_i - (1-H) e_i + \xi_{2i} \\
 & - \sum_{i=1}^N \rho_{1i} \xi_{1i} - \sum_{i=1}^N \rho_{2i} \xi_{2i} - \gamma d
 \end{aligned} \tag{II.42}$$

où $\alpha_{1i}, \alpha_{2i}, \rho_{1i}, \rho_{2i}$ et γ sont les multiplicateurs de Lagrange non négatif. Après différenciation par rapport à w, c, b, d, ξ_{1i} et ξ_{2i} et mise à zéro des résultats, nous obtenons :

$$\partial L / \partial w = 0 \Rightarrow w = \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) x_i \tag{II.43}$$

$$\partial L / \partial c = 0 \Rightarrow c = \frac{1-H}{K} \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) |x_i| \tag{II.44}$$

$$\partial L / \partial b = 0 \Rightarrow \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) = 0 \tag{II.45}$$

$$\partial L / \partial d = 0 \Rightarrow \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) = \frac{K - \gamma}{1 - H} \tag{II.46}$$

$$\partial L / \partial \xi_{1i} = 0 \Rightarrow \alpha_{1i} = P - \rho_{1i} \text{ et } \alpha_{1i} \leq P \tag{II.47}$$

$$\partial L / \partial \xi_{2i} = 0 \Rightarrow \alpha_{2i} = P - \rho_{2i} \text{ et } \alpha_{2i} \leq P \tag{II.48}$$

En substituant (II.43) - (II.48) dans (II.42), nous obtenons :

$$\begin{aligned}
 L = & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} - \alpha_{2i}) (\alpha_{1j} - \alpha_{2j}) \langle x_i \cdot x_j \rangle \\
 & - \frac{(1-H)^2}{2K} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} + \alpha_{2i}) (\alpha_{1j} + \alpha_{2j}) \langle |x_i| \cdot |x_j| \rangle \\
 & + \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) y_i + (1-H) \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) e_i
 \end{aligned} \tag{II.49}$$

Donc, le problème dual est :

$$\begin{aligned}
 \underset{\alpha_{1i}, \alpha_{2i}}{\text{Maximisez}} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} - \alpha_{2i})(\alpha_{1j} - \alpha_{2j}) \langle x_i, x_j \rangle \\
 & - \frac{(1-H)^2}{2K} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} + \alpha_{2i})(\alpha_{1j} + \alpha_{2j}) \langle |x_i|, |x_j| \rangle \\
 & + \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) y_i + (1-H) \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) e_i,
 \end{aligned} \tag{II.50}$$

sujet à :

$$\begin{aligned}
 \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) &= 0, \quad \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) \leq \frac{K}{1-H} \\
 \alpha_{1i}, \alpha_{2i} &\in [0, P].
 \end{aligned}$$

Il est facile de constater que :

$$w = \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) x_i \tag{II.51}$$

$$c = \frac{1-H}{K} \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) |x_i| \tag{II.52}$$

Bien que les paramètres b et d peuvent être déterminés par les conditions de KKT

$$\alpha_{1i} (\langle w, x_i \rangle + b + (1-H) (\langle c, |x_i| \rangle + d) - y_i - (1-H) e_i + \xi_{1i}) = 0 \tag{II.53}$$

$$\alpha_{2i} (-\langle w, x_i \rangle - b + (1-H) (\langle c, |x_i| \rangle + d) + y_i - (1-H) e_i - \xi_{2i}) = 0 \tag{II.54}$$

$$(P - \alpha_{1i}) \xi_{1i} = 0 \tag{II.55}$$

$$(P - \alpha_{2i}) \xi_{2i} = 0 \tag{II.56}$$

Pour certains $\alpha_{1i}, \alpha_{2j} \in (0, P)$, nous avons $\xi_{1i} = \xi_{2j} = 0$ et d'ailleurs le deuxième facteur de (II.53) et (II.54) doit disparaître. Par conséquent, b et d peuvent être calculées pour certains $\alpha_{1i}, \alpha_{2j} \in (0, P)$, comme suit :

$$b = -\frac{1}{2} \begin{pmatrix} \langle w, x_i \rangle + \langle w, x_j \rangle + (1-H) (\langle c, |x_i| \rangle - \langle c, |x_j| \rangle) \\ -y_i - y_j - (1-H) (e_i - e_j) \end{pmatrix} \tag{II.57}$$

$$d = -\frac{1}{2(1-H)} \begin{pmatrix} \langle w, x_i \rangle - \langle w, x_j \rangle + (1-H) (\langle c, |x_i| \rangle + \langle c, |x_j| \rangle) \\ -y_i + y_j - (1-H) (e_i + e_j) \end{pmatrix} \tag{II.58}$$

La fonction de régression linéaire floue est définie par la fonction d'appartenance suivante :

$$\mu_{Y^*}(y) = \begin{cases} 1 - \frac{|y - (\sum_{k=1}^N (\alpha_{1k} - \alpha_{2k}) \langle x, x_k \rangle + b)|}{\frac{(1-H)}{K} \sum_{k=1}^N (\alpha_{1k} + \alpha_{2k}) \langle |x|, |x_k| \rangle + d}, & \text{pour } x \neq 0 \\ 1, & \text{pour } x = 0 \text{ et } y = 0 \\ 0, & \text{pour } x = 0 \text{ et } y \neq 0 \end{cases} \tag{II.59}$$

II.4.2 Extension au modèle de régression non linéaire floue

Il y a seulement quelques papiers sur la régression non linéaire floue [5],[3], [4]. Ils assument généralement le modèle sous-jacent et traitent les procédures d'estimation de certains modèles particuliers tels que linéaires, polynômiaux et exponentielles. Cependant, nous pensons que ces méthodes sont dépendantes d'un modèle irréalisable. Nous allons utiliser l'idée de SVM pour la régression non linéaire simple [35]. L'idée de base est de changer l'espace des données. La transformation non linéaire des données peut permettre une représentation linéaire des données observées x_i par $\Phi : \mathcal{R}^n \rightarrow F$, dans un nouvel espace de grande dimension F et en appliquant ensuite l'algorithme de régression linéaire standard dans cet espace. Une propriété essentielle de l'algorithme proposé est que les quantités que nous avons besoin de calculer sont des produits scalaires, de la forme $\langle x_i, x_j \rangle$ et $\langle |x_i|, |x_j| \rangle$. L'algorithme dépendrait seulement des données par les produits intérieurs dans F , c'est à dire, sur les fonctions de la forme $\langle \Phi(x_i), \Phi(x_j) \rangle$ et $\langle \Phi(|x_i|), \Phi(|x_j|) \rangle$.

La forme de la fonction Φ n'a pas besoin d'être connue car elle est implicitement définie par le choix de la fonction noyau (Kernel), $K(x, y) \equiv \langle \Phi(x), \Phi(y) \rangle$. Par conséquent, il suffit de savoir que $K(x_i, x_j) \equiv \langle \Phi(x_i), \Phi(x_j) \rangle$ et $K(|x_i|, |x_j|) \equiv \langle \Phi(|x_i|), \Phi(|x_j|) \rangle$ au lieu de définir explicitement $\Phi(\cdot)$.

Ainsi, par substitution $\langle x_i, x_j \rangle$ et $\langle |x_i|, |x_j| \rangle$ avec $K(x_i, x_j)$ et $K(|x_i|, |x_j|)$ respectivement, nous obtenons le problème dual d'optimisation quadratique donné par (II.60). Ici, il convient de noter que les contraintes ne sont pas modifiées [14].

$$\begin{aligned}
 \underset{\alpha_{1i}, \alpha_{2i}}{\text{Maximisez}} \quad & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} - \alpha_{2i})(\alpha_{1j} - \alpha_{2j}) K(x_i, x_j) \\
 & - \frac{(1-H)^2}{2K} \sum_{i=1}^N \sum_{j=1}^N (\alpha_{1i} + \alpha_{2i})(\alpha_{1j} + \alpha_{2j}) K(|x_i|, |x_j|) \\
 & + \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) y_i + (1-H) \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) e_i,
 \end{aligned} \tag{II.60}$$

soit à :

$$\begin{aligned}
 \sum_{i=1}^N (\alpha_{1i} - \alpha_{2i}) &= 0, \quad \sum_{i=1}^N (\alpha_{1i} + \alpha_{2i}) \leq \frac{K}{1-H} \\
 \alpha_{1i}, \alpha_{2i} &\in [0, P].
 \end{aligned}$$

En utilisant différentes fonctions noyau, on peut construire différents types de machines d'apprentissage avec des fonctions arbitraires de régression non linéaire dans l'espace d'entrée. La fonction de régression non linéaire floue est définie par la fonction d'appartenance

suivante :

$$\mu_{Y^*}(y) = \begin{cases} 1 - \frac{|y - (\sum_{k=1}^N (\alpha_{1k} - \alpha_{2k}) K(x, x_k) + b)|}{\frac{(1-H)}{K} \sum_{k=1}^N (\alpha_{1k} + \alpha_{2k}) K(|x|, |x_k|) + d}, & \text{pour } x \neq 0 \\ 1, & \text{pour } x = 0 \text{ et } y = 0 \\ 0, & \text{pour } x = 0 \text{ et } y \neq 0 \end{cases} \quad (\text{II.61})$$

Les représentations des noyaux (Kernels) offrent une solution alternative en projetant les données dans un espace de grande dimension pour augmenter la puissance de calcul du modèle régressif linéaire floue. L'utilisation d'un modèle de régression linéaire floue dans sa représentation dual, permet d'effectuer cette étape implicitement. Comme indiqué au-dessus, les données de l'exemple ne semblent jamais isolées, mais toujours sous la forme de produits scalaires entre les paires des données. L'avantage d'utiliser le modèle de régression dans la représentation duale est que, dans cette représentation, le nombre des paramètres ajustables ne dépend pas du nombre d'attributs utilisés. En remplaçant le produit scalaire par une fonction noyau choisie, on peut effectuer une représentation implicite non linéaire à un espace de grande dimension sans augmenter le nombre des paramètres réglables [9].

Une autre attraction de la méthode du noyau est que les algorithmes d'apprentissages peuvent être largement découplés des spécificités de la zone d'application, qui doit simplement être encodés dans la conception d'une fonction noyau approprié. Par conséquent, le problème de choix d'une structure pour un modèle régressif est remplacé par le problème de choix d'un noyau approprié pour la SVM.

II.4.3 Exemple illustratif

Ce jeu de données est présenté précédemment dans le chapitre I (cf. tableau I.1). Ces données sont tirées à partir de Tanaka et Lee [31]. On remarquera ici que la représentation des intervalles par leurs midpoint, radius à été adoptée. Le jeu de données comporte donc $M = 8$ échantillons.

Nous considérons dans un premier temps une régression linéaire floue. Le modèle linéaire flou pour ces données est supposé :

$$Y(x) = Wx + B \quad (\text{II.62})$$

On applique l'algorithme de l'approche du SVM floue pour les données présentées dans le tableau (I.1). Dans cet exemple, nous avons mis les paramètres $H = 0.5$, $P = 15$ et $K = 25$.

