Université catholique de Louvain
Faculté des Sciences Appliquées
Département d'Electricité
Laboratoire de Microélectronique

Méthodes non linéaires pour séries temporelles :

prédiction par Double Quantification Vectorielle et sélection de délai en hautes dimensions

Jury:

Prof. **L. Vandendorpe** (Président)	Université catholique de Louvain, Belgique
Prof. **M. Verleysen** (Promoteur)	Université catholique de Louvain, Belgique
Prof. **G. Bontempi**	Université Libre de Bruxelles, Belgique
Prof. **M. Cottrell**	Université Paris I - Panthéon Sorbonne, France
Ph.D. **J. A. Lee**	Université catholique de Louvain, Belgique
Prof. **V. Wertz**	Université catholique de Louvain, Belgique

Thèse présentée par

Geoffroy Simon

en vue de l'obtention du grade de

Docteur en Sciences Appliquées

Juin 2007

A Samia, à Lilia

Remerciements

Bien que relativement courte, cette partie de ma thèse est sans aucun doute la plus importante pour moi.

En premier lieu, je tiens à remercier mon promoteur, Michel Verleysen. Je me vois encore, “débarquant” dans son bureau un après-midi de mai, il y a maintenant près de cinq ans. Le début de mon parcours scientifique à ses côtés. Cinq ans de collaboration, un mémoire, de nombreuses discussions (où je lui expliquais que je finirais bien par trouver ce bug ...), des articles, des conférences et puis cette thèse. Michel, un grand scientifique, mais aussi un homme que j'ai appris à connaître, avec son carisme, son humour et ses éclats de rire, sa fatigue accumulée suite aux heures de travail jour et nuit (notamment pour relire l'un ou l'autre article de ses doctorants, terminé en retard), son stress permanent à force de gérer des dizaines de questions et de dossiers en même temps, et ce sans compter l'organisation de cet évènement annuel dans la Venise du Nord. Et au-delà de tout cela, une disponibilité et une grande liberté offertes à ses chercheurs. Merci Michel, pour cette collaboration qui m'a permis d'apprendre à tant de niveaux différents au cours de ces dernières années. Comme je l'ai dit dans un autre contexte en avril 2006, à Bruges, c'est un plaisir de travailler avec toi.

Je souhaite remercier Marie Cottrell et Vincent Wertz, qui ont encadré mon travail scientifique. Marie est une des personnes les plus brillantes qu'il m'a été donné de rencontrer lors de mon parcours universitaire ; une grande personnalité scientifique dont les compétences n'ont d'égales que son engagement politique et sa modestie qui forcent le respect. Merci Marie, ce fut pour moi un honneur de travailler avec toi et un plaisir de discuter avec toi lors des différentes conférences où nous nous sommes revus. Vincent a toujours été présent aux moments clés de cette thèse, et il m'a également permis de participer à plusieurs conférences via un compte commun avec Michel (oui, tout cet argent dépensé par Michel ...). Au-delà de ces aspects financiers, il est toujours bon d'avoir un regard extérieur qui apporte ce que l'auto-critique oublie parfois. Merci Vincent pour ce rôle ô combien important que tu as tenu avec la parfaite rigueur du grand mathématicien que tu es.

Je remercie également John Lee et Gianluca Bontempi, pour leur participation au jury de thèse. Merci à vous deux d'avoir accepté la charge de lire les quelques dizaines de pages qui suivent et d'y avoir apporté, par vos remarques judicieuses et constructives, une contribution significative. Merci aussi à Luc Vandendorpe, président du jury, qui a accepté cette tâche malgré un agenda bien rempli.

Je voudrais ensuite remercier Amaury Lendasse. Amaury, qui m'a guidé au cours de mon mémoire. Le nombre élevé de fois où je suis venu frapper à sa

porte est au moins aussi grand que la disponibilité et la patience avec lesquelles il a, à chaque fois, pris le temps de me répondre. Même si tu t'es depuis envolé pour la Finlande, je n'oublierai pas les heures passées dans ton bureau où, devant le tableau, tu m'as expliqué bon nombre de concepts liés à ces fameuses séries temporelles.

Un merci tout particulier aux collègues du DICE membres du Machine Learning Group, pour la bonne humeur et les diverses compétences scientifiques et techniques : Cédric Archambeau, collègue de bureau et spécialiste de la relevance Bayesienne vraisemblable et robuste, à moins que ce ne soit l'inverse (tout ce que j'aime et ne comprends pas toujours, j'avoue), parti "respirer" le smog londonien ; Nabil Benoudjit, spécialiste des RBFN et de l'analyse de spectres, parti dans son Algérie natale ; Nicolas Donckers, spécialiste en électronique embarquée et sans doute l'assistant le plus dévoué pour ses étudiants, parti consulter en Cobol ; John Lee (encore lui ;-)), spécialiste de la PCA, de l'ICA, de la CCA, des SOM, d'ISOMAP et j'en passe dans les méthodes de projection, parti explorer les sous-sols de Saint-Luc ; Amaury Lendasse (encore lui ;-)), spécialiste en séries temporelles et en sélection de structure de modèles, parti se goinfrer de poissons en Finlande ; Damien François, spécialiste de la sélection et des hautes dimensions, appliqués entre autres à des spectres, en train de créer la spin-off qui va tous nous engager, finalement ; Frédéric Vrins, collègue de bureau et spécialiste de l'entropie, de l'information mutuelle et de l'ICA/BSS, dont l'auto-ICA (la séparation, quoi !) avec l'université semble se concrétiser ; Mahamadou Biga, spécialiste en analyse longitudinale et en attente de bonnes nouvelles pour la suite (c'est le minimum que je te souhaite !), et ceux dont la thèse était encore en cours lorsque mon temps au labo était écoulé : Nicolas Delannay, collègue de bureau et de projet pour le cours, Catherine Krier et Gaël De Lannoy.

Merci à Anne Adant et Viviane Sauvage, et aux différent(e)s secrétaires et comptables qui ont oeuvré en DICE. Grâce au travail de fourmi effectué par tout ce personnel administratif, je n'ai pas eu à me tracasser de bien des papasseries lors de mes années de thèse. Merci par ailleurs à Viviane dont j'ai partagé le bureau et les discussions le temps de midi pendant plus de deux ans. Merci aussi à celles et ceux, collègues en DICE, en EMIC, en TELE ou en ELEC, que j'ai appris à connaître au fil des mois et des années.

Au cours de mes années en DICE, j'ai également participé à la gestion du réseau informatique du laboratoire. Je remercie Brigitte Dupont, qui m'a fait confiance pour occuper cette tâche, et Damien Giry, qui m'a formé au travers de son expertise personnelle. Pour une fois, et malgré la meilleure volonté du petit Padawin, je crains que l'élève ne puisse jamais dépasser le grand maître Jadawin. Merci à François Hubin qui a repris cette gestion informatique et qui y a apporté le temps et l'énergie que je n'y ai plus investi pendant les derniers

mois de thèse au labo. Merci également à ceux, nombreux, que j'ai rencontré à l'UCL dans le cadre de cette gestion informatique, et au contact desquels j'ai pu enrichir mes connaissances.

Merci à toutes celles et ceux que j'ai eu le plaisir de rencontrer et au contact de qui j'ai tant appris lors de mes activités de représentant, pour l'ACSSA et pour le CORSCI. Que toutes ces personnes soient remerciées au travers des remerciements que j'adresse ici plus particulièrement à Sandra Soares Frazão et à Laurent De Briey, Présidente ACSSA et Président CORSCI, respectivement, du moins au moment où je quittais le monde universitaire. Merci de m'avoir fait confiance.

Au-delà du microcosme de l'UCL, je remercie mes parents et mes soeurs. Merci à mes parents d'avoir été là, même si la communication est parfois difficile : comment parler d'études en mathématiques, puis en informatique, et d'une thèse dans un domaine à mis chemin entre l'informatique appliquée et les mathématiques appliquées, à une mère plus ou moins allergique aux mathématiques et à un père qui voit l'informatique plus comme une contrainte que comme un outil ? J'ai par ailleurs la grande chance d'avoir une soeur dans le domaine littéraire et une soeur dans le domaine artistique. La Science ... Oui, la science ... Grâce à elles, j'ai pu relativiser mon travail. Merci à vous deux, Geneviève et Myriam.

Un merci chaleureux à vous tous qui, au cours de ces dernières années, m'avez offert un bon moment en famille ou entre amis, que ce autour d'un repas, d'un verre, d'un puzzle ou lors d'un whist ou d'une discussion : JFB et Noé, mes actuaires préférés, Rénald et Véro, mes enseignants préférés, Nathalie et Laurence, mes jumelles préférées ; Aurélie, Bu, Gav et Ariane, Frédéric et Patricia, Cog´ et Nancy, Fabrice et Delphine, mes informaticiens préférés ; Cristel, Steve, Stéphane, Coralie, Pierre, navetteurs de Louvain-la-Neuve à Namur et inversément ; les Daltoniens et les Dupont's Girls ; Anne et Vincent, Benoît et Nathalie, Carine et Jean-Phi et, tout particulièrement, Geneviève et Stéphane (ils savent pourquoi).

Toutes ces personnes représentent l'époque que je qualifierais "d'avant et pendant" la thèse. Aujourd'hui, et depuis quelques mois déjà, il me faut aussi parler de l'époque "d'après". Un petit mot donc à l'intention de mes nouveaux collègues de l'équipe Middleware de Smals qui m'ont accueilli parmi eux et dans un domaine à des années lumières des préoccupations de la recherche. Merci à vous tous, Olivier, Michel, Victor, Stéphane (a.k.a. Parrain), Gilles, Didier, Thomas, Philippe (P.), Michaël, Tijani, Alain, Christophe, Jean-Marc et Philippe (G.) (dans l'ordre dans lequel je peux vous voir depuis mon bureau ... d'ici le prochain déménagement !). Merci d'avoir rendu mon intégration au sein de l'équipe Middleware aussi facile et aussi agréable, et d'animer avec tant de diversité et de richesse les sujets de discussion durant les pauses café.

Mention spéciale pour celles et ceux que j'ai pu oublier ci-dessus malgré mon soucis d'exhaustivité : soyez également remerciés pour votre indulgence à mon égard.

Last but not least, je tiens à remercier du fond du coeur mon épouse Samia et notre petite Lilia. Merci à toi, Lilia, petit ange qui réalise sans te forcer l'exploit de dissiper toutes les petites contrariétés du quotidien en un seul de tes grands sourires. Merci à toi, Samia, la noblesse de ton nom n'est rien à côté de la noblesse de ta personne, de ta gentillesse, de ta générosité et de ton amour.

Table des matières

Chapitre 1

Introduction à la prédiction de séries temporelles

Que ce soit par le biais de la météorologie ou via les commentaires des analystes des marchés financiers, la plupart d'entre nous a déjà entendu parler de "prédiction". En effet, estimer la valeur future d'une grandeur en météorologie ou d'un produit financier est, aujourd'hui, une pratique courante.

Généralement, afin de pouvoir estimer la valeur future d'une variable d'intérêt, en d'autres termes afin de prédire cette variable, il est indispensable de disposer d'informations relatives à l'évolution passée de la variable. Pour revenir au contexte financier, comment pourrait-on prédire la valeur de demain d'un indice si personne ne connaissait les valeurs de cet indice sur plusieurs jours passés, au minimum ? Une "série temporelle" se définit donc intuitivement comme la suite des valeurs passées de la variable qu'on essaie de prédire.

Pour passer de l'évolution passée à la (aux) valeur(s) future(s), il est utile d'analyser et de modéliser cette évolution passée de la série. Ces questions d'analyse de la série et de la construction d'un modèle sont souvent le coeur même du problème en prédiction. Analyser une série permet de la comprendre et par exemple de choisir certains paramètres du modèle utilisé. Par ailleurs, le modèle, une fois construit, doit idéalement être capable de représenter aussi fidèlement que possible la dynamique de l'évolution propre à chaque série. Lorsque le modèle représente effectivement cette évolution passée, on peut raisonnablement considérer que la prédiction sera pertinente par rapport à la dynamique qui a été mesurée et qui se trouve dans la série. Par contre, un modèle qui ne correspond pas aux valeurs disponibles permettra certes de faire une prédiction, mais cette dernière n'aura aucun sens au regard de la série considérée. La construction d'un modèle conforme à une série et l'analyse de la dynamique de cette série sont donc des étapes importantes de tout problème de prédiction.

Dans ce chapitre, la prédiction de séries temporelles est formalisée, étape par étape. Les questions d'analyse et de modélisation, liées au but initial de la prédiction, sont également introduites de manière rigoureuse.

1.1 Définitions, notations et exemples

Formalisons à présent l'intuition de ce qu'est une série temporelle. Soit x une variable qui nous intéresse, par exemple la valeur d'un indice boursier. Comme x varie au cours du temps, on rend cette dépendance temporelle explicite en notant $x(t)$ où t représente un instant dans le temps. Une série temporelle S est une suite de valeurs $x(t)$ mesurées d'un processus variant au court du temps, à une fréquence d'échantillonnage le plus souvent constante. Les données temporelles $x(t)$ sont généralement ordonnées suivant leur indice dans le temps. Autrement dit, en considérant que le moment du premier échantillonnage est l'instant $t = 1$, on a :

$$S = \{x(1), x(2), x(3), \ldots, x(N)\} = \{x(t) \mid 1 \leq t \leq N\}, \tag{1.1}$$

où N est le nombre total de valeurs dans la série. Notons que $x(t)$ peut représenter soit une seule mesure, et est donc une valeur scalaire comme ci-dessus, soit un ensemble de mesures, alors reprises dans un vecteur. Par exemple, en météorologie, $x(t)$ peut représenter le vecteur comprenant la vitesse du vent, la température de l'air, le taux d'humidité de l'air et la pression atmosphérique mesurés à l'instant t. Dans le cas vectoriel, la donnée temporelle $\mathbf{x}(t)$ et ses différentes composantes sont notées :

$$\mathbf{x}(t) = \left(x^1(t), x^2(t), x^3(t), \ldots, x^d(t)\right), \tag{1.2}$$

où d est le nombre de composantes ou la dimension du vecteur. La définition 1.1 ci-dessus se généralise aisément au cas vectoriel.

Ces concepts et définitions sont illustrés à la figure 1.1. Cette figure représente l'évolution du nombre de taches observées à la surface du soleil, un phénomène étudié entre autres parce qu'il a une influence sur la transmission des ondes électromagnétiques (téléphonie mobile, localisation par GPS, etc.). La variable scalaire $x(t)$ représente le nombre de ces taches observées mensuellement entre janvier 1749 et février 2005 ; dans ce cas $N = 3074$ (144). Cette série des taches solaires est un benchmark bien connu dans la littérature sous le nom de la série SunSpot. Cette série est détaillée en Annexe A.

Outre ce premier exemple d'application, il existe en fait de très nombreux domaines où on rencontre des séries temporelles :

- en finance : les indices boursiers, les taux de change, les valeurs des actions et autres produits boursiers sur les différents marchés varient tout au long de la journée et, de jour en jour, tout au long de l'année ;
- en hydrologie : le débit des cours d'eau varie en fonction du moment de l'année mais aussi des saisons et des conditions climatiques ;
- en industrie : la demande en électricité varie continuellement suivant la consommation des clients, en fonction du moment de la journée, du moment dans la semaine, dans l'année ;
- en management : la charge de travail du personnel d'un magasin, ou d'un call-center, varie en fonction de l'heure dans la journée mais aussi du jour de la semaine voire du calendrier de certaines fêtes annuelles ou de périodes de congé ;

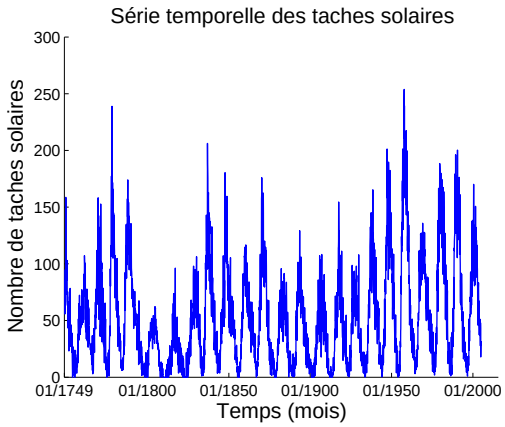

FIG. 1.1 – Série temporelle du nombre de taches solaires par mois, entre janvier 1749 et février 2005.

– etc.

Dans tous ces domaines, l'intérêt sous-jacent est, certes, spécifique à chaque problème :

– connaître la valeur future d'un indice précis permet de prendre une décision quant à la pertinence d'un investissement sur un marché boursier ;
– connaître l'impact, dans les heures à venir, des dernières précipitations sur le débit d'un cours d'eau permet d'éventuellement évacuer une zone d'habitations ;
– connaître la consommation future en électricité permet d'éventuellement ralentir la production ou au contraire de relancer l'activité d'une centrale ;
– connaître le nombre de clients qui se rendra au magasin ou appellera le call-center au cours de cette semaine, de ce mois, permet de planifier les horaires du personnel, voire la répartition entre les caisses et le réassortiment du magasin, par exemple.