On obtient le modèle régressif suivant :

$$Y^*(x) = (8.00, 0.58)x + (1.1525, 2.9310) \quad (\text{II.63})$$

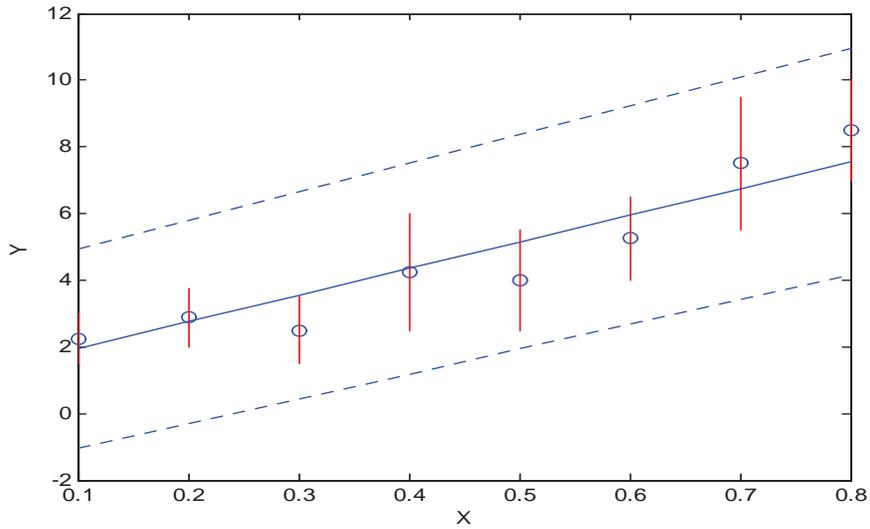


FIG. II.9 – Représentation du modèle linéaire identifié par SVM floue

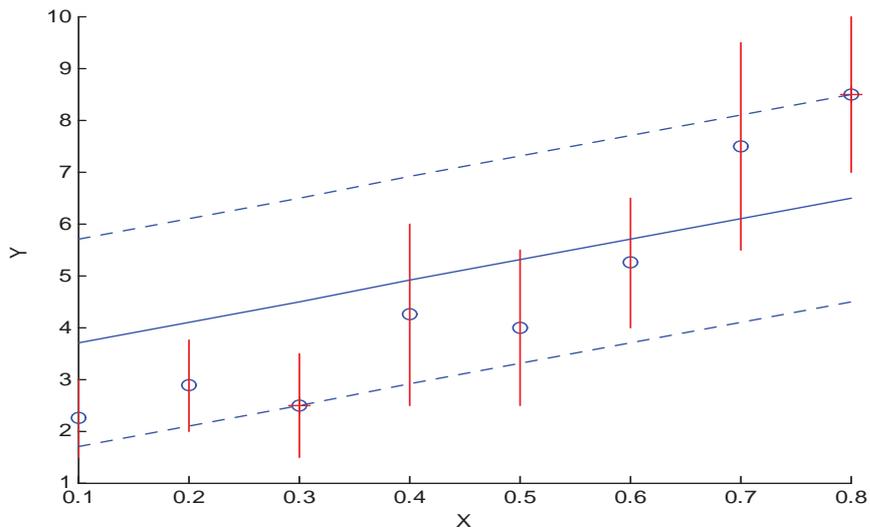


FIG. II.10 – Représentation du modèle linéaire identifié par SVM simple

On introduit maintenant le résultat de Tanaka [32], pour comparer et évaluer les deux méthodes dans le cas linéaire. Le modèle linéaire flou obtenu par la méthode de Tanaka avec $H = 0.5$, est :

$$Y^*(x) = (9.5417, 0.5)x + (0.125, 3.0410) \quad (\text{II.64})$$

On peut voir que les coefficients flous sont presque similaires. Nous avons également comparé les résultats obtenues par la méthode de régression par SVM simple, mais l'approche

de régression par SVM simple ne peut pas manipuler les données floues directement. Ainsi, dans cette expérience, on utilise $\{(x_i, y_i + e_i)\}_{i=1}^N$ et $\{(x_i, y_i - e_i)\}_{i=1}^N$ comme des données d'apprentissage pour estimer le modèle de régression par SVM, et pour la valeur de epsilon-insensitive est définie par $\epsilon = \max(e_i) = 2$. Les figures (II.9), (II.10) montrent le modèle identifié obtenue par SVM floue et SVM simple respectivement.

Dans la figure (II.11), le modèle de régression non linéaire semble approprié pour ces données. Par conséquent, nous appliquons la méthode de la régression floue non linéaire par SVM pour analyser cet ensemble de données. Ici, on utilise d'abord un noyau polynomial de degré trois pour l'estimation du modèle non linéaire floue. Notamment, le modèle non linéaire floue pour ces données est supposé que :

$$Y(x) = W_1.x^3 + W_2.x^2 + W_3.x + B \quad (\text{II.65})$$

En appliquant l'approche de SVM floue sur l'ensemble de données, nous obtenons le modèle de régression linéaire floue suivant :

$$\begin{aligned} Y^*(x) = & (1.6841, 0.1351) .x^3 + (2.7499, 0.4327) .x^2 \\ & + (1.8701, 0.8433) .x + (1.8418, 1.9585) \end{aligned} \quad (\text{II.66})$$

Dans cet exemple, nous avons mis les paramètres $H = 0.5$, $P = 30$ et $K = 35$ [14]. Le modèle de régression linéaire floue obtenu par l'approche de Tanaka, avec $H = 0.5$ est :

$$\begin{aligned} Y^*(x) = & (29.9037, 0.157) .x^3 + (-30.085, 0.27) .x^2 \\ & + (16.0619, 0.5) .x + (0.3222, 2.3025) \end{aligned} \quad (\text{II.67})$$

Pour la régression non linéaire floue par les SVM (FSVR), on n'obtient pas de bons résultats dans le cas d'un noyau RBF. Donc on utilise dans ce cas presque la même technique mais avec des nombres flous triangulaires non symétriques.

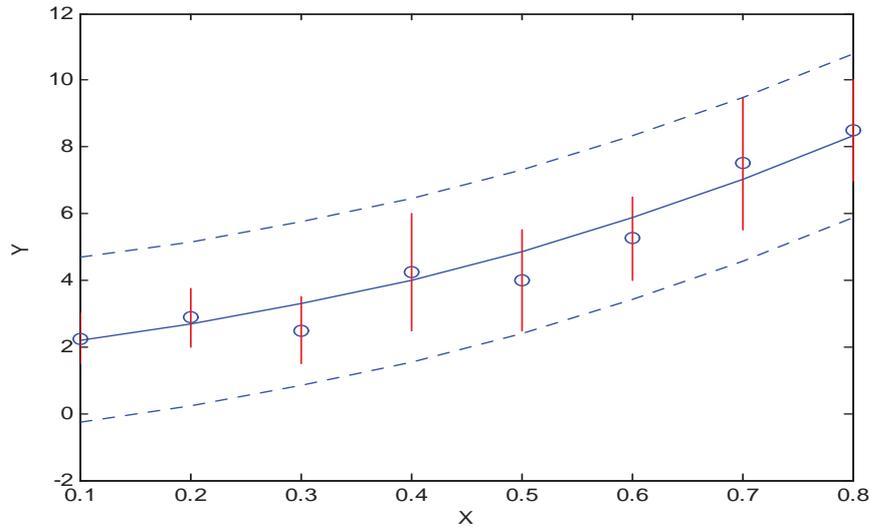


FIG. II.11 – Modèle non linéaire identifié par SVM floue (noyau polynomial de degré 3)

II.5 Une nouvelle approche

Dans cette section on propose une nouvelle approche d'identification floue basée sur la régression floue à vecteur de support (FSVR) et l'algorithme d'optimisation par essaim de particules unifiée (UPSO).

Premièrement, le modèle FSVR, avec la fonction noyau de la fonction de base radiale (RBF), est choisi pour faciliter le modèle d'identification. Ensuite, un optimiseur global, UPSO est utilisé pour optimiser les hyper-paramètres nécessaires à FSVR. Par la suite, une procédure est mise en avant servira comme outil efficace pour la prévision. L'application du modèle proposé sur des données différentes a donné des résultats prometteurs. En outre, les résultats expérimentaux démontrent en outre que le modèle proposé réalise une meilleure performance d'identification que la technique standard de la régression floue à vecteur de support. Ici, on utilise pour FSVR des nombres flous triangulaire non symétrique.

Tout d'abord on va présenter les techniques utilisée dans notre nouvelle approche afin d'expliquer les différentes étapes de cette dernière.

II.5.1 La régression floue par SVM de Hong

Dans cette section, nous examinons brièvement la régression floue à vecteur de support (FSVR) de Hong et al [15]-[17], cette technique utilise les nombres flous triangulaires non symétriques.

Pour ce faire nous avons besoin de quelques préliminaires. Soit $A = (m, r_L, r_R)$ un nombre flou triangulaire non symétrique, où, m est la valeur modale de A et r_L, r_R sont les radius de gauche et de droite, respectivement. On utilise la distance du Diamond d pour les nombres flous triangulaires non symétriques définie par l'équation I.31 (cf. chapitre I).

II.5.1.1 Régression linéaire floue

Supposons que, compte tenu d'un ensemble de paires de données $(x_i, Y_i), i = 1, \dots, N$, où les entrées x_i sont des nombres nettes, et les sorties Y_i sont des nombres flous triangulaires symétriques.

Soit le vecteur de poids flous $W = (w_1, w_2, \dots, w_M)$, où, $w_j = (m_{w_j}, r_{L_{w_j}}, r_{R_{w_j}}), j = 1; \dots, M$, est appartient à l'ensemble des nombres flous triangulaires non symétriques. et aussi le terme de biais flou $B = (m_B, r_{L_B}, r_{R_B}) \in NFT$ nonsymétrique.

On considère maintenant le modèle de régression floue suivant :

$$\begin{aligned} f(x) &= \langle W, x \rangle + B \\ &= w_1 x_1 + w_2 x_2 + \dots + w_M x_M + B \end{aligned} \quad (\text{II.68})$$

On définit $\|W\|^2 = \|m_W\|^2 + \|m_W - r_{L_W}\|^2 + \|m_W + r_{R_W}\|^2$, où $m_W = (m_{W_1}, \dots, m_{W_M})$, $r_{L_W} = (r_{L_{W_1}}, \dots, r_{L_{W_M}})$ et $r_{R_W} = (r_{R_{W_1}}, \dots, r_{R_{W_M}})$. Puis, nous arrivons au problème d'optimisation convexe suivant :

$$\begin{aligned} & \underset{m_W, r_{L_W}, r_{R_W}, m_B, r_{L_B}, r_{R_B}, \xi_{li}, \xi_{li}^*}{\text{Minimise}} \quad \frac{1}{2} \|W\|^2 + C \sum_{l=1}^3 \sum_{i=1}^n (\xi_{li} + \xi_{li}^*), \\ \text{sojet à} & \left\{ \begin{aligned} & m_{Y_i} - (\langle m_W, x_i \rangle + m_B) \leq \varepsilon + \xi_{1i} \\ & (\langle m_W, x_i \rangle + m_B) - m_{Y_i} \leq \varepsilon + \xi_{1i}^* \\ & (m_{Y_i} - r_{L_{Y_i}}) - (\langle m_W, x_i \rangle + m_B - \langle r_{L_W}, x_i \rangle - r_{L_B}) \leq \varepsilon + \xi_{2i} \\ & (\langle m_W, x_i \rangle + m_B - \langle r_{L_W}, x_i \rangle - r_{L_B}) - (m_{Y_i} - r_{L_{Y_i}}) \leq \varepsilon + \xi_{2i}^* \\ & (m_{Y_i} + r_{R_{Y_i}}) - (\langle m_W, x_i \rangle + m_B + \langle r_{R_W}, x_i \rangle + r_{R_B}) \leq \varepsilon + \xi_{3i} \\ & (\langle m_W, x_i \rangle + m_B + \langle r_{R_W}, x_i \rangle + r_{R_B}) - (m_{Y_i} + r_{R_{Y_i}}) \leq \varepsilon + \xi_{3i}^* \\ & \xi_{li}, \xi_{li}^* \geq 0, l = 1, \dots, 3, i = 1, \dots, N, \end{aligned} \right. \quad (\text{II.69}) \end{aligned}$$

On utilise la méthode de Lagrange pour résoudre ce problème d'optimisation, nous dérivons l'équation de Lagrange par rapport à $m_W, r_{L_W}, r_{R_W}, m_B, r_{L_B}, r_{R_B}, \xi_{li}, \xi_{li}^*$, on obtient :

$$m_W = \sum_{i=1}^N (\alpha_{1i} - \alpha_{1i}^*) x_i \quad (\text{II.70})$$

$$r_{LW} = \sum_{i=1}^N [(\alpha_{1i} - \alpha_{1i}^*) - (\alpha_{2i} - \alpha_{2i}^*)] x_i \quad (\text{II.71})$$

$$r_{RW} = \sum_{i=1}^N [(\alpha_{3i} - \alpha_{3i}^*) - (\alpha_{1i} - \alpha_{1i}^*)] x_i \quad (\text{II.72})$$

où $\alpha_{li}, \alpha_{li}^*, l = 1, \dots, 3$, et $i = 1, \dots, N$ sont les multiplicateurs de Lagrange non négatif.

Les deux radius r_{LW}, r_{RW} doit être toujours définis positifs, $r_{LW}, r_{RW} \geq 0$.