Au travers de tous ces exemples, on constate toutefois que le but commun de ces problèmes concrets est de pouvoir réaliser une prédiction.

1.2 Analyse, modélisation et prédiction

Soit S une série particulière, par exemple une série financière pour fixer les idées. Le processus qui génère les données temporelles est, dans ce cas, l'activité des opérateurs sur le marché boursier. Cette activité est ici mesurée dans une variable scalaire $x(t)$ qui représente, par exemple, le cours d'un titre. Si nous notons le processus évoluant au cours du temps par g, nous avons donc :

$$x(t) = \nu(g, t), \tag{1.3}$$

où $\nu(g, t)$ représente la mesure du processus g au temps t.

Bien évidemment, si on pouvait connaître g pour tout instant t, il ne serait plus nécessaire d'essayer de prédire $x(t)$. En effet, sous cette hypothèse, quel que soit le moment dans le temps qui est considéré, on connaîtrait non seulement toutes les valeurs

passées, mais également toutes les valeurs futures. Malheureusement, en pratique, g n'est pas connu : il n'est connu qu'au travers des mesures $x(t)$ qui en sont prises. Pour prédire l'évolution de $x(t)$, il faut donc recourir à un modèle de g, une reconstruction mathématique de ce processus inconnu, notée $f(t,\ldots)$, et telle que, en moyenne, sur l'ensemble des valeurs d'une série temporelle :

$$f(t,\ldots) \approx \nu(g,t). \tag{1.4}$$

Signalons que la notation $f(t,\ldots)$ est volontairement très générale ; seule la dépendance temporelle est explicitée. Cette notation est précisée et complétée au Chapitre 2.

En supposant le modèle $f(t,\ldots)$ connu, il est alors possible d'utiliser ce modèle pour calculer une valeur de sortie $\hat{y}(t)$ au temps t :

$$\hat{y}(t) = f(t,\ldots), \tag{1.5}$$

où le symbole $\hat{}$ dénote le fait que la valeur fournie par le modèle n'est qu'une valeur estimée par calcul à distinguer de la véritable valeur $y(t)$, lorsqu'elle existe. Dans le cadre temporel, $\hat{y}(t)$ n'est autre que l'estimation d'une valeur temporelle notée $\hat{x}(t)$.

On résume donc la modélisation par la relation suivante :

$$\underbrace{x(t) = \nu(g,t)}_{\text{dernière mesure connue}} \quad \approx \quad \underbrace{f(t,\ldots)}_{\text{modèle du processus}} \quad = \quad \underbrace{\hat{y}(t) = \hat{x}(t)}_{\text{estimation fournie par le modèle}} \quad . \tag{1.6}$$

En d'autres termes, grâce à la modélisation, on obtient une valeur en sortie du modèle qui correspond en moyenne à la mesure du véritable processus.

Plus spécifiquement, dans un contexte de prédiction, le modèle est construit de sorte qu'on puisse obtenir une valeur $\hat{x}(t+h)$, avec un horizon de prédiction h fixé dans le temps. Dès lors, pour un problème de prédiction, partant de la dernière valeur connue $x(t)$, le modèle permet de calculer la prédiction $\hat{x}(t+h)$. Notons qu'on parle de prédiction à court terme lorsque $h = 1$; de prédiction à long terme pour $h > 1$.

Généralement, pour construire le modèle $f(t,\ldots)$, seules les données passées $x(t-j)$, décalées par rapport à $x(t)$ d'un certain délai j dans le temps, sont disponibles ainsi qu'éventuellement d'autres données passées $u(t)$ issues de processus extérieurs à g mais l'influençant. Les variables $x(t)$ utilisées dans le modèle et représentant les valeurs passées de la séries sont appelées les variables endogènes ; les variables $u(t)$ sont appelées les variables exogènes. Pour un problème de prédiction des températures au sol, par exemple, les variables endogènes sont bien évidemment les températures elles-mêmes, alors que les variables exogènes sont entre autres le taux d'ensoleillement, le taux d'humidité, la vitesse du vent, ... Comme l'indique cet exemple, il peut y avoir plusieurs variables exogènes suivant la série considérée, tout comme il peut tout aussi bien ne pas y en avoir. Il peut également y avoir plusieurs variables endogènes, séparées dans le temps par des délais plus ou moins longs. Par la suite, les modèles qui sont utilisés ne contiennent pas de variables exogènes. Les méthodes de modélisation de séries temporelles abordées aux Chapitres 2 et 3 restent néanmoins valables lorsqu'on prend en compte des données exogènes.

Si le choix d'un type de modèle parmi les nombreux types existants est généralement un choix effectué *a priori*, des choix tels que de la présence ou non de variables exogènes, de leur nombre, de même que du nombre de valeurs passées pour chaque variable considérée ou encore du délai dans le temps entre deux valeurs d'une variable, etc. sont des choix de modélisation auxquelles il est possible d'apporter une réponse au moyen d'une analyse de la série temporelle. L'analyse d'une série temporelle recouvre un ensemble de techniques qui peuvent constituer une aide à la modélisation en donnant un ensemble d'indications sur, par exemple, la complexité du modèle à utiliser pour approximer le processus dont est issu une série. L'analyse de séries temporelles est abordée aux Chapitres 4 et 5.

Pour résumer cette section, retenons que la prédiction d'une série temporelle est un problème bien plus complexe que la prédiction en elle-même. En effet, tout problème de prédiction commence par deux étapes indispensables, l'analyse et la modélisation :

- l'analyse : l'étude des données numériques constituant la série temporelle qui permet, entre autres, de sélectionner les variables les plus pertinentes à utiliser, qu'il s'agisse de variables endogènes $x(t)$ et/ou exogènes $u(t)$, ou encore de fixer leurs nombres respectifs, etc. ;
- la modélisation : la construction d'un modèle $f(t, \ldots)$ sur base de la série à disposition, ce qui revient à fixer les valeurs respectives des paramètres du modèle choisi en se basant sur les informations obtenues lors de l'analyse ;
- la prédiction : l'utilisation du modèle $f(t, \ldots)$ pour estimer la (les) valeur(s) future(s) de la série temporelle, à court terme ou à long terme.

1.3 Organisation de la thèse et contributions

Les questions rencontrées dans les étapes d'analyse, de modélisation et de prédiction sont généralement interdépendantes. Elles seront donc abordées tout au long de cette thèse. Néanmoins, l'accent sera mis au chapitre 2 sur les modèles. Un modèle particulier, permettant de la prédiction à long terme est introduit au chapitre 3. Les chapitres 4 et 5 sont consacrés à l'analyse de séries temporelles. Le reste de cette section décrit brièvement le contenu de chaque chapitre, et renvoie également vers les publications des résultats obtenus. La liste complète des publications issues des travaux présentés dans cette thèse, classée par ordre chronologique, est reprise à la page 137.

Les outils algorithmiques de modélisation utilisés dans cette thèse sont présentés au Chapitre 2. L'algorithme des cartes auto-organisées est décrit, ainsi que les réseaux à fonctions radiales de base. Quelques stratégies permettant de sélectionner la structure des modèles, ou dit autrement la complexité "optimale" d'un modèle, sont introduites. Parmi ces stratégies, la méthode de bootstrap est plus particulièrement détaillée ainsi qu'une méthode originale, le fast-bootstrap, qui est une approximation du bootstrap classique. Une première contribution personnelle, un test statistique permettant de valider l'hypothèse de linéarité utilisée dans la méthode de fast-bootstrap, y est alors détaillée. Par ailleurs, ce chapitre propose une description des différentes approches de prédiction

à long terme. Ces approches de prédiction sont observées dans le chapitre 2. Plusieurs contributions à différentes conférences ont été publiées dans le cadre des travaux sur le fast-bootstrap, (gs-1; gs-2; gs-3; gs-9; gs-11). Tous ces travaux sont résumés dans un article de revue (gs-13).

Dans le Chapitre 3, la Double Quantification Vectorielle, méthode de prédiction proposée dans le cadre de cette thèse, est expliquée et illustrée au moyen de son application à plusieurs séries temporelles. Il est alors étudié la manière dont cette méthode permet d'obtenir des prédictions à long terme, selon les différentes approches détaillées dans le chapitre 2. Une preuve théorique relative au comportement de la méthode en prédiction à long terme est par ailleurs proposée. Outre ces contributions, les performances de tout modèle de prédiction à long terme sont discutées. Deux méthodes originales reprenant certaines idées de la Double Quantification Vectorielle sont également décrites en fin de chapitre. La méthode de Double Quantification Vectorielle ont conduit aux contributions (gs-5; gs-6; gs-10) lors de différentes conférences. Ces travaux ont été développés dans les articles de revue (gs-12; gs-14; gs-20). Les publications (gs-7; gs-8; gs-16) concernent les deux méthodes originales décrites en fin de chapitre.

Le Chapitre 4 débat de l'importance de la sélection de variables dans le cadre de l'application des méthodes de quantification vectorielle appliquées à des données temporelles. Une contribution au débat scientifique actuellement en cours est proposée : un critère géométrique permettant de sélectionner un délai entre variables temporelles. Grâce à ce critère original, il est possible d'obtenir des modèles de séries temporelles par quantification vectorielle qui ont un sens, contrairement à ce qui a été affirmé récemment dans la littérature. Une contribution à une conférence (gs-15), ainsi qu'un article de revue (gs-18) ont été publiés suite à ces travaux sur la sélection de variables par distance à la diagonale.

Le Chapitre 5 poursuit l'étude de la sélection de variables, introduite au Chapitre 4. Les principaux résultats théoriques sont brièvement rappelés, et les méthodes classiques de sélection du délai entre variables temporelles, basées sur les critères d'autocorrélation et d'information mutuelle, sont rappelées. Les limites de la sélection du délai par autocorrélation et information mutuelle sont alors illustrées. Une première contribution de ce chapitre, la généralisation à plus de deux variables des deux méthodes de sélection de délais précitées, est proposée. Une généralisation à plus de deux variables du critère géométrique, présenté au Chapitre 4, complète cette étude de la sélection du délai en utilisant un nombre quelconque de variables. Une comparaison des différentes méthodes est détaillée au moyen de résultats expérimentaux. Les travaux de ce chapitre sont décrits dans une contribution à une conférence (gs-17), de plus amples détails étant fournis dans l'article de revue (gs-19).

Le Chapitre 6 conclut cette thèse, en proposant un résumé des différentes contributions, ainsi que des perspectives pour des développements futurs.

L'annexe A présente les différentes séries temporelles utilisées dans ce document. Ces séries servent à illustrer les différentes méthodes au travers de résultats expérimentaux.

Chapitre 2

Modèles, sélection de structure et prédiction à long terme

Dans le Chapitre 1, un modèle a été défini comme un outil mathématique de reconstruction d'un processus inconnu. Dans ce chapitre, quelques modèles existants sont brièvement passés en revue, et les modèles utilisés dans le cadre de cette thèse sont expliqués plus en détails. Par ailleurs, des méthodes pour sélectionner la complexité d'un modèle, ou sa structure, sont également introduites. Une contribution à une méthode de sélection de structure de modèles est proposée lors de la présentation de la méthode correspondante. Les différentes approches de prédiction à long terme sont également décrites, avant un résumé concluant ce chapitre.

2.1 Modèles

2.1.1 Comment s'y retrouver parmi les nombreux modèles ?

La prédiction de séries temporelles est un problème qui se retrouve dans beaucoup de domaines d'application. De nombreuses personnes se sont penchées sur cette question, notamment en finance, en mathématique, en physique et en machine learning. Issus de tous ces domaines, de nombreux modèles ont été proposés. Dans cette section, différentes approches de modélisation sont brièvement passées en revue, afin de mieux situer le cadre général auquel appartiennent les modèles qui sont utilisés dans cette thèse et leurs spécificités.

Les modèles les plus simples sont les modèles linéaires. Pour ceux-ci, il est fait l'hypothèse que le processus qui génère les données évolue de façon linéaire au cours du temps. Ces modèles linéaires sont abondamment utilisés en statistique et en identification des systèmes. Il s'agit, notamment, des modèles auto-regressifs $AR(p)$, des modèles à moyenne mobile $MA(q)$, des modèles auto-régressifs à moyenne mobile $ARMA(p, q)$, etc. Tous ces modèles sont décrits en détails dans plusieurs ouvrages de références, comme (12; 14; 113). Ces mêmes modèles sont également utilisés en finance, avec quelques extensions qui conduisent aux modèles auto-régressifs avec hétéroskédasticité conditionnelle

$ARCH(p)$ et à leur généralisation $GARCH(p)$ (48; 57; 67). Pour ces derniers modèles, l'hypothèse de linéarité n'est plus totalement vérifiée, puisqu'en général, le conditionnement de l'innovation par la partie auto-régressive se fait suivant une loi de distribution Gaussienne, qui n'est pas une loi linéaire.

En complément aux modèles linéaires, il existe donc d'autres modèles pour lequel aucune hypothèse de linéarité n'est faite sur le processus. Ces modèles dits non linéaires sont donc plus généraux et potentiellement plus puissants : ils peuvent modéliser n'importe quel type de processus et ne sont pas limités au cas particulier du linéaire. Cependant, ces modèles non linéaires sont aussi beaucoup plus complexes et parfois délicats à mettre en oeuvre. Outre les modèles financiers mentionnés ci-dessus, il existe de nombreux modèles non linéaires, issus des statistiques (splines (40; 77), wavelets (27; 77), etc.) ainsi que du domaine du machine learning (11; 45; 52; 76; 78).

Comme l'indique son nom, dans le domaine du machine learning, des algorithmes exécutables sur ordinateur permettent d'obtenir une modélisation d'une relation existant à l'intérieur d'un ensemble de données. Les algorithmes de machine learning décrivent généralement des modèles abstraits. Au cours de leur exécution, ces algorithmes utilisent une règle dite d'apprentissage qui leur est propre pour spécialiser petit à petit le modèle abstrait. A la fin de l'exécution, l'algorithme est supposer fournir un modèle qui correspond spécifiquement à la relation se trouvant au sein des données de départ. En machine learning également, il existe de nombreux modèles différents. Une classification de ces modèles peut être faite suivant le type de règle d'apprentissage. Dans le cadre de cette thèse, les apprentissages supervisé et non supervisé sont considérés. En apprentissage supervisé, il est tenu compte à la fois des entrées $x(t)$ et de la sortie $y(t)$ lors de l'exécution de l'algorithme ; alors qu'en non supervisé, seules les entrées $x(t)$ sont prises en compte. Alors que les modèles supervisés peuvent être utilisés aussi bien en classification qu'en approximation de fonction, les modèles non supervisés sont plutôt utilisés en classification uniquement. Parmi les modèles supervisés, on trouve le perceptron multi-couches (Multi-Layer Perceptron ou MLP dans la littérature) (69; 133; 139; 179), les support vector machines (ou SVM) (16; 23; 33; 141; 142; 169) et les réseaux à fonctions radiales de base (Radial Basis Function Network ou RBFN) (11; 15; 63; 122; 128). Seul ce dernier modèle est décrit plus en détails, les autres modèles n'ayant pas été utilisés dans ce travail. Plusieurs modèles non supervisés sont quant à eux décrits, et plus principalement les cartes auto-organisées (Self-Organizing Maps ou SOM) (97), qui sont à la base de la méthode de prédiction proposée au Chapitre 3.

2.1.2 Quantification vectorielle

Considérons un problème de la vie réelle dont on veut connaître la dynamique. Prenons par exemple la mesure de la température au sol qu'on souhaiterait modéliser pour une certaine région. Cette température en un endroit particulier évolue continuellement. Un nombre important de données devient vite disponible en fonction de la fréquence des mesures mais aussi et surtout, dans ce cas-ci, en fonction du nombre d'endroits différents

où des mesures sont effectuées. Comme le montre cet exemple, on se retrouve parfois confronté à un grand nombre d'informations qu'il devient relativement difficile de pouvoir intégrer sans en réduire, d'une manière ou d'une autre, le volume.

Une méthode classique pour réduire le volume d'un ensemble d'informations est la quantification vectorielle (4; 62; 65; 66; 111). Cette méthode permet en effet de diminuer la taille d'un ensemble de données vectorielles tout en limitant la perte d'information subie lors de cette réduction.

Conceptuellement, le résultat d'une quantification vectorielle est un ensemble de vecteurs disposés à l'intérieur d'une distribution de données de sorte qu'ils représentent à eux seuls, approximativement, l'ensemble de la distribution initiale.

Plus formellement, supposons qu'on dispose d'un ensemble $\mathcal{D}$ de N données vectorielles de dimension d, défini par :

$$\mathcal{D} = \left\{ \mathbf{x}_i \mid \mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^d) \in \mathbb{R}^d, 1 \leq i \leq N \right\}, \tag{2.1}$$

où les différentes composantes $x_i^1, x_i^2, \ldots, x_i^d \in \mathbb{R}$. Notons que le vecteur $\mathbf{x}_i$ peut par exemple correspondre à $\mathbf{x}(t)$; cette notation est d'ailleurs utilisée lorsque le contexte se rapportera plus spécifiquement aux séries temporelles.

Après quantification vectorielle de cet ensemble $\mathcal{D}$, on dispose d'un ensemble $\mathcal{P}$ de n vecteurs dits représentatifs, également appelés *prototypes* :

$$\mathcal{P} = \left\{ \bar{\mathbf{x}}_j \mid \bar{\mathbf{x}}_j = (\bar{x}_j^1, \bar{x}_j^2, \ldots, \bar{x}_j^d) \in \mathbb{R}^d, 1 \leq j \leq n \right\}, \tag{2.2}$$

où n est (beaucoup) plus petit que N. Bien évidemment, ces prototypes sont de dimension d tout comme le sont les données qu'ils représentent.

Un algorithme implémentant cette quantification vectorielle est, par exemple, le Competitive Learning (68; 82; 137). Brièvement, cet algorithme itératif non supervisé est en mesure d'adapter la position d'un ensemble de prototypes à une distribution décrite dans un ensemble de données $\mathcal{D}$ comme suit. Au cours d'une itération, qui correspond intuitivement à l'étape durant laquelle une donnée $\mathbf{x}_i$ est traitée par l'algorithme, le prototype le plus proche au sens d'une mesure de distance $dist(.,.)$, noté $\bar{\mathbf{x}}_k$, est sélectionné :

$$\bar{\mathbf{x}}_k = \arg \min_{1 \leq j \leq n} dist(\bar{\mathbf{x}}_j, \mathbf{x}_i), \tag{2.3}$$

où :

- i, dont la valeur est fixée dans la relation (2.3), est l'indice permettant de parcourir l'ensemble de données $\mathcal{D}$,
- j est l'indice permettant de parcours l'ensemble des prototypes $\mathcal{P}$, et
- k est l'indice du prototype le plus proche, selon la mesure de distance $dist(.,.)$ qui est le plus souvent la distance euclidienne.

Ce prototype $\bar{\mathbf{x}}_k$ est ensuite modifié : sa position au sein de la distribution des données est modifiée pour qu'elle soit plus représentative de la donnée considérée après l'itération en cours qu'avant. En d'autres termes, on rapproche le prototype de la donnée pour que

la position de ce prototype tienne compte de l'information contenue dans la position de la donnée :

$$\bar{\mathbf{x}}_k(iter+1) = \bar{\mathbf{x}}_k(iter) + \alpha(iter)(\mathbf{x}_i - \bar{\mathbf{x}}_k(iter)), \tag{2.4}$$

où les notations $iter$ et $iter+1$ représentent respectivement l'itération en cours et la suivante, et où le paramètre $\alpha(.)$ correspond au taux d'apprentissage. Généralement ce paramètre $\alpha(.)$ est choisi de sorte qu'il suive une fonction décroissante en fonction du nombre d'itérations, afin d'assurer la convergence de l'algorithme (132). Les deux étapes (2.3) et (2.4) ci-dessus sont répétées pour tout l'ensemble $\mathcal{D}$, $\mathcal{D}$ étant présenté plusieurs fois consécutives à l'algorithme.

A la fin de l'exécution de cet algorithme, pour chaque vecteur $\bar{\mathbf{x}}_j \in \mathcal{P}$, on définit pour chaque prototype $\bar{\mathbf{x}}_j$ un sous-ensemble de $\mathcal{D}$ regroupant toutes les données $\mathbf{x}_i$ qui sont plus proche de ce prototype que de tous les autres. Ces groupes de données sont appelés des clusters et sont notés c_j, où l'indice j correspond au prototype $\bar{\mathbf{x}}_j$ correspondant. Bien évidemment, il y a autant de clusters que de prototypes. Plus formellement, pour un prototype $\bar{\mathbf{x}}_k$ par exemple, on obtient le cluster c_k défini par :

$$c_k = \{\mathbf{x}_i \in \mathcal{D} \mid \forall j \in [1..n], j \neq k, dist(\bar{\mathbf{x}}_k, \mathbf{x}_i) < dist(\bar{\mathbf{x}}_j, \mathbf{x}_i)\}\,. \tag{2.5}$$

Notons que la création de tous les clusters c_j, pour les différents prototypes $\bar{\mathbf{x}}_j$ de l'ensemble $\mathcal{P}$, avec $1 \leq j \leq n$, revient à partitionner l'ensemble $\mathcal{D}$.

Un autre algorithme permettant de réaliser une quantification vectorielle, et qui est à la base de la méthode de modélisation et de prédiction présentée au Chapitre 3 est l'algorithme des cartes auto-organisées.

2.1.3 Les cartes auto-organisées

L'algorithme des cartes auto-organisées (Self-Organizing Maps ou SOM) est un algorithme itératif de classification non supervisée proposé par T. Kohonen (97). Depuis son introduction dans les années 80, l'algorithme SOM a été appliqué avec succès dans de très nombreuses applications, telles que la classification d'images, la classification de textes, la compression d'images ou encore la robotique pour n'en citer que quelques unes. Le lecteur désireux d'explorer d'avantage les applications de ce SOM est invité à consulter la liste bibliographique de référence, disponible en ligne (2). La description du SOM ci-dessous se limitera à une introduction des principaux concepts et à une description du fonctionnement de l'algorithme. De plus amples détails sur l'algorithme peuvent être trouvés dans (97; 98), les principaux résultats assurant un fondement théoriques de cette méthode se trouvant dans (29; 53; 54; 74; 79; 131).

Conceptuellement, le principe de l'algorithme SOM est, tout comme pour le Competitive Learning, de déplacer itérativement des prototypes au sein d'une distribution de données contenues dans un ensemble $\mathcal{D}$. A la fin de l'exécution de l'algorithme SOM, la distribution des prototypes représente une approximation de la distribution initiale des

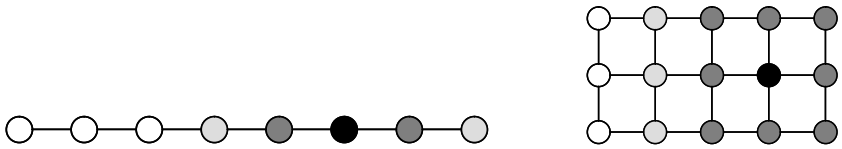

FIG. 2.1 – Exemples de grilles unidimensionnelle et bidimensionnelle, et des relations de voisinage définies pour un prototype sur chaque grille.

données. L'algorithme SOM permet donc de quantifier vectoriellement des données. Toutefois, le SOM possède une caractéristique particulière qui est l'utilisation d'une grille reliant les différentes unités entre elles.

La Figure 2.1 représente deux topologies de grilles les plus couramment utilisées : une grille unidimensionnelle en ficelle et une grille bidimensionnelle rectangulaire. Pour chacun de ces deux exemples, les prototypes sont représentés par des ronds reliés entre eux par des segments de droite. Outre l'intérêt immédiat du support visuel, l'apport principal d'une grille est de pouvoir définir une relation de voisinage entre chaque prototype et tous les autres prototypes. Pour définir cette relation de voisinage, les prototypes sont numérotés suivant leur position dans la grille, de haut en bas et de gauche à droite. Prenons le cas de la grille en ficelle, dans la partie gauche de la Figure 2.1. Sur cette grille, on définit les voisins du prototype 6, par exemple, représenté en noir. Les voisins de 6, à une distance de 1 sur la grille, sont 5 et 7 (en gris foncé). Les voisins à une distance de 2 sont 4 et 8 (en gris clair). Si on définit la limite de voisinage comme la distance maximale au-delà de laquelle les autres prototypes ne font plus partie du voisinage d'un prototype donné, et qu'on fixe dans cet exemple la limite de voisinage à 2, les prototypes 1, 2 et 3 (en blanc) ne sont donc pas dans le voisinage du prototype 6. De manière similaire, on définit un voisinage pour chacun des prototypes de cette grille. Concernant la grille rectangulaire de droite, où les voisinages considérés sont toujours carrés, les prototypes 7, 8, 9, 10, 12, 13, 14 et 15 sont voisins à une distance de 1 du prototype 11 ; les prototypes 4, 5 et 6 sont voisins à une distance de 2 ; les prototypes 1, 2 et 3 sont au-delà de la limite de voisinage.

Concrètement, le voisinage définit donc un ensemble d'informations pour chaque prototype en spécifiant qui est dans le voisinage de qui et à quelle distance sur la grille. Ces informations sont utilisées dans la règle de modification de la position des prototypes, après sélection du vainqueur.

L'algorithme SOM se définit comme suit. Lors de la phase d'apprentissage, les données de l'ensemble $\mathcal{D}$ sont présentées successivement à l'algorithme, itération après itération. Comme pour le Competitive Learning, lors de chaque itération, le prototype le plus proche, selon la mesure de distance considérée, de la donnée courante $\mathbf{x}_i$ est sélectionné :

$$\bar{\mathbf{x}}_k \;=\; \arg \min_{1 \leq j \leq n} dist(\bar{\mathbf{x}}_j, \mathbf{x}_i), \tag{2.6}$$

où les indices i, j et k ont la même signification que ci-dessus, et $dist(.,.)$ est, ici encore, la distance euclidienne.

Tout comme pour le Competitive Learning, le prototype $\bar{\mathbf{x}}_k$ est ensuite modifié afin de représenter la donnée $\mathbf{x}_i$. La différence est qu'ici les voisins $\bar{\mathbf{x}}_j$ de ce prototype $\bar{\mathbf{x}}_k$ sont eux aussi adaptés, en fonction des contraintes de voisinage données par la grille utilisée :

$$\begin{array}{rcll} \bar{\mathbf{x}}_j(iter+1) & = & \bar{\mathbf{x}}_j(iter) + \alpha(iter)(\mathbf{x}_i - \bar{\mathbf{x}}_j(iter)) & \forall\ \bar{\mathbf{x}}_j \in \text{Voisinage}(\bar{\mathbf{x}}_k), \\ \bar{\mathbf{x}}_j(iter+1) & = & \bar{\mathbf{x}}_j(iter) & \forall\ \bar{\mathbf{x}}_j \notin \text{Voisinage}(\bar{\mathbf{x}}_k). \end{array} \tag{2.7}$$

Les paramètres $\alpha(.)$ et Voisinage(.) sont respectivement le taux d'apprentissage et la fonction de voisinage du SOM. Le taux d'apprentissage décroît, comme dans le cas du Competitive Learning, afin d'assurer la convergence (132). La fonction de voisinage est bornée par la limite de voisinage qui est également choisie de sorte qu'elle soit décroissante au fur et à mesure que le nombre d'itérations augmente. Dans toutes les applications de l'algorithme SOM qui suivent, le taux d'apprentissage décroît suivant une loi exponentielle, alors que la distance de voisinage est linéairement décroissante.

L'influence du paramètre $\alpha(.)$ est la suivante : le déplacement du prototype $\bar{\mathbf{x}}_k$ et de ses voisins dans la direction de la donnée $\mathbf{x}_i$ est d'autant plus important que la valeur de ce paramètre est élevée. Comme la position initiale des prototypes est quelconque au début de l'exécution de l'algorithme, il est utile de prendre une valeur relativement élevée de ce paramètre pour le faire diminuer rapidement vers des valeurs beaucoup plus faibles, afin de permettre aux prototypes d'une carte auto-organisée d'avoir toute la latitude pour se déplacer correctement au sein de la distribution des données.

Typiquement, la limite de voisinage qui borne la fonction Voisinage(.) est relativement grande au début de l'exécution de l'algorithme, pour diminuer et être nulle pendant les dernières présentations de l'ensemble $\mathcal{D}$. En effet, la position initiale des prototypes étant quelconque, il est utile que la position de nombreux voisins soit modifiée au début de l'exécution afin que les prototypes s'ordonnent dans la distribution des données de façon similaire aux positions qu'ils occupent sur la grille. Par la suite, les prototypes se déplacent uniquement avec leurs voisins les plus proches pour couvrir, petit à petit, l'ensemble de la distribution, pour enfin ne plus se déplacer qu'indépendamment les uns des autres afin de prendre en compte les variations locales de la distribution initiale. Notons que c'est cette phase de positionnement auto-organisée des prototypes les uns par rapport aux autres au sein de la distribution qui a donné son nom à la méthode.

L'ensemble de prototypes $\mathcal{P}$ issus d'une application de l'algorithme SOM à un ensemble $\mathcal{D}$ possède les deux propriétés de quantification vectorielle et de respect de la topologie. La quantification vectorielle est la réduction de l'ensemble des N données de départ en un ensemble de n prototypes, telle que décrite dans la Section 2.1.2. Le respect de la topologie est une propriété qui découle de l'usage d'une grille et qui s'énonce comme suit. Deux données similaires au sein de la distribution initiale, se retrouvent, après convergence de la carte auto-organisée et création des clusters, soit dans le même cluster, soit dans deux clusters dont les prototypes sont des voisins sur la grille utilisée. Intuitivement, si deux données sont similaires, les distances entre ces données et un prototype sont sensiblement les mêmes, et elles sont alors dans le même cluster. Si ces distances sont sensiblement différentes, alors une des deux données est peut être plus proche d'un autre prototype

situé près du prototype considéré, et les données se retrouvent alors dans deux clusters dont les prototypes sont voisins.

Deux paramètres supplémentaires influencent la qualité avec laquelle la distribution des données est représentée par les prototypes d'une carte auto-organisée : la forme de la grille ainsi que le nombre de prototypes sur la grille. La topologie de la grille est généralement de forme unidimensionnelle (ficelle) ou bidimensionnelle (cartes carrée ou rectangulaire). En toute généralité, une ficelle, une carte rectangulaire ou une carte carrée peuvent être utilisées à peu près indifféremment. Cependant, en pratique, le choix entre ces différentes topologies de grille est influencé par toute connaissance qu'on peut obtenir par analyse de la distribution des données.

Concernant le nombre de prototypes sur la grille, il faut noter que l'algorithme SOM ne donne pas un modèle mais une famille de modèles. En effet, pour un ensemble de données $\mathcal{D}$, il existe un très grand nombre de modèles différents qui peuvent être utilisés pour représenter cet ensemble $\mathcal{D}$: une ficelle avec deux, trois, quatre, ... prototypes, une carte rectangulaire d'une longueur de deux, trois, quatre, ... prototypes et/ou d'une largeur de deux, trois, quatre, ... prototypes. En d'autres mots, il existe autant de modèles possibles que de configurations de grilles. Parmi tous ces modèles de complexités différentes, il est judicieux de sélectionner celui qui correspond le mieux aux données de l'ensemble $\mathcal{D}$ considéré. Plusieurs approches pour comparer différents modèles issus de différentes applications de l'algorithme SOM et pour sélectionner celui dont la complexité représente le mieux possible un ensemble de données particulier sont décrites dans la Section 2.2.

2.1.4 Les réseaux à fonctions radiales de base

Contrairement au SOM, l'algorithme des réseaux à fonctions radiales de base (Radial Basis Function Network ou RBFN) (11; 122; 128) est un algorithme supervisé. Autrement dit, durant l'apprentissage, l'algorithme utilise en plus des vecteurs $\mathbf{x}_i$ un ensemble de valeurs y_i tels que l'ensemble $\mathcal{D}$ est maintenant défini par :

$$\mathcal{D} = \left\{ (\mathbf{x}_i, y_i) \mid x_i \in \mathbb{R}^d, y_i \in \mathbb{R}, 1 \leq i \leq N \right\}. \tag{2.8}$$

Cet ensemble regroupe donc des couples entrées-sortie, où la sortie est ici supposée scalaire. Pour obtenir une sortie vectorielle, il faut généraliser l'algorithme classique décrit dans (11; 128). Notons également que $\mathbf{x}(t)$ et $y(t)$ peuvent être utilisés en lieu et place de $\mathbf{x}_i$ et y_i dans le contexte temporel.

Conceptuellement, au cours de l'apprentissage, l'algorithme RBFN place un ensemble de noyaux Gaussiens au sein d'une distribution de données de sorte que la somme pondérée des valeurs fournies par les différents noyaux approxime la sortie correspondant au vecteur en entrée du modèle, en moyenne, et ce pour tous les couples entrées-sortie de l'ensemble $\mathcal{D}$.