On obtient le problème d'optimisation dual correspondant comme suit :

$$\text{Maximise}_{\alpha_i, \alpha_{li}^* \geq 0} \left\{ \begin{array}{l} -\frac{1}{2} \sum_{l=1}^3 \sum_{i,j=1}^N (\alpha_{li} - \alpha_{li}^*)(\alpha_{lj} - \alpha_{lj}^*) \langle x_i, x_j \rangle \\ -\varepsilon \sum_{l=1}^3 \sum_{i,j=1}^N (\alpha_{li} + \alpha_{li}^*) + \sum_{i=1}^N m_{Y_i} (\alpha_{1i} - \alpha_{1i}^*) \\ + \sum_{i=1}^N (m_{Y_i} - r_{L_{Y_i}}) (\alpha_{2i} - \alpha_{2i}^*) + \sum_{i=1}^N (m_{Y_i} + r_{R_{Y_i}}) (\alpha_{3i} - \alpha_{3i}^*) \end{array} \right. \quad (\text{II.73})$$

$$\text{sujet à} \left\{ \begin{array}{l} \sum_{i=1}^N (\alpha_{li} - \alpha_{li}^*) = 0, \quad l = 1, \dots, 3, \\ \alpha_{li}, \alpha_{li}^* \in [0, C], \quad l = 1, 2, 3, \quad i = 1, \dots, N \end{array} \right. \quad (\text{II.74})$$

on peut écrit notre modèle flou comme suit :

$$f(\mathbf{x}) = \langle W, \mathbf{x} \rangle + B = (\langle m_W, \mathbf{x} \rangle, \langle r_{LW}, \mathbf{x} \rangle, \langle r_{RW}, \mathbf{x} \rangle) + B \quad (\text{II.75})$$

ainsi,

$$f(\mathbf{x}) = B + \left(\sum_{i=1}^N (\alpha_{1i} - \alpha_{1i}^*) \langle x_i, \mathbf{x} \rangle, \sum_{i=1}^N [(\alpha_{1i} - \alpha_{1i}^*) - (\alpha_{2i} - \alpha_{2i}^*)] \langle x_i, \mathbf{x} \rangle, \sum_{i=1}^N [(\alpha_{3i} - \alpha_{3i}^*) - (\alpha_{1i} - \alpha_{1i}^*)] \langle x_i, \mathbf{x} \rangle \right) \quad (\text{II.76})$$

Maintenant il faut trouver m_B, r_{L_B} et r_{R_B} .

Par l'utilisation des conditions de KKT, on peut calculer comme suit m_B :

$$\left\{ \begin{array}{l} \alpha_{1i} (\varepsilon + \xi_{1i} - m_{Y_i} + \langle W, m_{x_i} \rangle + m_B) = 0, \\ \alpha_{1i}^* (\varepsilon + \xi_{1i}^* - m_{Y_i} - \langle W, m_{x_i} \rangle - m_B) = 0, \\ (C - \alpha_{1i}) \xi_{1i} = 0, \quad (C - \alpha_{1i}^*) \xi_{1i}^* = 0 \end{array} \right.$$

Donc,

$$\left\{ \begin{array}{l} m_B = m_{Y_i} - \langle W, m_{x_i} \rangle - \varepsilon, \quad \text{for } \alpha_{1i} \in (0, C) \\ m_B = m_{Y_i} - \langle W, m_{x_i} \rangle + \varepsilon, \quad \text{for } \alpha_{1i}^* \in (0, C) \end{array} \right. \quad (\text{II.77})$$

Pour trouver r_{L_B} et r_{R_B} , nous avons besoin de résoudre le problème d'optimisation donné ci-dessous

$$\underset{r_{L_B}, r_{R_B} \geq 0}{\text{Minimise}} \begin{cases} \sum_{i=1}^N \left| m_{Y_i} - r_{L_{Y_i}} - \langle m_W - r_{L_W}, x_i \rangle - m_B + r_{L_B} \right|_{\varepsilon} \\ + \sum_{i=1}^N \left| m_{Y_i} + r_{R_{Y_i}} - \langle m_W + r_{R_W}, x_i \rangle - m_B - r_{R_B} \right|_{\varepsilon} \end{cases} \quad (\text{II.78})$$

où la fonction de perte ε -insensitive est définie par l'équation (II.17) dans la page 37.

II.5.1.2 Modèle de régression floue non linéaire

Pour étendre le modèle linéaire FSVR à l'approche non linéaire FSVR, nous utilisons ici la même idée de la fonction noyau (Kernel), qui est détaillée dans la section II.3.2 (cf. la page 49).

Ainsi, en remplaçant $\langle x_i, x_j \rangle$ par $K(x_i, x_j)$ dans le problème d'optimisation dual (II.73), nous obtenons le problème d'optimisation dual suivant :

$$\underset{\alpha_{li}, \alpha_{li}^* \geq 0}{\text{Maximise}} \begin{cases} -\frac{1}{2} \sum_{l=1}^3 \sum_{i,j=1}^N (\alpha_{li} - \alpha_{li}^*)(\alpha_{lj} - \alpha_{lj}^*) K(x_i, x_j) \\ -\varepsilon \sum_{l=1}^3 \sum_{i,j=1}^N (\alpha_{li} + \alpha_{li}^*) + \sum_{i=1}^N m_{Y_i} (\alpha_{1i} - \alpha_{1i}^*) \\ + \sum_{i=1}^N (m_{Y_i} - r_{L_{Y_i}}) (\alpha_{2i} - \alpha_{2i}^*) + \sum_{i=1}^N (m_{Y_i} + r_{R_{Y_i}}) (\alpha_{3i} - \alpha_{3i}^*) \end{cases} \quad (\text{II.79})$$

Dans ce cas, les contraintes sont les mêmes que dans le cas linéaire, comme suit :

$$\begin{cases} \sum_{i=1}^N (\alpha_{li} - \alpha_{li}^*) = 0, \quad l = 1, \dots, 3, \\ \alpha_{li}, \alpha_{li} \in [0, C], \quad l = 1, 2, 3, \quad i = 1, \dots, N \end{cases} \quad (\text{II.80})$$

Alors, on obtient le modèle de régression floue non linéaire suivant :

$$f(x) = B + \left(\sum_{i=1}^N (\alpha_{1i} - \alpha_{1i}^*) K(x_i, x), \sum_{i=1}^N [(\alpha_{1i} - \alpha_{1i}^*) - (\alpha_{2i} - \alpha_{2i}^*)] K(x_i, x), \sum_{i=1}^N [(\alpha_{3i} - \alpha_{3i}^*) - (\alpha_{1i} - \alpha_{1i}^*)] K(x_i, x) \right) \quad (\text{II.81})$$

Maintenant nous avons besoin de calculer $B = (m_B, r_{L_B}, r_{R_B})$. En utilisant les mêmes étapes que dans le cas de régression linéaire, d'abord en appliquant les conditions KKT, on peut calculer m_B comme suit :

$$\begin{cases} m_B = m_{Y_i} - \sum_{j=1}^N (\alpha_{1j} - \alpha_{1j}^*) K(x_j, x_i) - \varepsilon, & \text{for } \alpha_{1i} \in (0, C) \\ m_B = m_{Y_i} - \sum_{j=1}^N (\alpha_{1j} - \alpha_{1j}^*) K(x_j, x_i) + \varepsilon, & \text{for } \alpha_{1i}^* \in (0, C) \end{cases} \quad (\text{II.82})$$

Par la résolution du problème d'optimisation ci-dessous, on obtient r_{L_B} et r_{R_B} :

$$\underset{r_{L_B}, r_{R_B} \geq 0}{\text{Minimise}} \begin{cases} \sum_{i=1}^N \left| m_{Y_i} - r_{L_{Y_i}} - \sum_{j=1}^N (\alpha_{2j} - \alpha_{2j}^*) K(x_j, x_i) - m_B + r_{L_B} \right|_{\varepsilon} \\ + \sum_{i=1}^N \left| m_{Y_i} + r_{R_{Y_i}} - \sum_{j=1}^N (\alpha_{3j} - \alpha_{3j}^*) K(x_j, x_i) - m_B - r_{R_B} \right|_{\varepsilon} \end{cases} \quad (\text{II.83})$$

Dans cette approche, nous représentons les modèles RBF de régression floue non linéaire, en d'autres termes, nous utilisons un noyau RBF pour nos modèles de régression non linéaire floues.

Un noyau RBF est défini comme suit :

$$K(x, y) = \exp(-\|x - y\|^2 / 2 \cdot \sigma^2)$$

II.5.2 PSO et UPSO pour l'optimisation des hyper-paramètres

II.5.2.1 Algorithme standard d'optimisation par essaim de particules (PSO)

L'optimisation par essaim de particules est une technique évolutionnaire qui utilise "une population" de solutions candidates pour développer une solution optimale au problème. Le degré d'optimalité est mesuré par une fonction fitness (aptitude) définie par l'utilisateur (Kennedy et al. [12],[21]). La PSO diffère des autres méthodes de calcul évolutionnaire de façon que les membres de la population appelés "particules", sont dispersées dans l'espace du problème, [25].

Le comportement de l'essaim doit être décrit en se plaçant du point de vue d'une particule (Kennedy et al. [12],[21]).

Au départ de l'algorithme, un essaim est réparti au hasard dans l'espace de recherche, chaque particule ayant également une vitesse aléatoire.

Dans le contexte de PSO, la population est appelée un essaim et ses individus (les points de recherche) sont appelées particules. Chaque particule a trois caractéristiques principales : une vitesse adaptable avec laquelle elle se déplace dans l'espace de recherche, une mémoire où elle stocke la meilleure position qu'elle n'a jamais visité l'espace de recherche (la position de plus faible valeur de fonction), et le social partage de l'information, à savoir la connaissance de la meilleure position jamais visité par toutes les particules dans son voisinage. Les voisins sont généralement déterminés sur la base des indices des particules, donnant lieu à des deux principales variantes de PSO, à savoir la variante locale et globale.

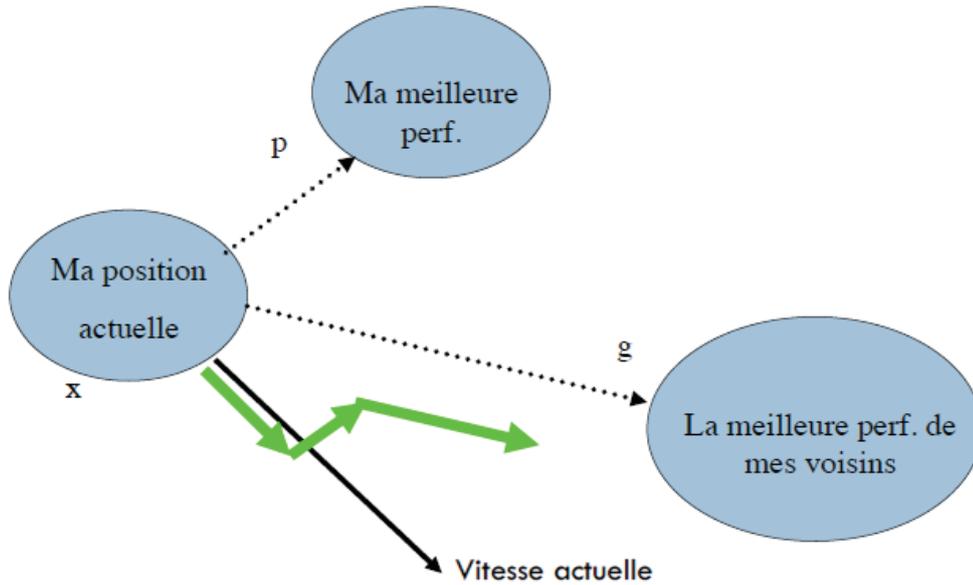


FIG. II.12 – Schéma de principe du déplacement d'une particule, [22]

A partir de quelques informations dont elle dispose, une particule doit décider de son prochain mouvement. Pour ce faire, elle combine trois informations : sa vitesse actuelle, sa meilleure position actuelle, et la meilleure performance (vitesse et position) de ses voisins (Figure II.12).

Le hasard joue un rôle, grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l'exploration de l'espace de recherche.

Supposons une fonction à n dimensions $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$, et un essaim $S = \{X_1, X_2, \dots, X_N\}$, de N particules. La $i^{\text{ème}}$ particule $X_i \in S$, sa vitesse V_i , ainsi que sa meilleure position $P_i \in S$, sont des vecteurs de n -dimensions. Un voisinage de rayon r_{neib} de X_i consiste en les particules $X_i - r_{neib}, \dots, X_i, \dots, X_i + r_{neib}$. Assumer b_i comme indice de la particule qui atteint la meilleure position précédente parmi toutes les particules dans le voisinage de X_i , et t est le compteur d'itération. Puis, selon la version de PSO par les coefficients de construction, l'essaim est mis à jour en utilisant les équations suivantes, [7],[25] :

$$V_i(t+1) = \chi [V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (P_{b_i}(t) - X_i(t))] \quad (\text{II.84})$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (\text{II.85})$$

où $i = 1, 2, \dots, N$, χ est le coefficient de constriction, c_1 et c_2 sont des constantes positives, appelées paramètres cognitifs et sociaux, respectivement, r_1 et r_2 sont des vecteurs aléatoires avec les composants uniformément réparties dans $[0, 1]$. Les valeurs par défaut de χ , c_1 et c_2 sont déterminés dans l'analyse théorique de Clerc et Kennedy [7].