La Figure 2.2 propose le schéma d'un réseau RBFN. Sur cette figure, les entrées sont propagées au travers des flèches aux noyaux Gaussiens $\phi_j, i \leq j \leq m$ constituant l'unique couche du réseau. Les valeurs des noyaux sont ensuite pondérées par les coefficients λ_j pour

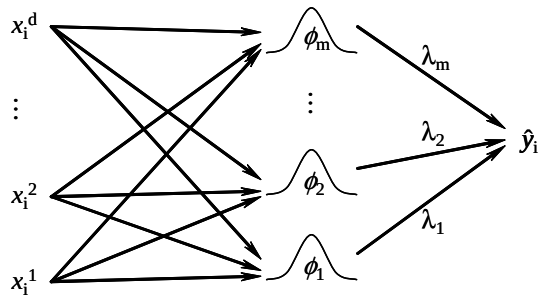

FIG. 2.2 – Schéma d'un réseau RBFN.

produire la valeur en sortie $\hat{y}_i$. Intuitivement, les coefficients λ_j représentent l'importance relative des différents noyaux dans la valeur calculée par le RBFN. Notons que le symbole $\hat{}$ représente, ici aussi, le fait qu'il s'agit d'une valeur calculée par le réseau et non de la vraie valeur y_i.

Formellement, la sortie d'un réseau RBFN est calculée comme :

$$\hat{y}_i = \sum_{j=1}^{m} \lambda_j \phi_j \left(\mathbf{x}_i, \mu_j, \sigma_j\right), \tag{2.9}$$

où les noyaux Gaussiens $\phi_j \left(., \mu_j, \sigma_j\right), 1 \leq j \leq m$ sont définis par :

$$\phi_j \left(\mathbf{x}_i, \mu_j, \sigma_j\right) = e^{-\left(\frac{\|\mathbf{x}_i - \mu_j\|}{\sqrt{2\sigma_j}}\right)^2}, \tag{2.10}$$

avec $e(.)$ la fonction exponentielle. Dans les expressions (2.9) et (2.10), les paramètres μ_j et σ_j désignent respectivement le centre et la largeur du noyau Gaussien ϕ_j, $1 \leq j \leq m$. En d'autres termes, la sortie du modèle correspond à une somme pondérée des entrées transformées non linéairement.

Très brièvement, lors de l'apprentissage, les noyaux Gaussiens sont placés au sein de la distribution des données, par exemple au moyen d'un algorithme de quantification vectorielle. Les paramètres μ_j des m différents noyaux correspondent aux prototypes issus de la quantification vectorielle. De cette quantification, on extrait également les clusters qui sont utilisés pour déterminer les largeurs des noyaux σ_j. Les coefficients de pondération λ_j des différents noyaux sont ensuite déterminés par résolution d'un système d'équations linéaires. De plus amples détails sur l'apprentissage des réseaux RBFN se trouvent par exemple dans (10; 125).

En pratique, on constate que les paramètres μ_j influencent de façon relativement limitée le résultat de l'algorithme. Les paramètres σ_j influencent fortement la qualité de l'approximation de la relation entrées-sortie fournie par l'algorithme. Il faut en effet que les différents noyaux Gaussiens se recouvrent suffisamment pour que les parties de la distribution où aucun noyau n'est venu se placer soient prises en compte par le RBFN. Cependant, il ne faut pas non plus que les noyaux soient tellement larges que chacun

d'entre eux recouvre l'ensemble de la distribution. Si tel était le cas, il n'y aurait plus grand intérêt à positionner des Gaussiennes à différents endroits de la distribution, une Gaussienne unique, très large et plate serait suffisante. Bien évidemment, l'approximation de la relation entrées-sortie serait dans ce cas très mauvaise. Les paramètres λ_j sont uniques, une fois les μ_j et σ_j fixés ; leur influence sur la sortie du modèle est directe, il s'agit du poids accordé à chaque noyau Gaussien dans la valeur de sortie du modèle.

Dans le cadre de cette thèse, parmi les nombreuses approches possibles, la méthode choisie est de sélectionner la largeur des noyaux comme étant la même pour tous les noyaux, et telle qu'elle vaut le rayon du plus grand cluster, multiplié par un facteur d'échelle (Width Scaling Factor dans (10)) fonction de l'ensemble de données.

Notons que, d'un point de vue théorique, le RBFN possède la propriété d'approximateur universel (11), ce qui signifie que l'algorithme est en mesure de fournir un réseau RBFN approximant n'importe quelle relation entrées-sortie d'un ensemble de données $\mathcal{D}$, sous condition que la complexité du RBFN soit suffisamment grande, c'est à dire sous condition qu'il y ait un nombre m de noyaux Gaussiens suffisamment élevé.

Enfin, notons également que, tout comme dans le cas de l'algorithme SOM, l'algorithme RBFN permet d'obtenir non pas un modèle, mais une famille de modèles de complexités différentes en fonction du nombre m de noyaux Gaussiens utilisés. Quelques approches permettant de comparer et sélectionner le "meilleur" modèle pour un ensemble de données particulier sont décrites dans la section suivante.

2.2 Sélection de structure de modèles

Comme mentionné aux sections 2.1.3 et 2.1.4 précédentes, les algorithmes décrivant les modèles SOM et RBFN ne définissent pas un modèle unique, mais une famille de modèles de complexités différentes. De plus, ces deux algorithmes possèdent un certain nombre de paramètres. Il est à présent nécessaire de préciser la notation $f(t, \ldots)$ qui a été utilisée jusqu'ici pour désigner un modèle. Par la suite, un modèle est représenté par la définition suivante :

$$\hat{y}(t) = f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D})\right), \tag{2.11}$$

où

- l'indice s représente les différentes structures possibles ou, en d'autres termes, les différentes complexités possibles (à savoir le nombre n pour l'algorithme SOM ou le nombre m pour le RBFN) ;
- $\theta_s(\mathcal{D})$ représente les différents paramètres du modèle de structure s dont les valeurs respectives sont déterminées lors de l'apprentissage sur l'ensemble $\mathcal{D}$;
- les autres notations conservent leur signification : $\mathbf{x}(t)$ est le vecteur des données en entrée du modèle, et $\hat{y}(t)$ est l'estimation de $y(t)$ calculée par le modèle.

Rappelons que, dans le cadre de la prédiction de séries temporelles, la sortie du modèle $\hat{y}(t)$ peut être $\hat{x}(t+1)$ ou encore $\hat{x}(t+h)$. En pratique, une question importante est d'obtenir un modèle qui corresponde globalement aux données à disposition, tout en étant

suffisamment général que pour correspondre également à d'autres données issues du même processus mais n'ayant pas servi à l'élaboration du modèle, comme le sont les valeurs à prédire. Il y a donc un double but pour un bon modèle : correspondre aux données à disposition mais aussi être capable de prédire. Un compromis est nécessaire entre ces deux buts. Le modèle réalisant le meilleur compromis entre ces deux buts sera dit optimal, et sa structure s sera sélectionné.

Pourquoi est-il aussi important de sélectionner une structure de modèle s particulière ? Comme nous venons de le voir, il y a en pratique deux types de paramètres à déterminer pour des modèles non linéaires tels que les SOM ou les RBFN : les paramètres du modèle θ_s et le paramètre de structure s. Lors de l'apprentissage, rien n'indique comment sélectionner une structure s plutôt qu'une autre : cette phase d'apprentissage n'est définie que pour une complexité fixée. Or, pour un même ensemble de données $\mathcal{D}$, la qualité de la modélisation obtenue peut varier fortement d'une structure à l'autre. Considérons le cas d'un modèle RBFN (le même raisonnement pouvant être obtenu en non supervisé pour un SOM). Si la structure de modèle est de complexité insuffisante, la relation entrées-sortie modélisée par le réseau ne sera qu'une approximation grossière et simplifiée du processus réel qui a généré les sorties correspondant aux différents vecteurs en entrée. Et rien ne dit que, sur de nouvelles données, le modèle permettra de fournir des prédictions pertinentes. A l'opposé, si la structure est trop complexe, le modèle alors obtenu sera tellement spécifique aux données de l'ensemble $\mathcal{D}$ qu'il sera alors plus proche de ces données que du processus réel dont les données ne sont qu'un aperçu. Il ne sera pas en mesure de prédire autre chose que ce qu'il a modélisé, qui n'est qu'une partie du processus réel, limité aux donnés à disposition. Dans le premier cas discuté ci-dessus, on dit que le modèle est en sous-apprentissage : il n'a pas pu apprendre toute l'information contenue dans l'ensemble de données disponible. Dans le second cas, on dit que le modèle est en sur-apprentissage : il modélise également des informations qui ne font pas partie de la véritable relation entrées-sortie, comme par exemple du bruit dû aux mesures.

En d'autres mots, un modèle avec une structure optimale est donc un compromis entre une structure trop simple conduisant à un sous-apprentissage et une structure trop complexe conduisant à un sur-apprentissage. Un tel modèle possède ce qu'on appelle une bonne capacité de généralisation : il est en mesure de reproduire la dynamique du processus au moyen de l'ensemble $\mathcal{D}$ d'apprentissage, de sorte que, pour de nouvelles données de la même série temporelle, autres que celles de l'ensemble $\mathcal{D}$, il fournisse des valeurs $\hat{y}(t)$ en sortie conforme aux valeurs réelles de la série temporelle. Qu'on soit en sous- ou en sur-apprentissage, les modèles obtenus possèdent de faibles capacités de généralisation. Il y a donc une nécessité de tester plusieurs complexités de modèle différentes sur un même ensemble de données, et de comparer leur capacité de généralisation, sur base d'un critère, afin de sélectionner le modèle optimal pour un ensemble de données particulier. Malheureusement, en pratique, on dispose de peu d'information pour se donner une idée sur la complexité à utiliser pour un ensemble de données particulier. Il faut donc faire une recherche, éventuellement exhaustive, pour tout un ensemble de modèles de complexités

différentes, entre deux bornes constituées par deux modèles de complexité s_{min} et s_{max} choisie.

Les sections suivantes vont présenter des critères permettant de mesurer les capacités de généralisation de modèles de complexité différentes, ainsi que des méthodologies permettant de comparer des modèles de complexité différentes sur base de leurs capacités de généralisation. L'accent est porté sur les capacités en généralisation, l'hypothèse commune à toutes ces méthodes étant que le modèle obtenu après apprentissage possède les paramètres θ_s idéaux pour une structure s fixée.

2.2.1 Critères de mesure de l'erreur de généralisation

En machine learning, un critère communément utilisé afin de mesurer les capacités de généralisation d'un modèle est l'erreur de généralisation. Intuitivement, ce critère mesure les écarts entre les valeurs fournies par le modèle, en sortie, et les véritables valeurs de sortie de l'ensemble $\mathcal{D}$. Cette différence est définie théoriquement pour un ensemble infini de données, ce qui permet d'obtenir l'erreur de généralisation exacte puisqu'on considère toutes les valeurs possibles. On définit l'erreur de généralisation $E(s)$ d'un modèle de structure s selon (102) :

$$E(s) = \lim_{N\to\infty} \sum_{t=1}^{N} \frac{(\hat{y}(t) - y(t))^2}{N}, \tag{2.12}$$

où N est le nombre de données de l'ensemble $\mathcal{D}$ et où $\hat{y}(t)$ est obtenu suivant la relation (2.11). La définition (2.12) n'est valable que lorsque la limite existe, ce qui peut être démontré sous certaines conditions d'invariance de la série considérée (88).

En pratique, un ensemble $\mathcal{D}$ de dimension infinie n'est pas disponible, N étant fini. On ne peut donc qu'estimer l'erreur de généralisation sur base des données disponibles, suivant :

$$\hat{E}(s) = \sum_{t=1}^{N_\mathcal{V}} \frac{(f_s(\mathbf{x}(t), \theta_s(\mathcal{D})) - y(t))^2}{N_\mathcal{V}}, \tag{2.13}$$

où la notation $N_\mathcal{V}$ représente le nombre de données de validation, concept qui sera détaillé dans la section suivante. Comme auparavant, le symbole $\hat{}$ sur le terme $E(.)$ désigne le fait qu'on estime l'erreur de généralisation théorique (2.12). Notons que ce critère est également appelé le critère de moyenne des moindres carrés (Mean Square Error ou MSE), dont il existe plusieurs variantes dans la littérature, notamment :

- Normalized MSE : $\frac{\sum_{t=1}^{N_\mathcal{V}} (\hat{y}(t)-y(t))^2}{\sum_{t=1}^{N_\mathcal{V}} (\hat{y}(t)-\mathrm{mean(y(t))})^2}$
- Root MSE : $\sqrt{\sum_{t=1}^{N_\mathcal{V}} \frac{(\hat{y}(t)-y(t))^2}{N_\mathcal{V}}}$
- Absolute Mean Error : $\sum_{t=1}^{N_\mathcal{V}} \frac{|\hat{y}(t)-y(t)|}{N_\mathcal{V}}$
- etc.

Par la suite, le critère utilisé est ce critère MSE.

2.2.2 Méthodes de validation et de cross-validation

En pratique, l'estimation de l'erreur de généralisation sur un ensemble fini de données peut s'avérer erronée si le calcul de cette estimation n'est pas fait avec précaution. En effet, utiliser un même ensemble de données $\mathcal{D}$ pour fixer les paramètres d'un modèle de structure s et pour estimer l'erreur de généralisation de ce modèle peut conduire à une mauvaise estimation de l'erreur de généralisation. Dans le cas d'un modèle en sur-apprentissage, le modèle est tellement spécifique que l'estimation de l'erreur de généralisation sur l'ensemble $\mathcal{D}$ sera sous-estimée, ce modèle décrivant très (trop) bien les données disponibles. On dit alors que l'estimation est biaisée, le biais correspondant à la différence moyenne entre d'erreur entre la valeur réelle et l'estimation fournie par la méthode. L'idée est donc de conserver une partie des données de l'ensemble de départ pour mieux estimer cette erreur de généralisation, après détermination des paramètres lors de l'apprentissage. Cette partie des données, appelée ensemble de validation, est donc utilisée pour sélectionner la complexité optimale en permettant de comparer entre eux différents modèles après apprentissage. La méthode de validation est la méthode la plus simple implémentant cette idée.

En validation, l'ensemble $\mathcal{D}$ est découpé en un ensemble d'apprentissage $\mathcal{A}$, servant à déterminer les paramètres θ_s d'un modèle de structure s fixée, et en un ensemble de validation $\mathcal{V}$, dont le rôle est de permettre le calcul de l'estimation de $\hat{E}(s)$ pour un modèle de structure s. La découpe de l'ensemble $\mathcal{D}$ en $\mathcal{A}$ et $\mathcal{V}$ se fait une seule fois pour toutes les structures s qui sont testées. L'estimation de l'erreur de généralisation est donc, dans le cas de la validation simple, fournie par :

$$\mathrm{MSE}_{\mathrm{valid}} = \hat{E}_{\mathrm{valid}}(s) = \sum_{t=1}^{N_{\mathcal{V}}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{A})\right) - y(t)\right)^2}{N_{\mathcal{V}}}, \tag{2.14}$$

où $N_{\mathcal{V}}$ est le nombre de données dans l'ensemble $\mathcal{V}$. De manière similaire, on définit $N_{\mathcal{A}}$, le nombre de données dans l'ensemble $\mathcal{A}$, de sorte que $N = N_{\mathcal{A}} + N_{\mathcal{V}}$. L'étape d'apprentissage est alors répétée pour chacune des structures s, ce qui conduit à une série de valeurs $\hat{E}_{\mathrm{valid}}(s)$ issues de la validation de chaque modèle. Le modèle sélectionné est celui dont la structure s minimise $\hat{E}_{\mathrm{valid}}(s)$.

D'un point de vue méthodologique, lorsque le nombre de données disponibles le permet, on découpe l'ensemble $\mathcal{D}$ non pas en deux mais en trois ensembles : les ensembles $\mathcal{A}$ et $\mathcal{V}$ et un dernier ensemble $\mathcal{T}$, l'ensemble de test. Cet ensemble permet d'effectuer un test du modèle sélectionné sur des données "nouvelles" qui n'ont servi ni à l'apprentissage ni à la validation. L'idée est de simuler des conditions réelles et d'observer comment se comporterait le meilleur modèle dans de telles conditions. L'usage de ces trois ensembles de données est illustré à la section 3.6 au moyen de résultats expérimentaux.

De manière générale, la proportion de données $N_{\mathcal{A}}$ par rapport à $N_{\mathcal{V}}$ fait l'objet d'un compromis entre le biais et la variance de l'estimateur $\hat{E}_{\mathrm{valid}}(.)$. En effet, le critère d'erreur des moindres carrés MSE peut être décomposé en deux termes : la décomposition biais-variance (60). Cette décomposition biais-variance a été largement étudiée dans la

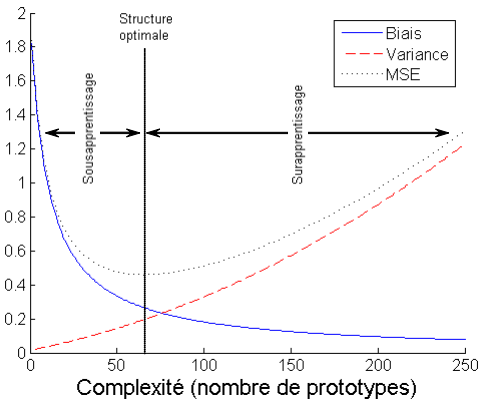

FIG. 2.3 – Forme générale des courbes de l'erreur d'apprentissage et de l'estimation de l'erreur de généralisation. L'estimation de l'erreur de généralisation est composée d'un terme de biais, correspondant à l'erreur d'apprentissage, et d'un terme de variance. La complexité -la structure- optimale est celle qui réalise le compromis entre sous-apprentissage et sur-apprentissage.

littérature, les travaux de (60) ayant été complétés entre autres par (85; 96; 168) et généralisés dans (44; 180). La décomposition biais-variance s'interprète comme suit : l'erreur de généralisation théorique est obtenue en additionnant le carré du biais et la variance de l'estimateur de l'erreur de généralisation. Le biais correspond à une mesure de la distance entre les valeurs fournies par le modèle et celles du véritable processus. Le terme de variance mesure la variabilité de la modélisation par rapport aux données d'apprentissage. Bien évidemment, le but étant de réduire l'erreur de généralisation, il faudrait idéalement réduire autant que possible le biais et la variance. Réduire le biais est très simple, il suffit d'augmenter la complexité du modèle : la distance entre les valeurs fournies par la modèle et celles de la vraie fonction diminue alors effectivement, le modèle approximant mieux le processus. Mais cette diminution du biais est accompagnée par une augmentation de la variance, et au final par une augmentation de l'erreur mesurée par le critère MSE. En effet, le biais et la variance évoluent en sens inverse comme le montre la Figure 2.3. Un estimateur fortement biaisé se situe sur la partie gauche du graphe, et correspond à un modèle de structure (trop) simple. Ce modèle est en sous-apprentissage. A l'inverse, un modèle plus complexe est non biaisé mais sa variance peut devenir importante, ce qui résulte en une augmentation de MSE et en un sur-apprentissage (121).

Tout comme en jouant sur la complexité du modèle, il est possible d'influencer le rapport biais/variance de l'estimateur de l'erreur de généralisation en modifiant la proportion de données $N_{\mathcal{A}}$ et $N_{\mathcal{V}}$. En effet, augmenter la taille de l'ensemble d'apprentissage $N_{\mathcal{A}}$ permet de réduire le biais de l'estimateur de l'erreur de généralisation. Disposant de plus d'information, on peut déterminer plus précisément les paramètres d'un modèle. L'erreur de généralisation alors estimée sera faible, le modèle étant proche du véritable

processus. Cependant, la variance de l'estimateur sera dans ce cas élevée. Etant donné l'information relativement importante à disposition, l'algorithme décrivant le modèle abstrait peut converger vers différents modèles, tous spécifiques aux données de l'ensemble $\mathcal{A}$, mais néanmoins différents. A l'opposé avec peu de données d'apprentissage, et donc peu d'information disponible, l'algorithme convergera la plupart du temps vers un même modèle, mais celui-ci sera probablement très différent du véritable processus. En d'autres termes, la variance de l'estimateur sera alors réduite, mais son biais sera important. En pratique, on utilise souvent des proportions de l'ordre de 66 à 90 % des données pour construire l'ensemble $\mathcal{A}$, le reste des données constituant l'ensemble $\mathcal{V}$ (95; 113). De telles proportions permettent d'avoir des estimations, par validation, peu biaisées mais avec une certaine variance.

Afin de réduire le biais de l'estimateur et de connaître plus précisément la variance, on peut répéter un certain nombre de fois le calcul de l'estimation de $\hat{E}_{\text{valid}}(.)$ en initialisation différemment les paramètres θ_s des algorithmes. Cependant, l'information contenue dans $\mathcal{A}$ ne changeant pas, l'algorithme devrait modéliser cet ensemble d'apprentissage de la même façon, si l'apprentissage se déroule correctement. Par conséquent, même en partant de plusieurs conditions initiales différentes, les estimations de $\hat{E}_{\text{valid}}(.)$ seront sensiblement les mêmes, l'ensemble $\mathcal{V}$ ne changeant pas non plus. La variance ne sera pas réduite. Il est donc utile de modifier le contenu des ensembles $\mathcal{A}$ et $\mathcal{V}$. Telle est l'idée de la cross-validation.

En cross-validation, la découpe de l'ensemble $\mathcal{D}$ est répétée un certain nombre de fois, le contenu des deux ensembles $\mathcal{A}_r$ et $\mathcal{V}_r$ étant modifié à chaque découpe, ou répétition r. En pratique, on sélectionne de façon aléatoire le contenu de $\mathcal{A}_r$ à l'intérieur de $\mathcal{D}$, le reste des données constituant l'ensemble $\mathcal{V}_r$. Pour chaque découpe, on estime l'erreur de généralisation des modèles au moyen de l'ensemble de validation, de sorte que l'estimation finale de l'erreur de généralisation, pour chaque structure, est la moyenne des estimations obtenues sur le nombre total de répétitions. Formellement, la cross-validation se résume suivant les étapes suivantes :

1. Création des ensembles $\mathcal{A}_r$ et $\mathcal{V}_r$,
2. Détermination des paramètres θ_s du modèle de structure s sur l'ensemble $\mathcal{A}_r$,
3. Estimation de l'erreur de généralisation du modèle de structure s sur $\mathcal{V}_r$ suivant :

$$\hat{E}_r(s) = \sum_{t=1}^{N_{\mathcal{V}_r}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{A}_r)\right) - y(t)\right)^2}{N_{\mathcal{V}_r}}, \tag{2.15}$$

4. Répétition des étapes 2. et 3. pour les structures $s \in [s_{min}..s_{max}]$, où s_{min} et s_{max} sont les complexités respectives des deux modèles servant de bornes à la recherche,
5. Répétition des étapes 1. à 4. pour un nombre total de R répétitions,

6. Détermination de l'estimation de l'erreur de généralisation par cross-validation, pour chaque structure $s \in [s_{min}..s_{max}]$ selon :

$$\hat{E}_{\text{cross-valid}}(s) = \frac{1}{R}\sum_{r=1}^{R}\hat{E}_r(s) = \frac{1}{R}\sum_{r=1}^{R}\sum_{t=1}^{N_{\mathcal{V}_r}}\frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{A}_r)\right) - y(t)\right)^2}{N_{\mathcal{V}_r}}. \tag{2.16}$$

A la fin de cette procédure, il est possible de comparer les performances des différentes structures de modèle en généralisation. Tout comme précédemment, le modèle sélectionné est celui dont la structure s minimise l'estimation de l'erreur de généralisation. Notons qu'en pratique le nombre de répétitions R est généralement de l'ordre de quelques dizaines. Par ailleurs, la proportion de données utilisées pour chacun des deux ensembles fait, ici encore, l'objet d'un compromis entre le biais et la variance de l'estimateur $\hat{E}_{Cross-Valid}(.)$. Les proportions utilisées sont généralement les mêmes que pour la méthode de validation présentée ci-dessus, conduisant ici aussi à des estimations peu biaisées mais avec une certaine variance. Bien que moins important que pour la validation, un biais d'estimation est toujours présent avec cette méthode.

Dans la littérature, la méthode de validation est également appelée méthode de validation simple. La cross-validation figure également dans la littérature sous les appellations de validation (par opposition à validation simple) ou encore Monte-Carlo cross-validation. Par ailleurs, il existe des variantes de ces deux méthodes, à savoir la k-fold cross-validation ou encore le Leave-One-Out. En k-fold cross-validation, l'idée de la cross-validation est utilisée, avec la différence que les données sont dans ce cas découpées de façon statique en k ensembles qui servent chacun tour à tour, et une seule fois, d'ensemble de validation $\mathcal{V}_r$. Le Leave-One-Out est un cas limite de la k-fold cross-validation où k = N, le nombre de données disponibles : chacune des données est utilisée une et une seule fois, tour à tour, comme ensemble de validation. Toutes ces méthodes apparaissent parfois dans la littérature sous l'appellation de méthodes de rééchantillonage. Le biais et la variance des estimateurs obtenus par ces différents méthodes de rééchantillonnage ont été étudiés et comparés dans divers travaux (46; 90; 95; 102; 108). Notons que ces travaux portent également sur le bootstrap, présenté dans la Section 2.2.3 suivante. Le bootstrap est une autre méthode de sélection de structure de modèle permettant d'estimer l'erreur de généralisation avec une variance beaucoup plus faible mais au prix d'un certain biais, au contraire des méthodes présentées dans cette section qui n'ont, pour la plupart, qu'un biais très faible mais une grande variance.

2.2.3 Bootstrap

Le bootstrap, introduit par B. Efron (47), est une méthode de rééchantillonnage particulière dans le sens où on crée de nouveaux ensembles $\mathcal{D}^*$ à partir de l'ensemble initial $\mathcal{D}$ par tirage aléatoire avec remise, où le symbole * désigne le fait que le nouvel ensemble $\mathcal{D}^*$ est un ensemble "bootstrapé". Le bootstrap permet donc de fournir un estimateur de l'erreur de généralisation, mais par une approche de rééchantillonnage différente.

Afin de simplifier la lecture des équations qui vont suivre, on définit la notation suivante :

$$\hat{E}_{\mathcal{D},\mathcal{D}'}(s) = \sum_{t=1}^{N_{\mathcal{D}'}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D})\right) - y(t)\right)^2}{N_{\mathcal{D}'}}. \tag{2.17}$$

$\hat{E}_{\mathcal{D},\mathcal{D}'}(s)$ n'est donc rien d'autre que l'estimation de l'erreur d'un modèle de complexité s appris sur un ensemble $\mathcal{D}$ et validé sur un ensemble $\mathcal{D}'$. Le mécanisme théorique du bootstrap peut se résumer selon le développement suivant :

$$\begin{aligned}
\hat{E}_{\text{bootstrap}}(.) &= \hat{E}_{\mathcal{D},\mathcal{D}'}(.) \\
&= \hat{E}_{\mathcal{D},\mathcal{D}'}(.) + \left(\hat{E}_{\mathcal{D},\mathcal{D}}(.) - \hat{E}_{\mathcal{D},\mathcal{D}}(.)\right) && (2.18)\\
&= \hat{E}_{\mathcal{D},\mathcal{D}}(.) + \left(\hat{E}_{\mathcal{D},\mathcal{D}'}(.) - \hat{E}_{\mathcal{D},\mathcal{D}}(.)\right) && (2.19)\\
&\approx \hat{E}_{\mathcal{D},\mathcal{D}}(.) + \left(\hat{E}_{\mathcal{D}^*,\mathcal{D}'^*}(.) - \hat{E}_{\mathcal{D}^*,\mathcal{D}^*}(.)\right) && (2.20)
\end{aligned}$$

Suivant la terminologie du bootstrap, l'erreur de généralisation estimée par bootstrap est la somme de deux termes. Le premier terme $\hat{E}_{\mathcal{D},\mathcal{D}}(.)$ est appelé l'*erreur apparente*. Cette erreur est obtenue sur l'ensemble d'apprentissage ; ce terme est donc biaisé. Le second terme, $\hat{E}_{\mathcal{D}^*,\mathcal{D}'^*}(.) - \hat{E}_{\mathcal{D}^*,\mathcal{D}^*}(.)$ est l'*optimisme*, qui peut être vu comme une correction de biais de l'erreur apparente. Dans le développement ci-dessus, le passage de l'étape (2.19) à l'étape (2.20) est le coeur du bootstrap, rendu possible grâce au principe du "plug-in". Selon ce principe, on peut remplacer une statistique sur une distribution de probabilités inconnue par son estimation basée sur la distribution empirique. Appliqué au bootstrap, ce principe dit qu'il est possible de remplacer des ensembles $\mathcal{D}$ par des ensembles bootstrapés $\mathcal{D}^*$.

Concrètement, le bootstrap se déroule comme suit :

1. Estimation de l'erreur apparente sur l'ensemble initial $\mathcal{D}$:

$$\hat{E}_{\mathcal{D},\mathcal{D}}(s) = \sum_{t=1}^{N} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D})\right) - y(t)\right)^2}{N}, \tag{2.21}$$

2. Répétition de l'étape 1. pour les différentes structures s, $s \in [s_{min}..s_{max}]$, où s_{min} et s_{max} sont les complexités respectives des deux modèles servant de bornes à la recherche,
3. Création de l'ensemble bootstrapé $\mathcal{D}_r^*$ contenant $N_{\mathcal{D}_r^*} = N$ données, pour la répétition, ou *réplication* dans la terminologie bootstrap, r en cours, par tirage aléatoire avec remise,
4. Détermination des paramètres θ_s du modèle de structure s sur l'ensemble $\mathcal{D}_r^*$,
5. Estimation de l'erreur d'apprentissage du modèle de structure s sur $\mathcal{D}_r^*$ suivant :

$$\hat{E}_{\mathcal{D}_r^*,\mathcal{D}_r^*}(s) = \sum_{t=1}^{N_{\mathcal{D}_r^*}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D}_r^*)\right) - y(t)\right)^2}{N_{\mathcal{D}_r^*}}, \tag{2.22}$$

6. Estimation de l'erreur de généralisation du modèle de structure s sur $\mathcal{D}$ suivant :

$$\hat{E}_{\mathcal{D}_r^*,\mathcal{D}}(s) = \sum_{t=1}^{N_\mathcal{D}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D}_r^*)\right) - y(t)\right)^2}{N_\mathcal{D}}, \tag{2.23}$$

où, bien évidemment, $N_\mathcal{D} = N$,

7. Calcul de l'optimisme pour la structure de modèle s et pour la réplication en cours :

$$\hat{E}_{\mathcal{D}_r^*,\mathcal{D}}(s) - \hat{E}_{\mathcal{D}_r^*,\mathcal{D}_r^*}(s), \tag{2.24}$$

8. Répétition des étapes 4. à 7. pour toutes les structures s considérées,
9. Répétition des étapes 3. à 8. pour un nombre total de R réplications,
10. Détermination de l'estimation de l'erreur de généralisation par bootstrap, pour chaque structure $s \in [s_{min}..s_{max}]$ selon :

$$\hat{E}_{\text{bootstrap}}(s) = \hat{E}_{\mathcal{D},\mathcal{D}}(s) + \left(\hat{E}_{\mathcal{D}_r^*,\mathcal{D}}(s) - \hat{E}_{\mathcal{D}_r^*,\mathcal{D}_r^*}(s)\right). \tag{2.25}$$

D'un point de vue théorique, l'estimateur de l'erreur de généralisation obtenu par bootstrap est biaisé. Par contre, cet estimateur possède une variance beaucoup plus petite que celle des estimateurs obtenus par validation ou cross-validation, la validation ayant lieu ici sur un ensemble bootstrapé de taille N. Tout comme pour les méthodes de validation et cross-validation, le modèle sélectionné est celui dont la complexité s est telle qu'elle minimise $\hat{E}_{\text{bootstrap}}(s)$. Bien que biaisé, la sélection de ce meilleur modèle est plus robuste dans le cas du bootstrap que dans le cas des méthodes de validation : quel que soit le nombre de réplications R, la complexité optimale sera toujours plus ou moins la même, alors qu'elle peut varier fortement avec les méthodes de validation. Enfin, notons que, dans la littérature, il existe des améliorations du bootstrap qui ont pour but de réduire le biais, telles le bootstrap .632 et le bootstrap .632+ (47). Ces deux méthodes ont donc les meilleures propriétés statistiques puisque l'estimateur de l'erreur de généralisation qu'elles proposent n'est pas biaisé et a une faible variance.

2.2.4 Fast-bootstrap

D'un point de vue opérationnel, le problème de toutes ces méthodes de sélection de structure de modèle est le fait qu'il faut répéter R fois la phase d'apprentissage, pour les différentes structures s. Or, dans le cas de modèles non linéaires, cette répétition des apprentissages peut prendre beaucoup de temps. Le fast-bootstrap a été introduit (106; 149; 150) en tant que méthode de sélection de structure de modèle ayant la particularité de demander un coût réduit en temps calcul.