La performance majeure d'un algorithme basé sur la population est fortement tributaire de l'arbitrage entre l'exploration et l'exploitation des capacités, c'est à dire, sa capacité à explorer de vastes zones de l'espace de recherche et sa capacité à converger rapidement vers les solutions les plus prometteuses, respectivement. La variante globale de PSO favorise l'exploitation car toutes les particules sont attirées par la meilleure position entraînant une convergence rapide vers le même point. . D'autre part, la variante locale possède des propriétés d'exploration meilleures par contre la meilleure position de chaque voisinage est communiquée au reste de l'essaim par des particules voisines. Par conséquent, l'attraction à des points spécifiques est plus faible.

Évidemment, la sélection appropriée de la taille du voisinage affecte le compromis entre exploration et exploitation. Toutefois, la sélection de la taille du voisinage est fortement basée sur l'expérience de l'utilisateur, [25].

II.5.2.2 L'optimisation par essaim particulaire unifiée (UPSO)

La technique de l'optimisation par essaim particulaire unifiée (UPSO) a été récemment proposé comme une alternative qui combine les propriétés d'exploration et d'exploitation des deux PSO variantes, locale et globale, [7],[25]. Le mécanisme présenté est basé sur la version du facteur de constriction de PSO, même s'il peut être directement défini aussi pour la version d'inertie. Soit $G_i(t + 1)$ et $L_i(t + 1)$ dénotent la mise à jour de vitesse de la $i^{\text{ème}}$ particule, pour la variante globale et locale de PSO, respectivement [7],[22],[25].

$$G_i(t + 1) = V_i(t) + c_1 r_1 (P_i(t) - X_i(t)) + c_2 r_2 (P_g(t) - X_i(t)) \quad (\text{II.86})$$

$$L_i(t + 1) = V_i(t) + c_1 r'_1 (P_i(t) - X_i(t)) + c_2 r'_2 (P_{g_i}(t) - X_i(t)) \quad (\text{II.87})$$

où t désigne l'itération courante, g est l'indice de la meilleure particule de l'essaim (variante globale), et g_i est l'indice du meilleur particules dans le voisinage de X_i (Variante locale). Les directions de recherche définies par les équations II.86 et II.87 sont regroupés dans une seule équation, résultant le mécanisme principal de UPSO [7],[25].

$$V_i(t + 1) = (1 - u) .G_i(t + 1) + u.L_i(t + 1) \quad (\text{II.88})$$

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (\text{II.89})$$

où le paramètre u est le facteur d'unification. Ce facteur équilibre l'influence des directions de recherche globale et locale dans l'algorithme final, le cas du standard PSO global est obtenu en mettant $u = 1$ dans l'équation II.88, $u = 0$ correspond au cas de standard local PSO. Toutes les valeurs de $u \in (0, 1)$, correspondent à composer les variantes de PSO qui combinent les caractéristiques d'exploration et d'exploitation de sa variante locale et globale.

II.5.2.3 La sélection des hyper-paramètres par standard PSO et UPSO

En utilisant les algorithmes de standard PSO et UPSO présentés dans la section précédente, les hyper-paramètres du modèle FSVR, C et σ , peuvent être optimisés. Dans la résolution du problème de la sélection des hyper-paramètres, chaque particule représente une solution potentielle. L'optimalité des hyper-paramètres est mesurée par des moyens de fonction fitness qui est défini par rapport au problème d'optimisation considéré. Dans le processus d'apprentissage et de test de FSVR, l'objectif est d'améliorer les performances de généralisation du modèle de régression floue, à savoir, réduire au minimum les erreurs entre les valeurs réelles et les valeurs de prévision des échantillons d'apprentissages, [38]. Par conséquent, la fonction de remise en forme peut être définie comme suit :

$$Fitness = \frac{1}{v} \sum_{i=1}^v \sqrt{\frac{1}{m} \sum_{j=1}^m (m_{f(x_{ij})} - m_{Y_{ij}})^2} \quad (\text{II.90})$$

où, v est le nombre de plis pour la validation croisée de v -plis (v -folds), [24], et m le nombre de chaque sous-ensemble de validation, $m_{Y_{ij}}$ est le centre de la valeur de sortie floue observée, $m_{f(x_{ij})}$ est le midpoint du modèle floue estimée de toute les données d'apprentissage.

L'objectif est de minimiser le fitness, de sorte que la particule avec la valeur minimale de fitness est mieux que les autres et doivent être réservés au cours du processus d'optimisation. Ainsi, les hyper-paramètres optimaux peuvent être sélectionnés.

II.5.3 L'algorithme proposé et les étapes de l'identification

Dans cette section, nous décrivons la procédure de notre technique d'identification. Les étapes de l'algorithme de UPSO-FSVR sont énumérés ci-dessous (les étapes de standard PSO-FSVR sont les mêmes que UPSO-FSVR) :

- **Étape 1** : L'optimisation des hyper-paramètres

Le processus d'UPSO pour l'optimisation des hyper-paramètres peut être décrit comme suit :

Étape 1.1 : Initialisation de la taille de l'essaim, le nombre maximal de générations, la vitesse et la position de chaque particule composées de l'hyper-paramètres C et σ . La vitesse est limitée dans le domaine $[-V_{\max}, V_{\max}]$, où V_{\max} est la pondération de la vitesse prédéfinie à partir des données expérimentales.

Étape 1.2 : Évaluation du fitness de chaque particule selon la formule (II.90), et régler la meilleure position de la particule avec le minimum fitness dans l'essaim.

Étape 1.3 : Pour chaque particule candidat, lancer un modèle FSVR avec les hyper-paramètres correspondants basés sur la validation croisée.

Étape 1.4 : mettre à jour la vitesse de chaque particule, pour la variante locale et globale du PSO par les équations (II.86) et (II.87), puis mettre à jour la vitesse et la position de chaque particule à l'aide des équations (II.88), (II.89), respectivement.

Étape 1.5 : Vérifier le critère d'arrêt. Si le nombre maximal de générations n'est pas encore atteint, retourner à l'étape 1.2, sinon, passer à la sous-étape suivante.

Étape 1.6 : Terminer l'algorithme et donner les hyper-paramètres optimaux.

– **Étape 2** : Apprentissages et tests.

Lancer un estimateur FSVR sur les échantillons d'entraînement avec les hyper-paramètres optimaux obtenus à partir de l'étape 1

Pour valider le modèle de régression flou obtenu, nous pouvons utiliser des mesures pour l'évaluer. L'organigramme de l'approche proposée est montré dans la figure II.13.

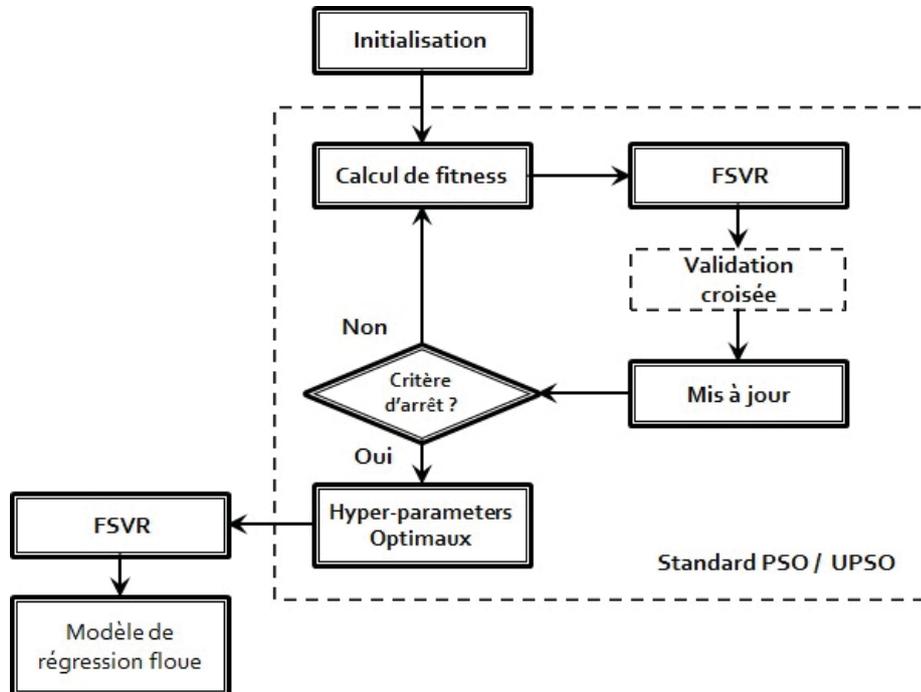


FIG. II.13 – Diagramme de l'algorithme proposé

II.5.4 Exemples illustratifs

Dans cette section, nous utilisons deux ensembles de données de type entrée nette-sortie floue, le premier ensemble (exemple 1) est pris à partir de Gunn [13] et Hong [15], et le deuxième (exemple 2) est prise à partir de Tanaka et Lee [31], Ces deux exemples sont présentés dans les tableaux (II.2) et (I.1), respectivement.

j	x	Y
1	1.0	(-1.6, 0.5)
2	3.0	(-1.8, 0.5)
3	4.0	(-1.0, 0.5)
4	5.6	(1.2, 0.5)
5	7.8	(2.2, 1.0)
6	10.2	(6.8, 1.0)
7	11.0	(10.0, 1.0)
8	11.5	(10.0, 1.0)
9	12.7	(10.0, 1.0)

TAB. II.2 – Le jeu de données observées

L’approche proposée de UPSO-FSVR est appliquée à ces ensembles de données à partir d’un point de vue illustratif.

D’abord, nous mettons en application l’algorithme d’UPSO pour trouver les hyperparamètres optimaux C et σ , pour chaque exemple à l’aide de la validation croisée de *5-plis*. La taille de la population initiale de l’essaim est choisie 40 telle qu’elle est suffisante pour couvrir l’espace de recherche au sein des générations limitées sur la base des essais expérimentaux. Basé sur beaucoup d’expériences, le nombre maximal des générations est fixé à 200, et le facteur d’unification est choisi $u = 0,5$, et le coefficient de constriction est fixé à 0,729, les deux paramètres cognitif et social ont la même valeur, qui est 1.05, la vitesse maximale est calculés avec $v_{\max} = (x_{\max} - x_{\min})/2$.

Et pour la fonction de perte ϵ -insensitive, on choisit $\epsilon = 0.1$.

Maintenant, nous considérons une mesure de précision, MAPE (Mean absolute percentage error) pour évaluer et comparer nos méthodes proposées UPSO-FSVR et standard PSO-FSVR.

La mesure de la précision pour les échantillons de données est définie comme,

Il est également intéressant de quantifier l’erreur relative entre les valeurs modales des

observations et des prédictions. Dans le cas de l'approche triangulaire, elle est définie par :

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{m_{f(x_i)} - m_{y_i}}{m_{y_i}} \right| \times 100 \quad (\text{II.91})$$

où m_{y_i} est le midpoint de la sortie floue vraie, $m_{f(x_i)}$ est le midpoint du modèle de régression floue estimée.

Dans le cas de l'approche trapézoïdale, elle est définie en ne conservant que les midpoints des intervalles supports, soit :

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{m_{S_{f(x_i)}} - m_{y_i}}{m_{y_i}} \right| \times 100 \quad (\text{II.92})$$

avec $m_{S_{f(x_i)}}$ est la valeur modale du support de la sortie floue estimée (prédiction), et m_{y_i} est le midpoint de la sortie observée.

On considère aussi la mesure de la somme des résidus pour évaluer les résultats. La somme des résidus pour un modèle flou triangulaire non symétrique est définie comme suit, [10] :

$$SUM_{res} = \sum_{i=1}^N \left[\left(m_{f(x_i)} - m_{y_i} \right)^2 + \left(\left(m_{f(x_i)} - r_{L_{f(x_i)}} \right) - (m_{y_i} - r_{y_i}) \right)^2 + \left(\left(m_{f(x_i)} + r_{R_{f(x_i)}} \right) - (m_{y_i} + r_{y_i}) \right)^2 \right] \quad (\text{II.93})$$

où, $r_{L_{f(x_i)}}$ et $r_{R_{f(x_i)}}$ sont les radius gauche et droite de modèle floue obtenu, respectivement.

et pour l'approche trapézoïdale la somme des résidus est définie par :

$$SUM_{res} = \sum_{i=1}^N \left[\left(S_{f(x_i)}^- - (m_{y_i} - r_{y_i}) \right)^2 + \left(K_{f(x_i)}^- - m_{y_i} \right)^2 + \left(K_{f(x_i)}^+ - m_{y_i} \right)^2 + \left(S_{f(x_i)}^+ - (m_{y_i} + r_{y_i}) \right)^2 \right] \quad (\text{II.94})$$

où, $S_{f(x_i)}^-$, $S_{f(x_i)}^+$ et $K_{f(x_i)}^-$, $K_{f(x_i)}^+$, sont les bornes inférieur et supérieur du support et du noyau de modèle flou trapézoïdal, respectivement.

Cas	Hyper-paramètres optimaux				SUM_{res}		MAPE(%)	
	par <i>UPSO</i>		par standard <i>PSO</i>		UPSO-FSVR	PSO-FSVR	UPSO-FSVR	PSO-FSVR
	<i>C</i>	σ	<i>C</i>	σ				
<i>Exemple1</i>	475.0687	0.8427	500.00	0.0148	0.4925	0.6247	0.4351	0.4351
<i>Exemple2</i>	500.00	0.1526	500.00	0.0834	0.0026	1.0469	0.2649	0.2649

TAB. II.3 – Performances d'identification et les hyper-paramètres optimaux pour les modèles des deux exemples

Le tableau II.3 montre l'hyper-paramètres obtenus C et σ , et les mesures SUM_{res} et $MAPE$, pour les deux méthodes proposées, standard PSO-FSVR et UPSO-FSVR. Ces résultats sont pour les deux exemples (1 et 2).

Nous pouvons observer que les valeurs de SUM_{res} et $MAPE$ dans le modèle UPSO-FSVR sont meilleures que ceux du modèle de standard PSO-FSVR, pour les deux ensembles de données de l'exemples 1 et 2. Ainsi, le modèle de UPSO-FSVR offre la meilleure compatibilité et la minimale somme de résidus aux données observées de l'exemple 1 par rapport au modèle original de FSVR, par Hong [15] ($SUM_{res} = 0.833$), et aussi par rapport au modèle standard PSO-FSVR. Généralement la méthode proposée d'UPSO-FSVR fonctionne très bien.

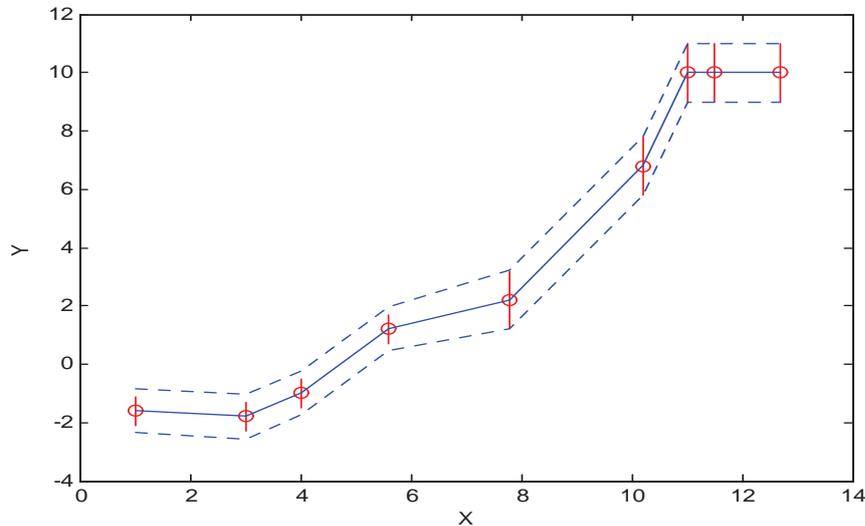


FIG. II.14 – Modèle de régression floue non linéaire obtenue par UPSO-FSVR pour l'exemple 1

Les figures (II.14) et (II.15) illustrent les résultats de l'exemple 1 pour les approches UPSO-FSVR, et standard PSO-FSVR, respectivement. Comme dans les figures (II.14)-(II.17), la ligne continue explique la ligne de régression ajustée pour le centre, tandis que les deux lignes en pointillées expliquent les radius de gauche et de droite (rayon) du modèle flou, respectivement.

Par contre les figures (II.16). et (II.17) illustrent les résultats de l'exemple 2 pour les deux approches proposées UPSO-FSVR, et standard PSO-FSVR, respectivement. On constate d'après ces deux figures que l'approche proposée pour les modèles de la régression flous non linéaire fonctionne assez bien.

Une propriété clé de l'algorithme proposé est que l'optimisation par essaim de particules unifiées combine les caractéristiques d'exploration et d'exploitation de sa solution globale et

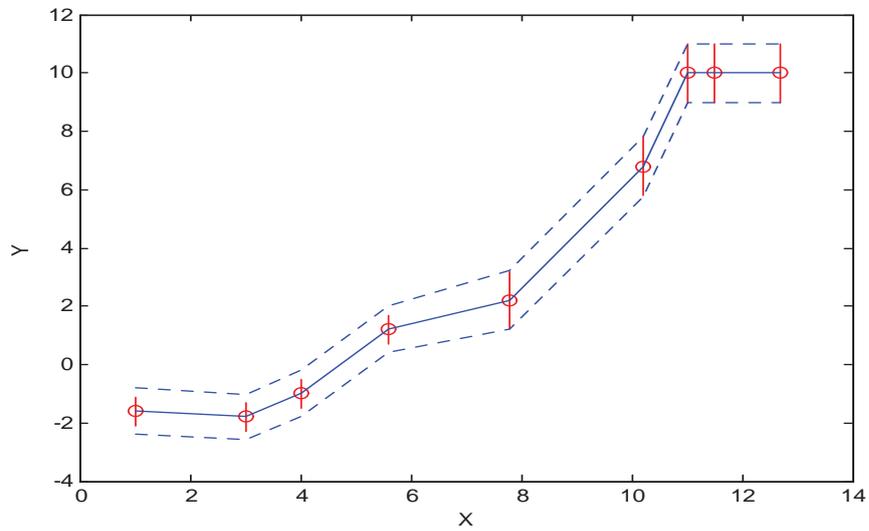


FIG. II.15 – Modèle de régression floue non linéaire obtenue par standard PSO-FSVR pour l'exemple1

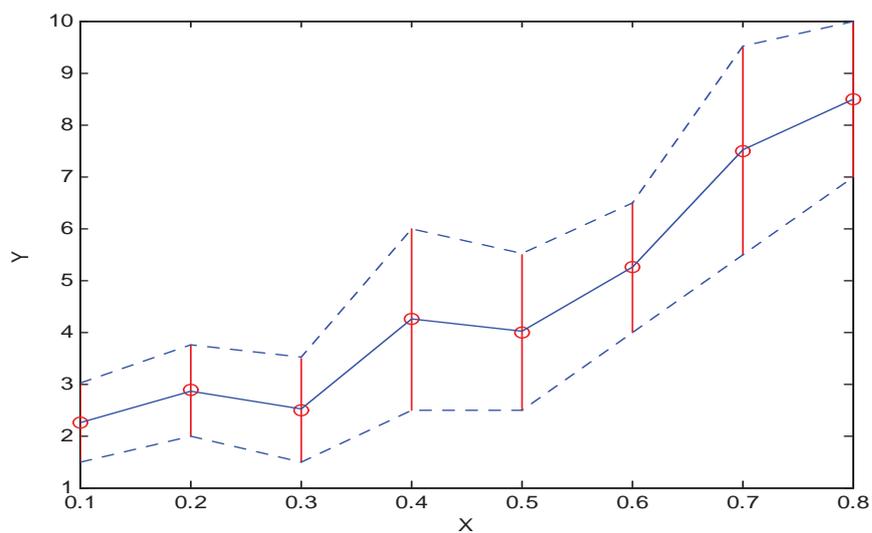


FIG. II.16 – Modèle de régression floue non linéaire obtenue par UPSO-FSVR pour l'exemple2

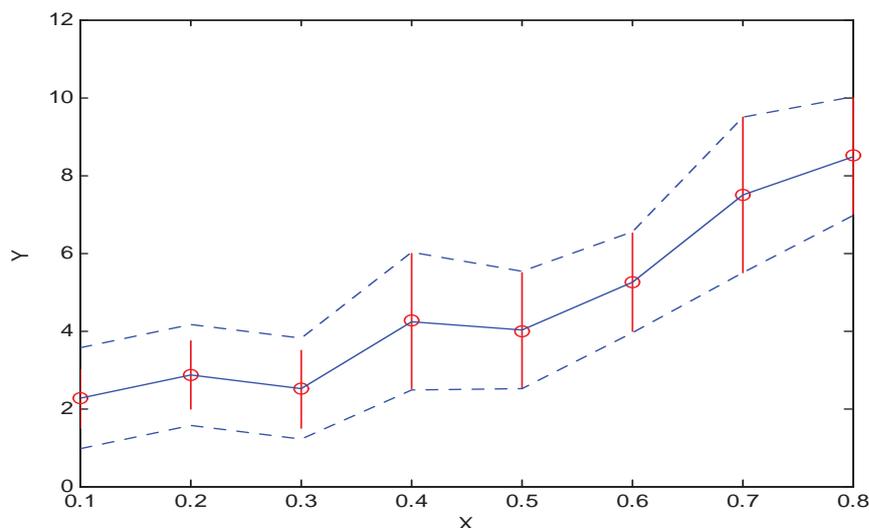


FIG. II.17 – Modèle de régression floue non linéaire obtenue par standard PSO-FSVR pour l'exemple2

locale.

II.6 Conclusion

Dans le chapitre précédent plusieurs limites de la régression floue linéaire et non linéaire ont été soulignées dans l'approche de Tanaka. Dans cette partie, nous avons présenté la régression floue à vecteur de support (FSVR), elle est pratique dans le cas non linéaire. Une nouvelle contribution a ainsi été développée.

Cette proposition concerne la formulation d'un critère d'identification utilisant FSVR avec un noyau RBF dont les hyper-paramètres sont sélectionnés à partir de l'optimisation par essaim particulaire unifiée (UPSO).

Cette approche initialement présentée pour des modèles simples ne comportant qu'une unique variable a été généralisée pour des modèles comportant plusieurs entrées.

Chapitre III

Régression floue : Application

III.1 Introduction

Dans ce chapitre, nous présentons quelques résultats de simulation numérique, sur des exemples fréquemment utilisés dans la littérature de la régression floue. Les méthodes développées dans les chapitres précédents sont analysées à travers une série de tests. Ces tests visent notamment à évaluer les performances de ces méthodes.

Ici, la problématique liée aux données multi-entrées sera abordée, permettant ainsi de mettre en évidence l'intérêt de l'identification de modèles de structure complexe. L'exemple traité permettra également d'éprouver la qualité des modèles obtenus lors d'une phase de validation sur des données de test, selon une procédure exploitée en apprentissage.

III.2 Identification de modèles multi-entrées

Dans cette section, une étude expérimentale, à travers un système multi-entrées mono-sortie donné dans la littérature [2],[6], sera menée pour mettre en évidence l'influence du choix de la structure du modèle sur la qualité de l'identification.

Dans cette application, le jeu de données, composé de 100 échantillons, présenté dans [2],[6] est considéré (cf. annexe A). La taille du modèle est supposée connue. En d'autres termes, le nombre des entrées est disponible ou a été déterminé a priori. L'objectif est alors d'identifier à l'aide de notre méthode un modèle régressif le plus représentatif de la relation entrées-sortie entre les cinq variables d'entrée x_1, \dots, x_5 et la variable de sortie y .

Le protocole de génération de ces données est basé sur l'utilisation d'un modèle mathématique précis que nous supposons ici inconnu. Certains échantillons de sortie (au nombre de 10, dénommés points aberrants (outliers) dans l'annexe A) sont bruités, afin d'étudier l'impact de ceux-ci sur le modèle identifié. Le bruit additif est une valeur aléatoire de distribution normale dont l'amplitude est comprise entre 0 et 2.

D'une manière générale et comme supposé dans la plupart des méthodes d'identification, le modèle obtenu est validé sur un ensemble de test. Cet ensemble peut être généré à partir des données aléatoires lorsque le système le permet. Dans d'autres situations, les données disponibles auront été divisées au préalable en deux sous-ensembles, un ensemble d'apprentissage et un ensemble de test. C'est cette approche qui sera exploitée dans cette application. En effet, sur les 100 échantillons disponibles, seuls 90 (dont les 10 bruités) sont utilisés pour l'identification, 10 étant conservés comme données de test et de validation. mais s'il y a

un nombre très réduit de données, il est possible de faire appel à des outils de validation statistiques.