Quelle que soit la méthode de sélection de structure de modèle utilisée, l'erreur de généralisation, définie en théorie, n'est jamais qu'estimée au moyen de la méthode utilisée. L'idée, pour aller plus vite, est d'approximer l'estimateur obtenu par une procédure de

bootstrap. En effet, expérimentalement, on constate que le terme d'optimisme du bootstrap, basé sur des erreurs de généralisation, augmente de façon linéaire en fonction de la structure s du modèle. En faisant l'hypothèse de linéarité de ce terme, on peut, au lieu de calculer des estimations des erreurs de généralisation pour toutes les structures de modèle avec un nombre de répétitions R élevé, estimer ces erreurs de généralisation sur un nombre réduit de quelques structures, avec un nombre de réplications R' beaucoup plus petit. On calcule alors l'optimisme pour ces quelques structures, et on interpole ensuite les valeurs du terme d'optimisme pour toutes les structures, en utilisant une simple régression linéaire. Tout comme pour le bootstrap classique, l'estimation de l'erreur de généralisation est alors obtenue en sommant l'erreur apparente, obtenue de la même manière qu'en bootstrap classique, et les résultats de la régression sur le terme d'optimisme. En effet, sous l'hypothèse de linéarité, la régression permet d'obtenir une valeur d'optimisme même pour les structures non testées. C'est également cette régression linéaire qui autorise une réduction du nombre de réplications. En effet, l'éventuelle erreur commise lors du calcul de l'optimisme pour une des structures testées est en grande partie lissée lors de la régression. Il ne faut donc plus tester qu'un minimum de structures différentes, tout en ne se limitant pas à deux valeurs même si deux valeurs sont théoriquement suffisantes pour déterminer une droite de régression.

Concrètement, le fast-bootstrap peut se résumer selon la procédure suivante :

1. Estimation de l'erreur apparente sur l'ensemble initial $\mathcal{D}$ suivant la relation (2.21),
2. Répétition de l'étape 1. pour toutes les structures $s \in [s_{min}..s_{max}]$, où s_{min} et s_{max} représentent les complexités minimales et maximales de la recherche,
3. Création de l'ensemble bootstrapé $\mathcal{D}_r^*$ par tirage aléatoire avec remise,
4. Détermination des paramètres θ_s du modèle de structure s sur l'ensemble $\mathcal{D}_r^*$,
5. Estimation de l'erreur d'apprentissage du modèle de structure s sur $\mathcal{D}_r^*$ suivant la relation (2.22),
6. Estimation de l'erreur de généralisation du modèle de structure s sur $\mathcal{D}$ suivant la relation (2.23),
7. Calcul de l'optimisme pour la réplication r en cours suivant la relation (2.24),
8. Répétition des étapes 4. à 7. pour les quelques structures s considérées,
9. Répétition des étapes 3. à 8. pour R' réplications, $R' < R$,
10. Régression afin d'obtenir l'estimation de l'optimisme pour toutes les structures $s \in [s_{min}..s_{max}]$,
11. Sommation des résultats obtenus aux étapes 2. et 10. Cette dernière étape permet de déterminer les estimations de l'erreur de généralisation $\hat{E}_{\text{fast}-\text{bootstrap}}(s)$ par fast-bootstrap, pour chaque structure s.

La question est maintenant de pouvoir valider l'hypothèse de linéarité du terme d'optimisme hypothèse qui est au coeur de la méthode de fast-bootstrap. Pour répondre à cette question, on utilise un test d'hypothèse suivant une analyse de la variance ANOVA

(112; 119). Le but est de pouvoir montrer que le terme d'optimise peut être modélisé par un polynôme d'ordre 1, soit l'équation d'une droite, alors que tout polynôme d'ordre supérieur constituerait un modèle moins adapté. Or justement, en ANOVA, le principe est de tester la différence entre la somme des résidus obtenus en utilisant deux modèles polynomiaux d'ordre différents. Un modèle polynomial s'écrit :

$$\hat{y}_i = \beta_0 + \beta_1 x_i^1 + \beta_2 x_i^2 + \ldots + \beta_o x_i^o, \tag{2.26}$$

où x_i et $\hat{y}_i$ sont respectivement les entrées et l'approximation de la sortie fournie par le modèle, et o représente l'ordre du modèle. On pose H_0, l'hypothèse nulle, telle que le modèle polynomial de l'optimisme est d'ordre o_0. L'hypothèse alternative H_1 est dès lors que le modèle polynomial est d'ordre $o_1 > o_0$. On calcule alors la somme des résidus $SR(o_j)$ des approximations $\hat{y}_i$ d'un modèle d'ordre o_j suivant :

$$SR(o_j) = \sum_{i=1}^{N} \|\hat{y}_i - y_i\|^2, \tag{2.27}$$

où N est le nombre de données. L'acceptation de l'hypothèse H_0 est conditionnée par la réussite d'un test sur la statistique de Fisher :

$$F_{o_1-o_0,N-o_1-1} = \frac{(SR(o_0) - SR(o_1))/(o_1 - o_0)}{SR(o_1)/(N - o_1 - 1)}, \tag{2.28}$$

où $F_{o_1-o_0,N-o_1-1}$ suit une loi de Fisher-Snedecor avec tres $o_1 - o_0$ et $N - o_1 - 1$ degrés de liberté. La valeur obtenue est comparée aux valeurs des tables pour un certain niveau de confiance, généralement 5%.

Dans le cadre du fast-bootstrap, cette procédure de test d'hypothèse est appliquée comme suit :

1. Application de la procédure de fast-bootstrap pour quelques modèles de structures différentes,
2. Régression d'un modèle polynomial (2.26) d'ordre $o_0 = 1$ sur base des quelques valeurs de l'optimisme obtenues pour les différentes structures considérées,
3. Régression d'un modèle polynomial (2.26) d'ordre $o_1 = 2$ sur base des mêmes valeurs,
4. Calcul des résidus pour chacun des deux modèles,
5. Calcul de la statistique de Fisher $F_{1,N-3}$ suivant (2.28),
6. Comparaison de la valeur obtenue au niveau de confiance de 5% avec les valeurs de la table de Fisher, avec $o_1 - o_0 = 1$ et $N - o_1 - 1 = N - 3$ degrés de liberté.

Cette contribution à la méthode du fast-bootstrap est illustrée ici sur la sélection du meilleur modèle dans le cadre de la prédiction à un pas de la série Santa Fe A (177). La série Santa Fe A est un benchmark classique en prédiction de séries temporelles depuis la compétition de prédiction organisée en 1991 dont elle a fait l'objet. La série est décrite en Annexe A. La figure 2.4 propose une représentation graphique de la série temporelle proposée lors de la compétition, comptant 1000 données scalaires.

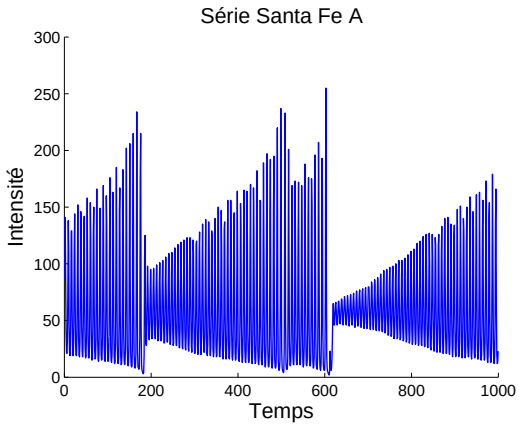

FIG. 2.4 – Série temporelle A de la compétition Santa Fe.

Dans cet exemple, la prédiction est effectuée au moyen d'un modèle supervisé RBFN. On peut dès lors préciser le contenu de l'ensemble $\mathcal{D}$, dont les éléments sont les couples composés des entrées $(x(t), x(t-1), x(t-2), x(t-3), x(t-5), x(t-6))$ et de la sortie $x(t+1)$, avec $7 \leq t \leq N$ (150). Le meilleur modèle RBFN est ici sélectionné par fast-bootstrap et par bootstrap, afin de comparer les deux approches sur une série concrète. La sélection de la structure de modèle optimale se fait sur base de l'estimation de l'erreur de généralisation définie MSE dans la relation (2.13). Dans un premier temps, une procédure de bootstrap complète a été réalisée. La correction de biais apportée par l'optimisme a été obtenue en utilisant $R = 150$ réplications. Pour chacun des 150 ensembles bootstrapés, des modèles de complexité croissante ont été testés, comprenant de 5 à 200 noyaux Gaussiens, par incrément de 5 noyaux. Au total, il y a donc eu 150 * 200/5 soit 6000 répétitions de la phase d'apprentissage d'un modèle RBFN. Par ailleurs, une procédure de fast-bootstrap a été appliquée, avec $R' = 10$ réplications. Pour cette procédure, seules les structures de modèle comptant 5, 50, 100, 150 et 200 noyaux ont été testées, ce qui représente l'apprentissage de 50 modèles RBFN, au total. La Figure 2.5 présente les résultats obtenus en utilisant les deux approches. Dans la partie gauche de cette figure, on voit le terme d'erreur apparente, ainsi que l'optimisme obtenu par bootstrap classique, les valeurs de l'optimisme obtenue par fast-bootstrap et l'estimation de l'optimisme obtenue par régression. La partie de droite de cette même figure présente les estimations de l'erreur de généralisation obtenue par bootstrap et par fast-bootstrap. On constate que l'estimation $\hat{E}_{\text{fast-bootstrap}}$ est le plus souvent supérieure à celle obtenue par bootstrap classique $\hat{E}_{\text{bootstrap}}$. Ce fait peut s'expliquer par l'utilisation d'une approximation pour le terme d'optimisme, qui est une correction de biais. La correction de biais apportée par le fast-bootstrap est moins bonne, $\hat{E}_{\text{fast-bootstrap}}$ reste donc partiellement biaisé. Cependant, ce biais est présent pour toutes les structures de modèles et n'empêche donc pas la sélection d'un minimum. Dans le cas du fast-bootstrap, la structure de modèle RBFN réalisant le minimum de $\hat{E}_{\text{bootstrap}}$ compte 70 noyaux Gaussiens, complexité qui est également sélectionnée par bootstrap classique.

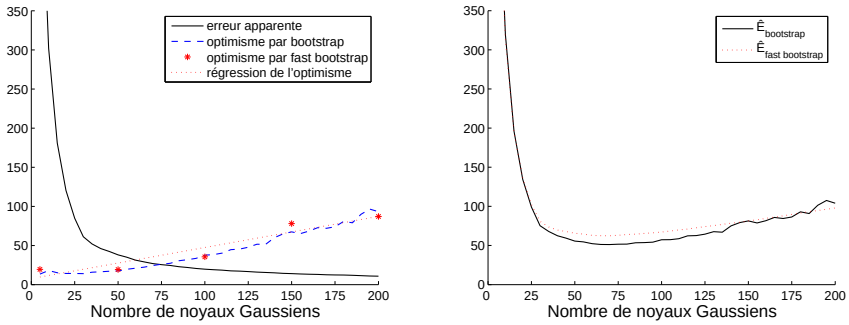

FIG. 2.5 – Sélection de structure de modèle RBFN par bootstrap et fast-bootstrap. La figure de gauche représente l'erreur apparente et les différents calculs de l'optimisme, celle de droite présente les estimations des erreurs de généralisation obtenues par les deux méthodes.

Pour cet exemple, on teste l'hypothèse $H_0 \equiv o_0 = 1$ contre l'alternative $H_1 \equiv o_1 = 2$, au niveau 5%. Le calcul de la statistique du test de Fisher nous donne une valeur de 0.7074, très nettement inférieure à la valeur de 18.51 observée dans les tables de la statistique, avec $(o_1 - o_0) = 1$ et $(N - o_1 - 1) = 2$ degrés de liberté. L'hypothèse H_0 n'est donc pas rejetée, la linéarité de l'optimisme est statistiquement validée pour cet exemple.

Notons que cette procédure de fast-bootstrap a été appliquée avec succès à plusieurs séries temporelles, afin de sélectionner la complexité de plusieurs modèles non linéaires différents, tels que les RBFN, mais aussi les MLP et les LS-SVM (162; 163; 164). Dans toutes ces expériences, l'hypothèse de linéarité de l'optimisme a toujours été validée par le test de Fisher décrit ci-dessus, quel que soit le type de modèle utilisé, même si un pré-traitement des valeurs de l'optimisme s'est montré nécessaire dans le cas du LS-SVM (105; 106).

Rappelons également que, comme illustré dans l'exemple ci-dessus, le fast-bootstrap a le double avantage de permettre 1) de ne tester qu'un nombre limité de structures et 2) avec un nombre réduit de réplications.

Enfin, concluons cette section en mentionnant que, bien que le fast-bootstrap ait été introduit comme amélioration du bootstrap classique, la procédure est tout aussi valable pour les autres versions non biaisées du bootstrap et peut donc être appliquée au bootstrap .632 et au bootstrap .632+.

2.3 Prédiction à long terme

Dans cette section, les différentes approches permettant de réaliser une prédiction à long terme sont décrites en toute généralité. La souplesse d'utilisation de la Double Quantification Vectorielle, présentée au Chapitre 3, est soulignée en étudiant la possibilité de prédire avec cette méthode à long terme suivant les différentes approches présentées ici.

2. MODÈLES, SÉLECTION DE STRUCTURE ET PRÉDICTION À LONG TERME

Soit S une série temporelle de valeurs scalaires $x(t)$, avec $1 \leq t \leq N$. Dans ce chapitre, un modèle a été défini de manière générale suivant la relation (2.11), à savoir :

$$\hat{y}(t) = f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D})\right).$$

Si on considère que $f_s(.)$ est un modèle auto-régressif, les composantes du vecteur d'entrées $\mathbf{x}(t)$ ne sont rien d'autres que les valeurs passées de la série $x(t)$, $x(t-1)$, $x(t-2)$, ... Par ailleurs, étant dans le contexte des séries temporelles, la sortie n'est autre que l'estimation de la valeur suivante de la série, à savoir $\hat{y}(t) = \hat{x}(t+1)$. On précise la définition rappelée ci-dessus comme suit :

$$\hat{x}(t+1) = f_s\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_s(\mathcal{D})\right), \tag{2.29}$$

où p est le nombre de valeurs passées prises en compte par le modèle.

La notation $\mathbf{x}_t$ utilisée jusqu'ici désigne donc dans le cas auto-régressif le vecteur comprenant les différentes valeurs passées de la série considérées dans la construction du modèle. Ce vecteur est ici appelé le régresseur et défini par :

$$\mathbf{x}_t = \left(x(t), x(t-1), \ldots, x(t-p+1)\right). \tag{2.30}$$

p est la taille du régresseur, correspondant bien évidemment à la dimension de ce vecteur. Notons que le choix de p, ainsi que du délai entre les composantes du régresseur est le sujet des chapitres 4 et 5. La description ci-dessous est proposée en supposant la dimension p et le délai quelconques, mais fixés.

Jusqu'à présent, la prédiction considérée est la valeur suivante de la série dans le cas d'une série scalaire, à savoir $x(t+1)$. En toute généralité, il est tout à fait possible de chercher à connaître une (ou plusieurs) valeur(s) future(s) plus ou moins éloignée(s) dans le temps. Considérons le cas de la prédiction d'une valeur dans le futur à l'instant $t+h$, avec h l'horizon de prédiction, $h \geq 1$. Pour prédire $x(t+h)$, deux approches sont possibles :

- l'approche *récursive*, où les prédictions obtenues selon la relation (2.29) sont répétées en avançant dans le temps jusqu'à l'horizon de prédiction final h :

$$\begin{aligned}
\hat{x}(t+1) &= f_s\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_s(\mathcal{D})\right) \\
\hat{x}(t+2) &= f_s\left(\hat{x}(t+1), x(t), \ldots, x(t-p+2), \theta_s(\mathcal{D})\right) \\
\hat{x}(t+3) &= f_s\left(\hat{x}(t+2), \hat{x}(t+1), \ldots, x(t-p+3), \theta_s(\mathcal{D})\right) \\
&\vdots \\
\hat{x}(t+h) &= f_s\left(\hat{x}(t+h-1), \hat{x}(t+h-2), \ldots, x(t-p+h), \theta_s(\mathcal{D})\right),
\end{aligned} \tag{2.31}$$

- l'approche *directe* :

$$\hat{x}(t+h) = f_s\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_s(\mathcal{D})\right). \tag{2.32}$$

Ces deux approches sont les plus couramment utilisées pour effectuer une prédiction à long terme. L'approche récursive a l'"avantage" de donner toutes les valeurs intermédiaires jusqu'à l'horizon de prédiction final. Cet "avantage" de l'approche récursive est en fait fort

relatif : on peut très bien construire plusieurs modèles en approche directe pour obtenir toutes les valeurs intermédiaires, chaque modèle servant pour un seul horizon dans le temps :

$$\begin{aligned}
\hat{x}(t+1) &= f_{s_1}\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_{s_1}(\mathcal{D})\right) \\
\hat{x}(t+2) &= f_{s_2}\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_{s_2}(\mathcal{D})\right) \\
\hat{x}(t+3) &= f_{s_3}\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_{s_3}(\mathcal{D})\right) \\
&\vdots \\
\hat{x}(t+h) &= f_{s_h}\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_{s_h}(\mathcal{D})\right).
\end{aligned} \tag{2.33}$$

Evidemment, il faut donc construire h modèles différents, ce qui implique la répétition de h procédures de sélection de structure de modèle. Et, en pratique, h répétitions d'une procédure de sélection de modèles non linéaires peut prendre beaucoup de temps.