Dans un premier temps un système linéaire de la forme :

$$\hat{Y} = f(x) = B \oplus \sum_{i=1}^N W_i \cdot (x_i - shift_i) \quad (III.1)$$

D'après les données de l'annexe A, il est clair que toutes les entrées sont comprises entre 0 et 1. Par conséquent, dans un premier temps, les décalages retenus sont définis par $shift_i = 0$, $i = \{1, 2, 3, 5\}$ et $shift_4 = 1$. [2]

Nous appliquons maintenant notre approche proposée PSO/UPSO-FSVR à cet ensemble de données. Premièrement, nous considérons un modèle floue de la forme (III.1) (modèle 1) et nous utilisons le noyau RBF pour mapper les données d'entrée dans l'espace caractéristique.

Le modèle obtenu (Modèle1) est illustré dans les figures III.1 et III.2.

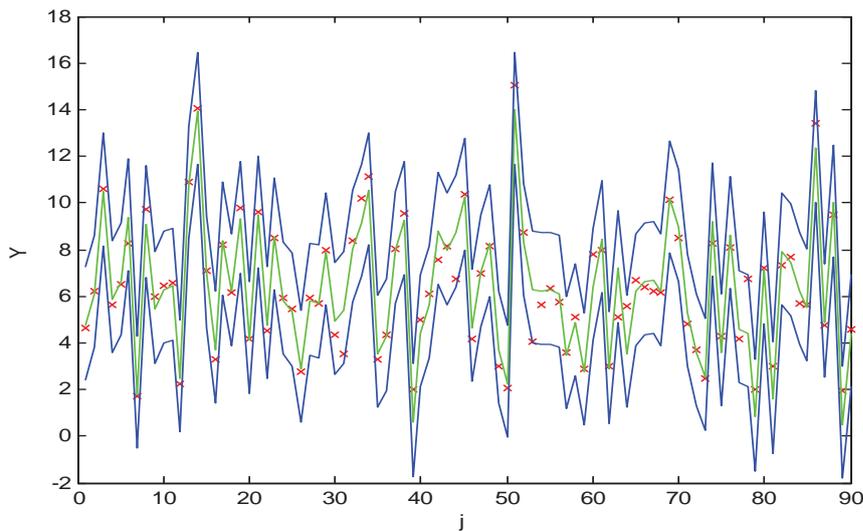


FIG. III.1 – Modèle multivariable obtenu par l'approche proposée (Modèle1)

Maintenant on passe à la tâche de validation de ce modèle obtenu utilisant les données de test (cf. annexe A).

Pour ce faire, il faut calculer la mesure $MAPE$ (équation II.91), et $MAPE_j$ (c'est formules sont pour les modèles flous triangulaire) comme suit :

$$MAPE_j = \left| \frac{m_{f(x_j)} - m_{y_j}}{m_{y_j}} \right| \times 100 \quad (III.2)$$

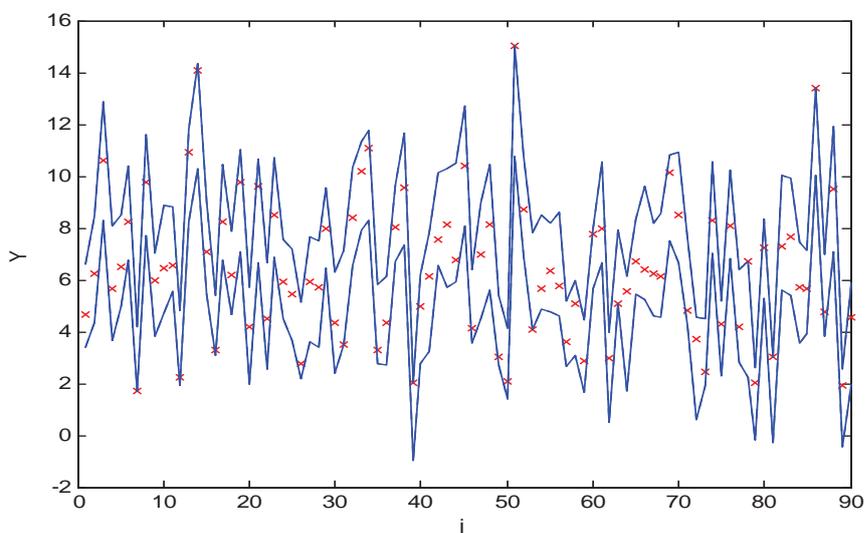


FIG. III.2 – Modèle obtenu par l'approche linéaire (Modèle1)

j	y	$m_{f(x)}$	$MAPE_j$ (%)	\hat{Y}
1	1.3915	1.4034	0.8577	[-0.7966, 3.8034]
5	1.1886	0.5755	51.5825	[-1.6245, 2.9755]
26	1.7484	1.7666	1.0395	[-0.4334, 4.1666]
32	2.3165	1.7800	1.0395	[-0.4200, 4.1800]
33	4.0561	3.7710	7.0282	[1.5710, 6.1710]
34	5.1463	5.6657	10.0919	[3.4657, 8.0657]
41	2.2748	1.7938	21.1427	[-0.4062, 4.1938]
67	1.4327	0.2268	84.1699	[-1.9732, 2.6268]
91	2.896	5.3393	84.3687	[3.1393, 7.7393]
93	2.7117	2.6410	2.6071	[0.4410, 5.0410]
$MAPE$ (données de test)			28.6049	
$MAPE$ (données d'identification)			10.3451	

TAB. III.1 – Test sur les données de validation pour le modèle1 par l'approche proposée UPSO-FSVR

Dans le tableau III.1, on a rassemblé la mesure de $MAPE_j$ pour chaque donnée de test, et aussi l'erreur $MAPE$ de toute les données de test, et pour toute les données d'identification.

Pour les modèles trapézoïdaux, on peut calculer la mesure $MAPE$ à partir de l'équation II.92, et $MAPE_j$ comme suit :

$$MAPE_j = \left| \frac{m_{S_f(x_j)} - m_{y_j}}{m_{y_j}} \right| \times 100 \quad (\text{III.3})$$

j	y	$m_{S_{\hat{y}}}$	$MAPE_j$ (%)	\hat{Y}
1	1.3915	1.5300	9.9503	[0.2675, 1.8734]
5	1.1886	-0.0340	102.8584	[-1.9941, 0.4201]
26	1.7484	2.2275	27.4029	[1.6781, 2.9302]
32	2.3165	2.0436	11.7818	[0.5660, 2.9745]
33	4.0561	3.7446	7.6806	[3.0730, 5.3822]
34	5.1463	5.6391	9.5763	[3.8384, 5.7367]
41	2.2748	2.7977	22.9880	[1.4329, 3.4925]
67	1.4327	0.4190	70.7555	[-1.3283, 1.3375]
91	2.896	5.1179	76.7243	[3.8305, 5.4942]
93	2.7117	2.4326	10.2926	[1.5477, 3.5363]
$MAPE$ (données de test)			35.0011	
$MAPE$ (données d'identification)			14.3083	

TAB. III.2 – Test sur les données de validation pour le modèle1 par la technique linéaire

Il est possible de comparer les résultats de validation de notre modèle avec ceux présentés dans [2], pour un modèle trapézoïdal a entrées décalées.

En ce qui concerne l'inclusion des sorties observées dans les intervalles prédits, on voit dans le tableau III.2 que sur les 10 données de validation, juste une ne respecte pas cette inclusion (indice $j = 91$). Par contre pour la méthode linéaire (LP-FR [2]), on trouve 3 échantillons (indices $j = 5, 67, 91$) de test qui ne respectent pas l'inclusion. Cela montrant ainsi que notre méthode est meilleure que la technique linéaire présentée dans [2]. Le modèle identifié, de par sa structure non adaptée, échoue à représenter de manière correcte les données d'identification, et par conséquent à être performant sur les tests de validation.

La deuxième phase de l'étude menée sur cet exemple consiste à considérer un autre modèle à identifier, c'est-à-dire, un modèle dont les entrées ne sont plus nécessairement x_1, \dots, x_5 .

Pratiquement, le choix de la structure est fait par des considérations empiriques qui consistent à essayer plusieurs structures et n'accepter que celle qui donne la meilleure performance.

Ainsi, il est possible d'incorporer une technique d'identification de la structure du modèle en amont de l'utilisation de notre méthode.

Le modèle à identifier est donc de la forme [6] :

$$\hat{Y} = f(x) = B + W_1.(x_1.x_2) \oplus W_2.x_3 \oplus W_3.(x_3.x_4) \oplus W_4.(x_5^2) \quad (\text{III.4})$$

Les modèles obtenus à l'aide de notre méthode et par la méthode linéaire sont présentés sur les figures III.3 et III.4, respectivement.

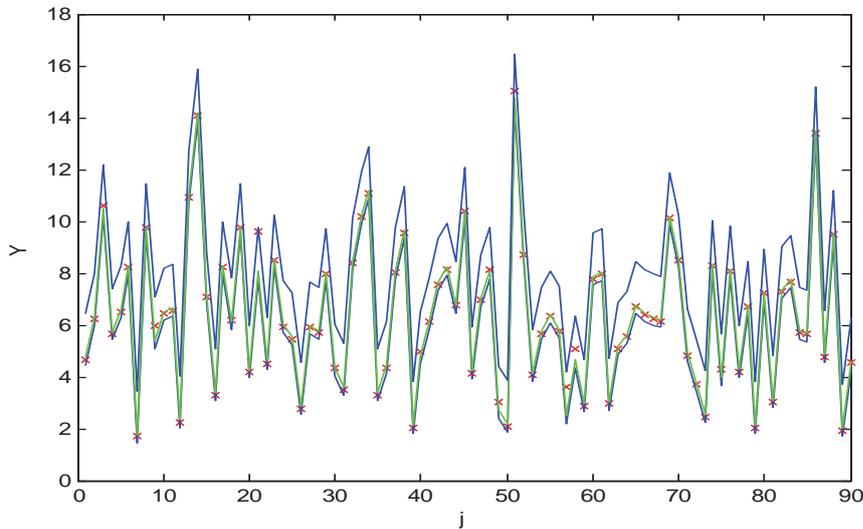


FIG. III.3 – Modèle multi entrées obtenu par l'approche proposée (Mdèle2)

On applique la même procédure de validation au 2^{ème} modèle (III.4). Les résultats sont présentés dans le tableau III.4.

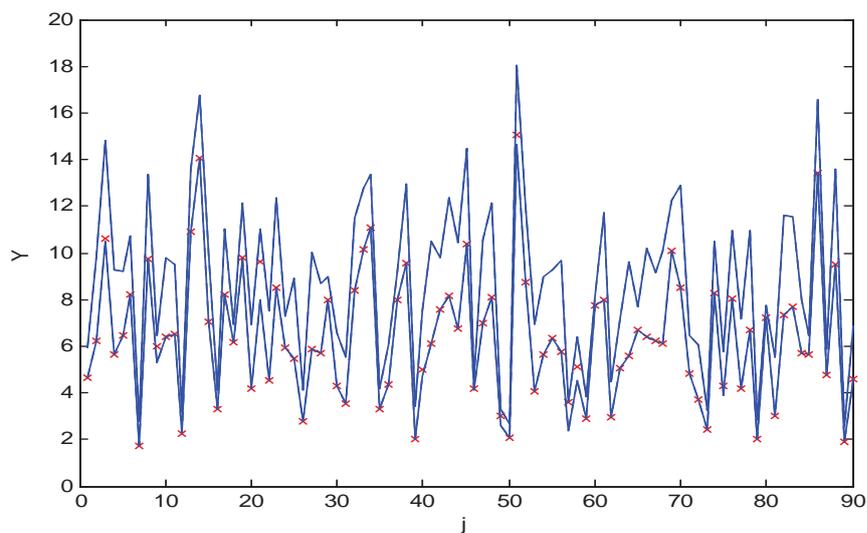


FIG. III.4 – Modèle multi entrées obtenu par la technique linéaire (Mdèle2)

j	y	$m_{f(x)}$	$MAPE_j$ (%)	\hat{Y}
1	1.3915	1.5056	8.1966	[1.2931, 3.0912]
5	1.1886	1.1868	0.1502	[0.9743, 2.7725]
26	1.7484	1.8671	6.7913	[1.6547, 3.4528]
32	2.3165	2.4218	4.5441	[2.2093, 4.0074]
33	4.0561	4.1455	2.2042	[3.9330, 5.7312]
34	5.1463	5.2071	1.1813	[4.9946, 6.7928]
41	2.2748	2.3787	4.5672	[2.1662, 3.9644]
67	1.4327	1.5369	7.2752	[1.3245, 3.1226]
91	2.896	3.0087	3.8930	[2.7963, 4.5944]
93	2.7117	2.8255	4.1963	[2.6130, 4.4111]
$MAPE$ (données de test)			4.30	
$MAPE$ (données d'identification)			2.3853	

TAB. III.3 – Test sur les données de validation pour le deuxième modèle par l'approche proposée UPSO-FSVR

j	y	$m_{S_{\hat{y}}}$	$MAPE_j$ (%)	\hat{Y}
1	1.3915	1.4240	2.3379	[1.3911, 1.4570]
5	1.1886	1.2257	3.1253	[1.1872, 1.2643]
26	1.7484	1.7769	1.6327	[1.7483, 1.8055]
32	2.3165	2.4808	7.0933	[2.3159, 2.6457]
33	4.0561	4.2129	3.8653	[4.0561, 4.3697]
34	5.1463	5.1715	0.4900	[5.1450, 5.1980]
41	2.2748	2.4392	7.2264	[2.2744, 2.6040]
67	1.4327	1.5108	5.4512	[1.4318, 1.5898]
91	2.896	2.9482	1.8026	[2.8955, 3.0009]
93	2.7117	2.7713	2.1976	[2.7117, 2.8309]
$MAPE$ (données de test)			3.5223	
$MAPE$ (données d'identification)			20.1699	

TAB. III.4 – Test sur les données de validation pour le deuxième modèle

On remarquera ici que l'inclusion est respectée pour les 10 données de test, et aussi que la mesure MAPE moyenne est très faible par rapport au premier modèle (III.1), pour l'approche proposée et pour la technique linéaire.

Cas	$Hyper\text{-}paramètres$ optimaux				$MAPE(\%)$		$MAPE(\%)$	
	par $UPSO$		par standard PSO		Identification		Test	
	C	σ	C	σ	UPSO-FSVR	PSO-FSVR	UPSO-FSVR	PSO-FSVR
$Modèle1$	1601.4	0.5184	4074.6	0.6025	10.3451	10.3636	28.6049	29.2138
$Modèle2$	3730.9	0.6874	3993.6	3.0897	2.3853	2.6402	4.300	5.1118

TAB. III.5 – Performances d'identification et les hyper-paramètres optimaux pour les deux modèle multi-entrées

D'après le tableau III.5, nous pouvons constater que notre approche proposée de prévision nous donne des résultats prometteuses pour le cas des données réels bruitées multi entrée - mono sortie (MISO), pour des modèles différents. La valeur de la mesure MAPE (Test) est petite pour les deux modèles, mais la valeur de l'erreur MAPE (erreur entre la sortie observée et la sortie prédite) pour le second modèle (Modèle2) est plus petite.

Nous pouvons voir que l'approche UPSO-FSVR est meilleure que la méthode standard PSO-FSVR.

En ce qui concerne l'inclusion de la sortie observée dans celle estimée, elle est respectée pour les deux modèles, avec toutes les données d'identification. Par conséquent, pour les données de test, les sorties non bruitées sont parfaitement représentées par les bornes supérieure et inférieure de la sortie de notre modèle.

<i>Cas</i>	<i>MAPE</i> (%)		<i>MAPE</i> (%)		<i>SUM_{res}</i>	
	Identification		Test			
	UPSO- FSVR	LP- FR	UPSO- FSVR	LP- FR	UPSO- FSVR	LP- FR
<i>Modèle1</i>	10.3451	14.3083	28.6049	39.8200	1104.9	1413.3
<i>Modèle2</i>	2.3853	20.1699	4.300	3.5223	247.876	1343.8

TAB. III.6 – Performances d'identification et la somme des résidus pour les deux modèle multi-entrées

Le tableau III.6 présente une comparaison entre notre méthode de régression flou et la méthode linéaire de la régression floue représenté dans [2]. Premièrement, concernant l'erreur entre la sortie observée et la sortie prédite (*MAPE*) pour les données de test et pour les données d'identification, elle est minime pour les deux méthodes, mais notre approche est meilleure, surtout dans le cas des données d'identification (*MAPE* identification). Deuxièmement, on peut observer que la somme des résidus (*SUM_{res}*) pour notre approche est très inférieur par rapport à l'approche linéaire (pour les deux modèles 1 et 2).

D'après les résultats présentés dans le tableau III.6, on peut conclure que notre approche proposée (UPSO-FSVR) basée sur des modèles flous triangulaires non symétriques est mieux que l'approche linéaire (LP-FR) basée sur des modèles flous trapézoïdaux.

Conclusions et perspectives

Dans ce mémoire, on a présenté deux types de méthodes appliquées dans l'identification d'un modèle régressif flou. L'une consiste à minimiser un critère linéaire ou quadratique dépendant des données observées et/ou des sorties mesurées, c'est l'approche conventionnelle, tandis que l'autre revient à minimiser un critère complètement indépendant de ces dernières, c'est la régression floue à vecteur de support (FSVR).

Il faut noter aussi, que si l'on souhaite obtenir un modèle de possibilité, qui respect l'inclusion complète des données dans la sortie prédite, la minimisation doit se faire sous contraintes dans tous les cas, et comme la complexité d'un problème d'optimisation linéaire étant moins importante, il est plus judicieux d'utiliser l'approche linéaire proposée.

Cependant, si l'inclusion n'est plus indispensable, c'est-à-dire que l'on cherche un modèle de nécessité (modèle le plus proche des données), il sera donc plus intéressant d'utiliser une méthode quadratique de minimisation de distance, puisque dans ce cas les contraintes peuvent être relaxées.

Aussi, dans ce travail, nous avons proposé une nouvelle contribution de l'identification des modèles régressifs flous basée sur la méthode de FSVR avec des nombres flous triangulaires non symétriques et l'optimisation par essaim de particules unifiée (UPSO), cette nouvelle méthode on l'a testée dans deux exemples illustratifs monovariés, puis on l'a appliquée à un jeu de données multi-entrées bruitées, elle donne des résultats prometteurs meilleur que la technique linéaire modifiée de Bissier. Ce concept permet au final de disposer des modèles régressifs flous identifiés dont l'imprécision est optimisée, et dont la sortie englobe intégralement l'ensemble des observations.

Enfin, d'autres développements peuvent être introduits dans cette nouvelle approche pour traiter des jeux de données dont les entrées et la sortie sont imprécises.

Annexe A

Le jeu de données multivariables

j	$type$	x_1	x_2	x_3	x_4	x_5	y
1	test	0.36031	0.28594	0.014864	0.8952	0.51015	1.3915
2	identification	0.54851	0.39413	0.28819	0.94239	0.71396	4.6557
3	identification	0.26177	0.50301	0.81673	0.33508	0.51521	6.2132
4	outlier	0.59734	0.72198	0.98548	0.43736	0.60587	10.5848
5	test	0.049278	0.30621	0.017363	0.47116	0.9667	1.1886
6	identification	0.57106	0.11216	0.81939	0.14931	0.82212	5.658
7	identification	0.70086	0.44329	0.62114	0.13586	0.31775	6.4823
8	identification	0.96229	0.46676	0.56022	0.5325	0.5877	8.2347
9	identification	0.75052	0.014669	0.24403	0.72579	0.1302	1.7014
10	identification	0.73999	0.66405	0.82201	0.3987	0.25435	9.7441
11	outlier	0.43187	0.72406	0.26321	0.35842	0.80303	5.9823
12	identification	0.63427	0.28163	0.75363	0.28528	0.66785	6.4305
13	identification	0.80303	0.26182	0.65964	0.86864	0.013626	6.5469
14	identification	0.083881	0.70847	0.21406	0.62641	0.56158	2.2481
15	identification	0.94546	0.78386	0.60212	0.24117	0.45456	10.919
16	identification	0.91594	0.98616	0.60494	0.97808	0.90495	14.06
17	identification	0.60199	0.47334	0.6595	0.6405	0.28216	7.0714
18	identification	0.25356	0.90282	0.18336	0.22985	0.065034	3.2945
19	identification	0.87345	0.45106	0.63655	0.68134	0.47659	8.2171
20	identification	0.5134	0.80452	0.17031	0.66582	0.98371	6.1764

Annexe A : Le jeu de données multivariables

j	$type$	x_1	x_2	x_3	x_4	x_5	y
21	identification	0.73265	0.82886	0.5396	0.13472	0.92235	9.7668
22	identification	0.42223	0.16627	0.62339	0.022493	0.5612	4.162
23	outlier	0.96137	0.39391	0.68589	0.2622	0.65232	9.6279
24	identification	0.072059	0.52076	0.67735	0.11652	0.77268	4.5169
25	identification	0.55341	0.71812	0.87683	0.069318	0.10618	8.4911
26	test	0.29198	0.56919	0.012891	0.85293	0.0010734	1.7484
27	identification	0.85796	0.46081	0.3104	0.18033	0.54176	5.911
28	identification	0.33576	0.44531	0.77908	0.032419	0.0068578	5.4411
29	identification	0.6802	0.087745	0.3073	0.73393	0.45134	2.7881
30	identification	0.053444	0.44348	0.92668	0.53652	0.19566	5.9031
31	identification	0.35666	0.3663	0.67872	0.27603	0.78714	5.6943
32	test	0.4983	0.30253	0.074321	0.36846	0.61856	2.3165
33	test	0.43444	0.85184	0.070669	0.012886	0.015521	4.0561
34	test	0.56246	0.75948	0.01193	0.88921	0.89085	5.1463
35	identification	0.61662	0.94976	0.22715	0.86602	0.7617	7.9658
36	outlier	0.11334	0.55794	0.51625	0.25425	0.90704	4.3186
37	identification	0.89825	0.014233	0.4582	0.56948	0.75857	3.5161
38	identification	0.75455	0.59618	0.7032	0.15926	0.38073	8.3834
39	identification	0.79112	0.81621	0.58248	0.59436	0.33111	10.172
40	identification	0.81495	0.97709	0.50921	0.3311	0.50408	11.1
41	test	0.67	0.22191	0.07429	0.65861	0.56457	2.2748
42	identification	0.20088	0.70368	0.19324	0.86363	0.7672	3.3021
43	identification	0.27309	0.52206	0.3796	0.56762	0.77987	4.3628
44	identification	0.62623	0.9329	0.27643	0.98048	0.4841	8.0007
45	identification	0.53685	0.71335	0.77088	0.79183	0.80221	9.5484
46	identification	0.059504	0.22804	0.31393	0.15259	0.47101	2.023
47	outlier	0.088962	0.44964	0.63819	0.83303	0.20276	4.9731
48	identification	0.27131	0.1722	0.98657	0.19186	0.57961	6.1146
49	identification	0.40907	0.96882	0.50288	0.63899	0.6665	7.5644

Annexe A : Le jeu de données multivariables

j	$type$	x_1	x_2	x_3	x_4	x_5	y
50	identification	0.47404	0.35572	0.9477	0.669	0.67677	8.1508
51	identification	0.90899	0.049047	0.82803	0.77209	0.94251	6.7529
52	identification	0.59625	0.75534	0.91756	0.37982	0.77015	10.382
53	identification	0.32896	0.89481	0.11308	0.44159	0.7374	4.1526
54	identification	0.47819	0.28615	0.81213	0.48306	0.86626	6.964
55	identification	0.59717	0.2512	0.90826	0.60811	0.99095	8.128
56	outlier	0.16145	0.93274	0.15638	0.176	0.50393	3.0023
57	identification	0.82947	0.13098	0.12212	0.002026	0.62909	2.0933
58	outlier	0.95612	0.94082	0.76267	0.79022	0.79261	15.0394
59	identification	0.59555	0.70185	0.7218	0.51361	0.44865	8.7316
60	identification	0.028748	0.84768	0.65164	0.21323	0.52436	4.0547
61	identification	0.81212	0.20927	0.75402	0.10345	0.17147	5.655
62	identification	0.61011	0.45509	0.66316	0.15734	0.13067	6.3181
63	identification	0.70149	0.081074	0.88349	0.40751	0.21878	5.7541
64	outlier	0.092196	0.85112	0.27216	0.40776	0.10548	3.5862
65	outlier	0.42489	0.56205	0.41943	0.052693	0.14143	5.0938
66	identification	0.37558	0.3193	0.21299	0.94182	0.45697	2.8742
67	test	0.16615	0.3749	0.0356	0.14997	0.78813	1.4327
68	identification	0.83315	0.8678	0.081164	0.38437	0.28106	7.7773
69	identification	0.83864	0.37218	0.85057	0.31106	0.22479	7.9538
70	identification	0.45161	0.07369	0.3402	0.16853	0.90887	2.9745
71	identification	0.9566	0.19984	0.46615	0.89665	0.007329	5.0784
72	identification	0.14715	0.049493	0.91376	0.32272	0.58874	5.578
73	identification	0.86993	0.56671	0.22858	0.734	0.54212	6.7023
74	identification	0.76944	0.12192	0.86204	0.4109	0.65352	6.3838
75	identification	0.44416	0.52211	0.65662	0.39979	0.31343	6.2254
76	identification	0.62062	0.11706	0.89118	0.50552	0.23116	6.1369
77	identification	0.95169	0.76992	0.48814	0.16931	0.41606	10.106
78	identification	0.64001	0.37506	0.99265	0.52475	0.2988	8.4947