Une alternative est de construire un modèle dont la sortie est vectorielle, ce qui permet de ne construire qu'un seul modèle qui fournit en une seule fois le vecteur des h valeurs prédites pour les différents horizons dans le temps jusqu'au terme final :

$$\{\hat{x}(t+h), \ldots, \hat{x}(t+2), \hat{x}(t+1)\} = f_s\left(x(t), x(t-1), \ldots, x(t-p+1), \theta_s(\mathcal{D})\right). \tag{2.34}$$

Bien évidemment, un modèle capable de prédire de façon vectorielle peut être utilisé de manière itérative ou directe, selon les besoins de l'application.

En général, quel que soit le modèle utilisé ($AR(p)$, $ARMA(p,q)$, $(G)ARCH(p)$, RBFN, MLP, SVM, etc.), les stratégies de prédiction à long terme récursive (2.31), directe (2.32) et directe avec plusieurs modèles (2.33) sont possibles, les modèles étant tous définis dans le cas d'une sortie scalaire. Par contre, la stratégie de prédiction vectorielle (2.34) demande quant à elle une généralisation des modèles existants pour permettre une sortie vectorielle, ce qui n'est bien souvent possible qu'au prix d'une complexité accrue.

La possibilité d'utilisation de la Double Quantification Vectorielle suivant toutes les approches décrites dans cette section est étudiée au Chapitre 3.

2.4 Résumé du chapitre

Dans ce chapitre, les modèles issus du machine learning ont été brièvement introduits, en tant qu'outils non linéaires de modélisation mathématique complémentaires aux autres modèles classiques.

Parmi les modèles existants, ceux qui sont plus spécifiquement utilisés dans cette thèse ont été détaillés : les cartes auto-organisées SOM et les réseaux à fonction radiale de base RBFN. Pour ces deux modèles, les algorithmes les décrivant respectivement ont été expliqués. Les paramètres de ces algorithmes ont été présentés et leur influence respective discutée. Les principales propriétés ont également été mentionnées.

La question du choix de la complexité optimale pour ces deux algorithmes a ensuite été abordée. Plusieurs méthodes de sélection de structure ont été décrites, dont les méthodes de validation et de bootstrap. Une première contribution a été présentée dans la description du fast-bootstrap : l'application d'une procédure ANOVA avec un test de la statistique de

Fisher, utilisé afin de tester l'hypothèse de linéarité du terme d'optimisme. Cette hypothèse s'est vue validée, procurant un argument plus théorique en faveur de la méthode de fast-bootstrap.

Les différentes approches de prédiction à long terme ont été décrites, présentant le problème de la prédiction temporelle dans un cadre plus général : il n'est pas uniquement question de prédire la valeur scalaire suivante. Suivant l'application, il peut en effet être nécessaire de prédire plusieurs valeurs dans le temps, voire plusieurs vecteurs de valeurs dans le temps.

Chapitre 3

Double Quantification Vectorielle

Dans ce chapitre, la prédiction de séries temporelles par la méthode de Double Quantification Vectorielle est introduite, notamment la prédiction de tendance à long terme. Une description générale de la méthode de prédiction est proposée, et l'emploi des cartes auto-organisées est expliqué, en tant qu'outil de base de la méthode. L'utilisation des cartes auto-organisées dans le contexte des séries temporelles est décrite au travers de la littérature existante. La méthode de Double Quantification Vectorielle est alors détaillée dans un cas simple. La méthode de prédiction est ensuite généralisée et commentée, les capacités en prédiction à long terme sont étudiées. Une preuve théorique de la stabilité de la méthode en prédiction à long terme est également proposée. Afin d'illustrer les différentes possibilités d'utilisation de la Double Quantification Vectorielle en pratique, ainsi que son intérêt en cas de prédiction à long terme, la méthode est utilisée sur quelques séries temporelles. Par ailleurs, une discussion est proposée sur les performances attendues à long terme, dans un contexte plus large que la Double Quantification Vectorielle, pour tout modèle de prédiction. Enfin, deux autres méthodes originales reprenant certaines particularités de la Double Quantification Vectorielle sont présentées en fin de chapitre. Ce chapitre se termine par un résumé des différentes contributions.

3.1 Prédiction par Double Quantification Vectorielle

La Double Quantification Vectorielle (DQV) (147; 148) est une méthode de modélisation et de prédiction de séries temporelles, certes capable de prédire des valeurs numériques à court terme, mais dont l'intérêt principal se situe dans le cadre de la prédiction à long terme. En effet, le modèle de la série obtenu par DQV permet d'obtenir des informations qualitatives sur les tendances à long terme de la série. Plus concrètement, il est possible d'établir une distribution des prédictions et d'en déduire un ensemble d'informations statistiques, telles que moyenne, intervalles de confiance, bornes, etc.

Une caractéristique de la DQV est la partition des valeurs passées de la série considérée. Ces valeurs de la série, manipulées sous forme de vecteurs, sont regroupées en clusters au moyen d'un algorithme de quantification vectorielle. Dans le cadre de la DQV, la quantification vectorielle est réalisée à deux reprises, sur deux ensembles de vecteurs différents,

comme il le sera détaillé à la section 3.3. L'idée de la méthode est non seulement d'observer les clusters de valeurs passées de la série, mais aussi de modéliser comment on passe d'un cluster à l'autre, et donc d'une valeur à l'autre dans le temps. De la sorte, il est possible de reproduire la dynamique de l'évolution de la série. En pratique, grâce à la discrétisation des valeurs passées obtenues lors de la quantification vectorielle, on obtient différents clusters. En simulant à répétition le passage d'un cluster à l'autre, on obtient une suite de prédictions ou *trajectoire*. Le modèle est en mesure de produire une collection de trajectoires différentes dont on peut tirer une série d'informations qualitatives, mentionnées ci-dessus.

Pour réaliser la quantification vectorielle, plusieurs algorithmes peuvent être utilisés, tels que notamment le Competitive Learning ou les cartes auto-organisées, décrits au chapitre 2. Dans le cadre de la DQV, le choix s'est porté sur les cartes auto-organisées. L'algorithme SOM s'exécute en effet plus rapidement que les autres algorithmes de quantification vectorielle, et ce pour une complexité algorithmique similaire (39). De même, la quantification vectorielle obtenue représente plus fidèlement les données de départ. En effet, comparé au Competitive Learning, par exemple, les cartes auto-organisées ont certes un paramètre en plus, la distance de voisinage. Cependant, la présence de la grille du SOM revient à mieux initialiser les prototypes d'un Competitive Learning, puisque dans le cas du SOM les prototypes sont liés les uns par rapport aux autres. Intuitivement, au début de l'exécution de l'algorithme SOM, le voisinage est non nul et les prototypes évoluent ensemble. Par après, lorsque le voisinage devient nul, l'exécution de l'algorithme SOM est équivalente à celle du Competitive Learning, à la différence que les prototypes sont déjà correctement positionnés au sein des données grâce à la présence de la grille.

Dans la section suivante, la littérature décrivant l'utilisation de l'algorithme SOM dans le cadre des séries temporelles est étudié. Le reste du chapitre est consacré à une description approfondie de la DQV, aux niveaux théorique et expérimental.

3.2 Cartes auto-organisées et séries temporelles

Les cartes auto-organisées sont généralement considérées comme un outil de classification non supervisée : grâce à leur propriété de quantification vectorielle, l'application de l'algorithme SOM revient à partitionner un ensemble de données au moyen de clusters. Les éléments d'un de ces clusters partagent les mêmes caractéristiques, et peuvent être considérés comme une classe à part des autres clusters obtenus par l'application de l'algorithme. Les SOM ont été utilisés dans un grand nombre d'applications de classification, telles que la classification d'images, de texte, de protéines ou de séquences ADN, de phonème, de pays sur base de facteurs économiques, etc. (2)

En plus de ces nombreuses applications en classification, sur des données vectorielles, d'autres travaux portent sur l'application du SOM à des données vectorielles ayant une dépendance temporelle. A priori, l'algorithme SOM n'est pas directement utilisable dans le contexte de données fonction du temps, et plusieurs extensions de l'algorithme ont été

proposées. En effet, la modélisation des séries temporelles revient à modéliser l'influence du passé dans l'état actuel du système et à en déduire l'évolution future du système. Or, dans la description de l'algorithme à la Section 2.1.3, rien ne permet de tenir compte de dépendances temporelles. Une classification récente, mais partielle, des travaux portant sur l'application du SOM dans le contexte temporel est proposée dans (71).

Une première approche pour prendre en compte des dépendances temporelles est le Temporal Kohonen Map (TKM) et son utilisation des "leaky integrators" (24). Les "leaky integrators" permettent de tenir compte de l'historique des vecteurs d'entrée considérés précédemment lors du calcul de la distance entre le vecteur actuel et les différents prototypes. Ce calcul de distance est en fait une somme pondérée des distances entre les vecteurs précédents et le prototype considéré, pour chaque prototype. La pondération suit une loi exponentielle décroissante au fur et à mesure qu'on s'éloigne dans le passé, le choix du prototype étant plus fonction des quelques derniers vecteurs qui viennent d'être présentés à l'algorithme que des premiers vecteurs présentés en début d'exécution de l'algorithme. La somme pondérée correspond donc à une intégration de la direction du changement d'une itération à l'autre de l'algorithme. Cette approche basée sur des "leaky integrators", couplé à des SOM hiérarchiques, est appliquée par exemple en reconnaissance de la parole (87), ou à de la reconnaissance de séquences musicales (20). L'algorithme SOM with Temporal Activity Diffusion (SOMTAD) (49; 50) utilise également les "leaky integrators" en mémorisant par ailleurs la diffusion de l'activité au travers de la structure du SOM.

Bien évidemment, cette idée de tenir compte de l'exécution passée de l'algorithme, de manière récursive, a été utilisée dans d'autres travaux, tels le Recurrent SOM (RSOM) (100). La principale différence entre le RSOM et le TKM est que le TKM n'intègre la direction du changement que lors du calcul de distance, alors que le RSOM intègre également ce changement lors de la modification de la position du prototype, le prototype conservant par ailleurs cet historique des changements pour les comparaisons ultérieures. Pour le reste, les deux algorithmes sont très proches ; une comparaison des deux approches est d'ailleurs proposée dans (170; 171). Une classification des algorithmes de type TKM et RSOM est proposée dans (123). Dans le SOM with Sequential Activation, Retention, and Delay (SARDNET) (84), l'idée de la rétention récursive de l'information est complétée par un mécanisme de désactivation temporaire des prototypes, qui ne peuvent plus être modifiés pendant un certain temps après une adaptation. L'algorithme SARDNET a de plus été utilisé avec d'autres modèles dans le cadre de la reconnaissance de dépendances temporelles à long terme (117).

Une autre approche est le Recursive SOM (RecSOM) (174). Dans cet algorithme, chaque prototype possède un vecteur poids représentant sa position dans la distribution, mais possède également un autre vecteur représentant le contexte d'activation de toute la carte lors de l'itération précédente. La sélection du prototype le plus proche se base dans ce cas sur une distance tenant compte d'une part de la différence entre la donnée et le poids du prototype et d'autre part de la différence entre le contexte précédent et le contexte du prototype. Bien entendu, la mise à jour du prototype nécessite la modification du poids

et du contexte du prototype vainqueur et de ses voisins. Notons que le Contextual SOM (CSOM) (173) est en fait le même algorithme que le RecSOM.

L'algorithme Merge SOM (MSOM) (156; 159) est très proche de l'algorithme RecSOM. Dans cet algorithme, les prototypes possèdent également un vecteur de poids et un vecteur de contexte. La différence principale réside dans la définition du vecteur de contexte. Dans le RecSOM, le vecteur de contexte représente l'activation des différents prototypes de la carte lors de l'itération précédente. Dans le MSOM, le contexte est constitué d'une combinaison linéaire entre le poids à l'itération précédente et le contexte à l'itération précédente. Le contexte est donc plus riche, mais aussi plus complexe.

Il est par ailleurs possible de particulariser le SOM for Structured Data (SOMSD) (70; 155) pour que la structure considérée soit la structure temporelle. Le SOMSD a été proposé pour des structures d'arbre, en modifiant la mesure de distance utilisée dans l'algorithme SOM classique. Il est possible de préciser cette distance pour le cas particulier de données structurées temporellement, en utilisant la position des prototypes dans la grille qui est effectivement une structure, fixée a priori. En tenant compte de la position dans la grille du prototype vainqueur à l'itération précédente, l'algorithme SOMSD peut être utilisé dans le contexte temporel et devient l'algorithme SOM for Sequences, SOM-S (156; 160). De même, l'extension du SOMSD à des topologies de grilles plus générales avec des voisinages triangulaires, tels que définit dans le Hyperbolic SOM (HSOM) (130), conduit au HSOM-S (156; 160).

Face à ces algorithmes proposés dans des contextes différents, mais reprenant pour la plupart les mêmes concepts, il a été proposée une taxonomie pour classer les réseaux connexionnistes non supervisés développés pour le contexte spatio-temporel (36; 38). De manière indépendante, il a été proposé d'unifier ces différents algorithmes au travers d'un contexte plus général, le General SOM for Structured Data (GSOMSD) (72; 73), défini pour le traitement de tout graphe acyclique, donc entre autres pour le cas des séquences temporelles.

Tous les algorithmes brièvement décrits ci-dessus modifient l'algorithme SOM soit en intégrant l'information des itérations passées dans l'itération en cours, soit en ajoutant un contexte aux poids des prototypes, sous une forme ou une autre. D'autres travaux utilisent l'algorithme SOM tel quel, afin de créer des classes de régresseurs (28; 107), ou éventuellement en utilisant un modèle local de prédiction à l'intérieur de chaque cluster, le modèle de prédiction étant linéaire (172; 175) ou non linéaire (34). De même, il est possible de compléter les approches récursives avec des modèles locaux, comme par exemple le RecSOM avec modèles locaux linéaires (99) ou avec modèles locaux de types k-plus-proches-voisins (6). En outre, il est possible de modifier les vecteurs en entrée du SOM pour qu'ils prennent en compte la sortie attendue du modèle, soit pour effectuer une prédiction directe (comme dans le Vector-Quantized Temporal Associative Memory (VQTAM) (37)), soit pour servir à un modèle local supervisé (34). Par ailleurs, d'autres approches utilisent les SOM pour prédire un vecteur de valeurs au moyen du prototype d'un des clusters (31; 135; 136) ou en combinaison avec des modèles linéaires ou de type MLP (30).

D'autres algorithmes de quantification vectorielle ont également été utilisés dans le cadre des données temporelles. Mentionnons par exemple les travaux utilisant le Neural Gas (116; 158), le Learning Vector Quantization (LVQ) (120) ou une version pondérée de ce LVQ (Generalized Relevance Learning Vector Quantization) (156; 157).

Dans cette thèse, la méthode de prédiction proposée se base sur les SOM, en utilisant une approche encore différente de toutes les approches mentionnées dans cette section. En effet, pour cette méthode, l'algorithme SOM n'est pas modifié mais utilisé à deux reprises, sur deux ensembles de données vectorielles distincts. Le résultat obtenu après ces deux quantifications est complété au moyen d'une structure de données qui permet de relier l'information obtenue lors de cette double application de l'algorithme SOM.

3.3 La Double Quantification Vectorielle

La Double Quantification Vectorielle (DQV) est une méthode de modélisation et de prédiction de séries temporelles basée sur les cartes auto-organisées (SOM) et développée spécifiquement dans le cadre de la prédiction à long terme (147; 148).

La méthode DQV modélise deux informations distinctes : d'une part les valeurs passées de la série sous forme de régresseurs et d'autre part les déformations passées de la série. Une déformation est définie comme la différence entre deux régresseurs de valeurs passées, considérées chronologiquement, ce qui correspond intuitivement à l'incrément permettant de passer d'une valeur temporelle à une autre. La DQV est donc un modèle auto-régressif non linéaire : elle n'utilise que des régresseurs et des déformations de valeurs passées de la série, partitionnés en une série de clusters disjoints. Les informations contenues dans les deux ensembles de régresseurs et de déformations sont ensuite quantifiées séparément au moyen de l'algorithme SOM. Le lien entre les deux applications du SOM est conservé dans une matrice des transitions. La matrice des transitions indique comment passer d'un régresseur à une déformation. C'est donc grâce à cette matrice des transitions que le modèle construit par DQV peut être utilisé en prédiction, à court et à long terme.