Annexe A : Le jeu de données multivariables

<i>j</i>	<i>type</i>	<i>x1</i>	<i>x2</i>	<i>x3</i>	<i>x4</i>	<i>x5</i>	<i>y</i>
79	identification	0.24733	0.82339	0.37333	0.6412	0.67244	4.8341
80	identification	0.3527	0.046636	0.53138	0.016197	0.93826	3.7189
81	identification	0.18786	0.59791	0.18132	0.83685	0.34315	2.4511
82	identification	0.49064	0.94915	0.50194	0.80346	0.56296	8.2901
83	outlier	0.40927	0.2888	0.42219	0.69778	0.11889	4.2938
84	identification	0.46353	0.88883	0.66043	0.46189	0.16902	8.0608
85	identification	0.61094	0.10159	0.67365	0.082613	0.2789	4.178
86	identification	0.071168	0.065315	0.95733	0.82072	0.55681	6.7146
87	identification	0.31428	0.2343	0.19187	0.19302	0.48559	2.0056
88	identification	0.60838	0.9331	0.11122	0.44535	0.95222	7.2387
89	identification	0.17502	0.063128	0.56505	0.012958	0.23192	3.0042
90	identification	0.62103	0.26422	0.96917	0.30874	0.47866	7.3143
91	test	0.24596	0.99953	0.023744	0.87535	0.52652	2.896
92	identification	0.58736	0.21199	0.87022	0.83526	0.79272	7.6784
93	test	0.50605	0.49841	0.026877	0.3331	0.19301	2.7117
94	identification	0.46478	0.29049	0.51953	0.88071	0.9096	5.6903
95	identification	0.54142	0.67275	0.19229	0.47969	0.9222	5.6388
96	identification	0.94233	0.95799	0.71569	0.56082	0.013266	13.409
97	identification	0.34176	0.76655	0.25067	0.61591	0.76755	4.771
98	identification	0.4018	0.66612	0.93386	0.6619	0.94734	9.4795
99	identification	0.30769	0.13094	0.13719	0.61663	0.81331	1.9195
100	identification	0.41157	0.095413	0.52162	0.68514	0.92383	4.569

Bibliographie

- [1] Bissier, A., Reda Boukezoula*, Sylvie Galichet. A revisited approach for linear fuzzy regression using trapézoïdal fuzzy intervals, INS 8716, 24 june 2010.
- [2] Bissier, A. Une approche paramétrique de la régression linéaire floue - formalisation par intervalles, Thèse doctorat, UNIVERSITE DE SAVOIE 2009.
- [3] Buckley, J., Feuring, T., and Hayashi, Y., “Multivariate non-linear fuzzy regression : An evolutionary algorithm approach,” *Int. J. Uncertain., Fuzziness Knowl.-Based Syst.*, vol. 7, pp. 83–98, 1999.
- [4] Buckley, J. and Feuring, T. “Linear and non-linear fuzzy regression : Evolutionary algorithm solutions,” *Fuzzy Sets Syst.*, vol. 112, pp. 381–394, 2000.
- [5] Celmins, A. “A practical approach to nonlinear fuzzy regression,” *SIAM J. Sci. Statist. Comput.*, vol. 12, no. 3, pp. 521–546, 1991.
- [6] Chan, KY., Kwong, CK., and Fogarty, TC. Modeling manufacturing processes using a genetic programming based fuzzy regression with detection of outliers. *Information Sciences*,180(4) :506–518, 2010.
- [7] Clerc, M., Kennedy, J. : The particle swarm–explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* 6 (2002) 58–73.
- [8] Cornillon Pierre-André et Eric Matzner-Lober (2007). Régression théorie et applications.
- [9] Cristianini, N. and Shawe-Taylor, J., *An Introduction to Support Vector Machines*. Cambridge, U.K. : Cambridge University Press, 2000.
- [10] Diamond, P. et R. Körner, (1997), *Extended Fuzzy Linear Models and Least Squares Estimates*, . *Computers & Mathematics with Applications*, Vol. 33, 9, pp. 15–32.
- [11] Diamond, P., (1988), *Fuzzy Least Squares*, *Information Science*, Vol. 46, 3, pp. 141–157.
- [12] Eberhart, R.C., Kennedy, J. : A new optimizer using particle swarm theory. In : *Proc. 6th Symp. Micro Mach. Hum. Sci.*, IEEE Service Center 39–43 (1995).

- [13] Gunn, S. Support Vector Machines for Classification and Regression, ISIS Tech. Report, University of Southampton, 1998.
- [14] Hao Pei-Yi and Chiang jung-Hsien, *Senior Member, IEEE*. Fuzzy Regressions Analysis by Support Vector Learning Approach, VOL. 16,NO. 2, APRIL 2008.
- [15] Hong, D. H., and Hwang, C. : Support vector fuzzy regression machines. *Fuzzy Sets and Systems* 138 (2003) 271–281 (2003).
- [16] Hong, D. H., and Hwang, C. : Extended fuzzy regression models using regularization method. *Information Sciences* 164 (2004) 31–46.
- [17] Hong, D. H., and Hwang, C. : Ridge regression procedures for fuzzy models using triangular fuzzy numbers *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 12 (2004) 145–159.
- [18] Hong, D. H.1, and Hwang, C.2 : Fuzzy Nonlinear Regression Model Based on LS-SVM in Feature Space, *FSKD 2006, LNAI 4223*, pp. 208–216, 2006. Springer-Verlag Berlin Heidelberg (2006).
- [19] Idri, A., (2003), Un modèle intelligent d'estimation des coûts de développement de logiciels, Thèse de . doctorat en Informatique, Montréal, Université du Québec.
- [20] Joachims, T. : Making large-scale support vector machine learning practical". In *Advances in Kernel Methods : Support Vector Machines*, A. S. Scholkopf, C. Burges, Ed. MIT Press, Cambridge, MA, 1998.
- [21] Kennedy, J., Eberhart, R.C. : *Swarm Intelligence*. Morgan Kaufmann Publ. (2001).
- [22] Konstantinos E. Parsopoulos, Michael N. Vrahatis : *Particle Swarm Optimization and Intelligence : Advances and Applications*. Published in the United States of America by Information Science Reference (an imprint of IGI Global), pages 89–98 (2010).
- [23] Liaoa Ruijin , Hanbo Zhenga*, Stanislaw Grzybowski, Lijun Yanga : Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers. *Electric Power Systems Research* 81 (2011) 2074– 2080 (2011).
- [24] Pahikkala, T., Boberg, J. and Salakoski, T. Fast n-fold cross-validation for regularized least-squares. In T. Honkela, T. Raiko, J. Kortela, and H. Valpola, editors, *Proceedings of the Ninth Scandinavian Conference on Artificial Intelligence (SCAI 2006)*, pages 83–90, Espoo, Finland, 2006. Otamedia Oy.

-
- [25] Parsopoulos, K.E.1, and Vrahatis, M.N.2. : Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems ICNC 2005, LNCS 3612, pp. 582–591. Springer-Verlag Berlin Heidelberg 2005.
- [26] Platt, J., N. Cristianini and J. Shawe-Taylor. "Large margin dags for multiclass classification". S. Solla, T. Leen, and K.-R. Mueller, Eds.
- [27] Samprit Chattefuee et Alis Hadi (2006). Regression Analysis by Example. fourth edition.
- [28] Savic, D. A. et W. Pedrycz, (1991), Evaluation of Fuzzy Linear Regression Models, Fuzzy Sets and . Systems, Vol. 39, 1, pp. 51–63.
- [29] Smola, A. J. Regression estimation with support vector learning machines. Master's thesis, Technische Universit at M unchen, 1996.
- [30] Tanaka, H., S. Uejima et K. Asia, (1982), Linear Regression Analysis with Fuzzy Model, IEEE . Transactions on SMC, Vol. 12, pp. 903–907.
- [31] Tanaka, H. et Lee, H. (1998). Interval Regression Analysis by Quadratic Programming Approach, . IEEE. Transactions on Fuzzy Systems, Vol. 6, pp. 473–481.
- [32] Tanaka, H. et Lee, H. (1997). Fuzzy Linear Regression Combining Central Tendency and Possibilistic . Properties. IEEE. Fuzzy Sets and Systems, 24 :063-068,
- [33] Tong-Tong, J-R. (1995). La logique floue, Hermes, Paris.
- [34] Vapnik, V. N., The Nature of Statistical Learning Theory. Springer -Verlag New York, 1995.
- [35] Vapnik, V. , Golowich, S., and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," Adv. Neural Inform. Process. Syst., vol. 9, pp. 281–287, 1996.
- [36] Wang, H.-F. et R.-C. Tsaur, (2000), Insight of a Fuzzy Regression Model, Fuzzy Sets and Systems. . Vol.112, 3, pp. 355–369.
- [37] Wu, B. et N.-F. Tseng,(2002), A New Approach to Fuzzy Regression Models with Application to Busines . Cycle Analysis, Fuzzy Sets and Systems, Vol. 130, 1, pp. 33–42.
- [38] Wuqi*, Rob Lawb : Fuzzy support vector regression machine with penalizing Gaussian noises on triangular fuzzy number space Expert Systems with Applications 37 (2010) 7788–7795, Elsevier Ltd (2010).
- [39] Zadeh, L. A., (1978), Fuzzy Sets as a Basis for a Theory of Possibility, Fuzzy Sets and Systems, . Vol. 1, 1, pp.3–28.
-

Mots clé : Logique floue, intervalle flou, régression floue, opérateurs d'agrégations, machines à vecteur de support (SVMs).

Résumé : Les travaux de recherche sur la régression floue se sont multipliés et il est maintenant très difficile d'établir une liste exhaustive de toutes les stratégies développées, les premiers développements ont été initiés par Tanaka et al. L'idée consiste à exploiter des concepts possibilistes pour identifier les paramètres d'un modèle linéaire en minimisant la dispersion des sorties prédites. Un programme d'optimisation linéaire sous contraintes traduit alors la stratégie mis en œuvre pour réaliser cette identification.

Parallèlement à ces travaux, l'idée d'une identification basée sur la minimisation d'un critère quadratique se concrétise avec l'apparition des moindres carrés flous. Dans ce contexte, diverses distances ont été exploitées en tant que critère à optimiser.

Dans le contexte de la réalisation de l'inclusion, des contraintes ont été introduites dans le problème d'optimisation pour aboutir à une programmation quadratique sous contraintes.

Les critères conventionnels présentent l'inconvénient d'être dépendants des données d'apprentissage. Nous avons montré au travers d'exemples illustratifs que cette dépendance nuit à la robustesse des modèles identifiés. Pour remédier à cela nous allons proposer un critère indépendant des données, la défaillance en terme de robustesse évoquée précédemment est alors surmontée.

Abstract : Research on fuzzy regression has increased and it is now very difficult to establish an exhaustive list of all the strategies developed, the first developments were initiated by Tanaka et al. The idea is to exploit possibilistic concepts to identify the parameters of a linear model by minimizing the dispersion of the predicted outputs. A program of linear optimization under constraints then translates the strategy implemented to achieve this identification.

Alongside this work, the idea of identification based on the minimization of a quadratic criterion is realized with the advent of fuzzy least squares. In this context, different distances were used as a criterion to optimize. In the context of achieving the inclusion of constraints were introduced into the optimization problem to achieve a constrained quadratic programming.

The conventional criteria have the disadvantage of being dependent on the training data. We have shown through illustrative examples that this dependence affects the robustness of the models identified. To remedy this we will propose a novel approach with independent data criterion, the failure in terms of robustness mentioned above is then overcome.