Par la suite, la phase d'apprentissage du modèle est décrite. Ensuite, l'utilisation du modèle obtenu en prédiction est expliquée. Afin d'alléger les notations et d'améliorer la lisibilité de cette section, la méthode est présentée dans le cas scalaire, pour de la prédiction à un pas, c'est à dire pour $h = 1$. L'utilisation de la méthode en prédiction à long terme, et en toute généralité, fait l'objet de la Section 3.3.3 suivante.

3.3.1 Phase d'apprentissage

Soit S une série temporelle de valeurs scalaires $x(t)$, avec $1 \leq t \leq N$. De cette série temporelle, on construit des régresseurs, suivant la relation (2.30) rappelée ici :

$$\mathbf{x}_t = (x(t), x(t-1), \ldots, x(t-p+1)) \,.$$

Notons que dans le cadre de la présentation de la DQV, la taille du régresseur, notée p, ainsi que le délai entre valeurs temporelles sont considérées comme étant quelconques,

mais fixes. Lors de l'application de la méthode à quelques séries temporelles réelles, la taille du régresseur et la valeur du délai seront choisies au cas par cas en fonction de la série temporelle considérée. La question du choix de ce délai et son importance sont étudiées aux chapitres suivants.

Les régresseurs $\mathbf{x}_t$ peuvent être manipulés afin de créer les déformations $\mathbf{z}_t$, définies selon :

$$\mathbf{z}_t = \mathbf{x}_{t+1} - \mathbf{x}_t. \tag{3.1}$$

Intuitivement, les déformations $\mathbf{z}_t$ contiennent une description de la modification du processus qui a permis de passer d'un état décrit par $\mathbf{x}_t$ à l'instant t en un état suivant $\mathbf{x}_{t+1}$ à l'instant $t+1$. Par définition, pour chaque instant t, il existe une déformation $\mathbf{z}_t$ à laquelle est associée un et un seul régresseur $\mathbf{x}_t$, et réciproquement. C'est grâce à cette relation unissant déformations $\mathbf{z}_t$ et régresseurs $\mathbf{x}_t$, à chaque instant t, que peut être construite la matrice des transitions.

Jusqu'à présent, la double application du SOM a permis de construire un ensemble $\mathcal{X}$ de $N-p+1$ régresseurs $\mathbf{x}_t$ et un second ensemble $\mathcal{Z}$ comptant $N-p$ déformations. L'algorithme des cartes auto-organisées est ensuite appliqué, séparément, sur chacun des deux ensembles $\mathcal{X}$ et $\mathcal{Z}$. De par sa propriété de quantification vectorielle, l'algorithme SOM conduit à une partition des régresseurs et à une partition des déformations. Ces deux partitions sont obtenues au moyen des clusters issus de la quantification vectorielle.

Suite à l'application du SOM à l'ensemble $\mathcal{X}$, on crée les clusters de régresseurs associés aux différents prototypes, notés $\bar{\mathbf{x}}_i$ $(1 \leq i \leq n_1)$, les clusters de régresseurs étant notés c_i. De manière similaire, pour les déformations, on crée des clusters c'_j associés aux différents prototypes $\bar{\mathbf{z}}_j$ $(1 \leq j \leq n_2)$, issus de la seconde quantification par l'algorithme SOM. A ce stade, l'information contenue dans ces deux ensembles de vecteurs est résumée de manière statique au moyen de deux ensembles de prototypes et de deux ensembles de clusters. La matrice des transitions permet de recréer la dynamique de l'évolution de la série temporelle en reliant l'information contenue dans ces différents ensembles.

La matrice des transitions $T(.,.)$ est construite afin de décrire la relation entre les régresseurs $\mathbf{x}_t$ de tous les clusters c_i et les déformations $\mathbf{z}_t$ de tous les clusters c'_j qui sont associées aux régresseurs $\mathbf{x}_t$; chaque $\mathbf{z}_t$ correspondant, pour rappel, à un et un seul $\mathbf{x}_t$ pour t fixé. Cette matrice $T(.,.)$ est définie selon :

$$T(i,j) = \frac{\#\{\mathbf{x}_t \in c_i \text{ tels que } \mathbf{z}_t \in c'_j\}}{\#\{\mathbf{x}_t \in c_i\}}, \tag{3.2}$$

avec $1 \leq i \leq n_1$ et $1 \leq j \leq n_2$. La formule (3.2) ci-dessus s'interprète comme la probabilité d'obtenir, à l'instant t, une déformation $\mathbf{z}_t$ du cluster c_j sachant que le régresseur considéré $\mathbf{x}_t$ est dans le cluster c_i. La matrice des transitions peut dès lors être interprétée comme une approximation discrète des lois conditionnelles entre les régresseurs et les déformations qui leur sont associées. Chaque ligne i décrit la loi de probabilité conditionnelle propre à chaque cluster c_i, $1 \leq i \leq n_1$. En pratique, ces probabilités sont estimées par les fréquences empiriques observées sur base des régresseurs et des déformations obtenus de la série

temporelle considérée. Bien évidemment, pour un i fixé, la somme des éléments $T(i,j)$, $1 \leq j \leq n_2$, de la ligne correspondante vaut 1.

C'est à ce niveau que réside le point principal de la méthode de Double Quantification Vectorielle. En effet, de par la propriété de quantification vectorielle de l'algorithme SOM, on sait que les régresseurs d'un cluster c_i, avec i fixé, sont semblables entre eux. Or, les déformations associées aux régresseurs de ce cluster sont elles aussi relativement semblables entre elles, pour autant que la série temporelle ne soit pas totalement aléatoire. Dès lors, les déformations se trouvent dans le même cluster, ou alors dans des clusters voisins les uns des autres. Au niveau du modèle, tout ceci se traduit par la présence attendue d'un certain nombre de zéros dans la matrice $T(.,.)$, comme illustré dans la section 3.6.

On résume la phase d'apprentissage de la Double Quantification Vectorielle comme suit :

1. Création des régresseurs $\mathbf{x}_t$ suivant la relation (2.30),
2. Application de l'algorithme SOM à l'ensemble des régresseurs $\mathcal{X}$,
3. Création des clusters de régresseurs c_i, $1 \leq i \leq n_1$, selon les prototypes obtenus à l'étape 2.,
4. Création des déformations $\mathbf{z}_t$ suivant la relation (3.1),
5. Application de l'algorithme SOM à l'ensemble des déformations $\mathcal{Z}$,
6. Création des clusters de déformations c'_j, $1 \leq j \leq n_2$, selon les prototypes obtenus à l'étape 5.,
7. Création de la matrice des transitions $T(.,.)$ selon la relation (3.2).

La section suivante présente la façon dont cette matrice est utilisée pour prédire l'évolution de la série temporelle sur base des clusters et des lois de probabilités conditionnelle de la table des transitions construits suivant la procédure ci-dessus, et comment on peut obtenir des statistiques sur cette évolution.

3.3.2 Phase de prédiction

Lorsque la phase d'apprentissage est terminée, on dispose de n_1 prototypes $\bar{\mathbf{x}}_i$ pour l'ensemble des régresseurs $\mathcal{X}$, de n_2 prototypes $\bar{\mathbf{z}}_j$ pour l'ensemble des déformations $\mathcal{Z}$ et d'une matrice des transitions. Sur base de ces éléments, il est possible de prédire l'évolution d'une série temporelle, à court terme et à long terme.

Supposons que nous soyons à l'instant t dans le temps. La dernière valeur connue de la série temporelle est $x(t)$ et le régresseur correspondant à cette dernière valeur connue est $\mathbf{x}_t$. On cherche maintenant le prototype $\bar{\mathbf{x}}_k$ le plus proche de ce régresseur $\mathbf{x}_t$ parmi tous les $\bar{\mathbf{x}}_i$, $1 \leq i \leq n_1$, où "le plus proche" signifie le plus proche au sens de la distance Euclidienne utilisée dans l'algorithme SOM. Or, on sait, par la quantification vectorielle et les clusters qui en découlent, que les régresseurs les plus semblables à $\mathbf{x}_t$ sont dans le cluster du prototype $\bar{\mathbf{x}}_k$. De plus, on sait quelles sont les déformations associées aux régresseurs du cluster c_k de ce prototype k. Cette information est résumée dans la matrice des transitions.

On peut donc retrouver une déformation probable pour le régresseur de départ $\mathbf{x}_t$ au travers des prototypes de l'ensemble $\mathcal{Z}$. Pour cela, il suffit de sélectionner un prototype $\bar{\mathbf{z}}_l$ parmi tous les prototypes pouvant être atteints à partir des régresseurs du cluster c_k. Le choix de ce prototype de déformation est effectué sur base de la loi de probabilité conditionnelle qui se trouve dans la ligne k de la matrice $T(.,.)$, les différents prototypes $\bar{\mathbf{z}}_j$ étant sélectionnés suivant leur fréquence empirique respective $T(k,j)$. Une fois cette déformation sélectionnée, on obtient le régresseur pour l'instant $t+1$ en "inversant" la relation définissant les déformations (3.1) :

$$\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t + \bar{\mathbf{z}}_l. \tag{3.3}$$

Puisque $\hat{\mathbf{x}}_{t+1}$ est lui-même un vecteur, on en extrait la valeur de la première composante qui correspond à $\hat{x}(t+1)$, suivant la définition (2.30). De cette manière, il est possible d'obtenir une prédiction unique à un horizon $h = 1$, pour une série scalaire. La prédiction finale à l'horizon $h = 1$ est obtenue en prenant la moyenne de plusieurs prédictions uniques à l'horizon $h = 1$. En effet, le choix de la déformation est basé sur une loi de probabilité. Le tirage est donc répété un certain nombre de fois au travers d'une procédure de type Monte-Carlo, afin que le choix des déformations soit conforme aux probabilités de la loi conditionnelle.

Pour obtenir une prédiction à long terme, soit $h > 1$, on répète la prédiction à un pas jusqu'à l'horizon final h de la prédiction. Cette répétition jusqu'à l'horizon h constitue une *simulation* de l'évolution de la série.

Telle que décrite jusqu'ici, la méthode permet donc d'obtenir des prédictions à court et à long terme au travers d'une simulation de l'évolution de la série temporelle. Or, on peut obtenir d'autres informations sur l'évolution de la série temporelles, par exemple en répétant la procédure de simulation. Le tirage des déformations peut en effet varier. En répétant les simulations, on peut obtenir un ensemble de simulations, dont il est possible d'obtenir une série d'informations qualitatives, telles que la moyenne de ces simulations, la variance, des intervalles de confiance ou encore des bornes, etc.

Pour conclure, voici une description synthétique de la prédiction scalaire de $\hat{x}(t+1)$ au moyen de la Double Quantification Vectorielle :

1. Création du régresseur $\mathbf{x}_t$ pour le dernier instant t connu,
2. Sélection du prototype $\bar{\mathbf{x}}_k$ le plus proche du régresseurs $\mathbf{x}_t$, parmi les prototypes $\bar{\mathbf{x}}_i$, $1 \leq i \leq n_1$,
3. Sélection du prototype $\bar{\mathbf{z}}_l$ selon la loi décrite dans la ligne k de la matrice $T(.,.)$,
4. Estimation du régresseur $\hat{\mathbf{x}}_{t+1}$ suivant la relation (3.3),
5. Extraction de $\hat{x}(t+1)$, la première composante du régresseur estimé $\hat{\mathbf{x}}(t+1)$.

Jusqu'à présent, pour obtenir une prédiction à l'horizon $h > 1$, il faut donc itérer h fois la prédiction à l'horizon 1. La méthode de Double Quantification Vectorielle est toutefois bien plus souple puisqu'elle permet de prédire à long terme suivant les différentes approches détaillées dans la section 2.3, et ce moyennant une modification des vecteurs manipulés mais sans aucune modification particulière de l'algorithme en lui-même.

3.3.3 Prédiction à long terme par Double Quantification Vectorielle

La prédiction par Double Quantification Vectorielle a été décrite à la Section 3.3.2 ci-dessus dans le cas scalaire. Elle permet donc de prédire à long terme suivant l'approche récursive (2.31).

Concernant la prédiction directe (2.32), la manière la plus simple, dans le cas d'un modèle supervisé, est de donner d'une part le régresseur en entrée du modèle et d'autre part $x(t+h)$ en sortie du modèle. De cette façon, l'algorithme supervisé apprend directement la relation entre les entrées et la sortie pour l'horizon de prédiction h considéré. Malheureusement, la DQV est basée sur l'algorithme SOM, qui est non supervisé. Il est toutefois possible d'adapter l'idée pour construire des "régresseurs" contenant toujours les valeurs passées de la série mais aussi la sortie à l'horizon h :

$$\mathbf{x}_t = \{x(t+h), x(t), x(t-1), \ldots, x(t-p+2)\}\,, \tag{3.4}$$

mais cette approche n'est toutefois pas adaptée pour la DQV. En effet, il faudrait connaître, à l'instant t, la valeur $x(t+h)$ afin de pouvoir construire le régresseur selon (3.4), ce qui n'est pas possible. De plus, la prédiction obtenue en application la procédure de prédiction de la DQV serait alors $\hat{x}(t+h+1)$ et non $\hat{x}(t+h)$ comme souhaité.

Toutefois, la DQV est bien plus souple : il suffit en fait de manipuler simplement les déformations au moment de leur construction pour tenir compte de l'horizon de prédiction h :

$$\mathbf{z}_t = \mathbf{x}_{t+h} - \mathbf{x}_t. \tag{3.5}$$

De cette manière, les déformations décrivent non plus comment passer d'une valeur à la suivante, mais comment passer d'une valeur à celle située à un horizon h. La prédiction directe suivant l'approche (2.32) est donc possible, sans modification particulière de l'algorithme, si ce n'est une construction appropriée des déformations décrite ci-dessus (3.5). Bien entendu, lorsque cette construction des déformations est utilisée, elle n'est valable que pour un horizon h fixé. Il faut donc construire autant de modèles que d'horizons de prédiction, en utilisant à chaque fois des déformations différentes, lorsqu'on souhaite prédire à long terme suivant une approche directe avec plusieurs modèles (2.33).

Par ailleurs, la Double Quantification Vectorielle peut être généralisée aisément au cas vectoriel, rendant possible la prédiction vectorielle (2.34). En effet, la DQV étant définie sur base de régresseurs de données scalaires, il suffit de construire des régresseurs de données vectorielles pour que la méthode soit directement applicable au cas de la prédiction vectorielle (2.34). En d'autres mots, tout comme ils étaient la concaténation de plusieurs scalaires dans (2.30), les régresseurs vectoriels sont définis comme la concaténation de plusieurs vecteurs selon :

$$\mathbf{X}_t = \{\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), \ldots, \mathbf{x}(t-q+1)\}\,, \tag{3.6}$$

où chaque vecteur $\mathbf{x}(t) = \left(x^1(t), x^2(t), x^3(t), \ldots, x^d(t)\right)$ selon la relation (1.2), et où q est la taille du régresseur vectoriel $\mathbf{X}_t$. De la même manière, si on considère des vecteurs dans le temps, il suffit de concaténer des régresseurs de scalaires (2.30) selon :

$$\mathbf{X}_t = \left\{\mathbf{x}_t, \mathbf{x}_{t-p}, \mathbf{x}_{t-2p}, \ldots, \mathbf{x}_{t-(q-1)*p}\right\}. \tag{3.7}$$

Dès lors, on obtient des déformations vectorielles en généralisant la relation (3.1) suivant :

$$\mathbf{Z}_t = \mathbf{X}_{t+h} - \mathbf{X}_t. \tag{3.8}$$

Dans le cas où $h = 1$, on effectue de la prédiction vectorielle à un pas, soit la prédiction du vecteur suivant, de dimension d dans le cas de données indépendantes ou de dimension p dans le cas de données temporelles. Bien évidemment, sur base de la relation (3.8), on voit qu'il est possible d'utiliser la DQV pour des prédictions vectorielles avec $h > 1$.

Le reste de la DQV se généralise directement au cas vectoriel : l'algorithme SOM est appliqué à deux reprises, aux deux ensembles de régresseurs vectoriels et de déformations vectorielles. On obtient donc des prototypes et des clusters, tout comme précédemment. La définition de la matrice des transitions se généralise elle aussi sur base des clusters de régresseurs vectoriels et des clusters de déformations vectorielles. La procédure de simulation est la même que dans le cas scalaire, à la seule différence qu'on extrait maintenant un vecteur de d (ou p) composantes de la prédiction vectorielle :

$$\hat{\mathbf{X}}_{t+h} = \mathbf{X}_t + \bar{\mathbf{Z}}_l, \tag{3.9}$$

en lieu et place d'un unique scalaire.

Une propriété intéressante de la DQV peut maintenant être soulignée, dans le cas de la prédiction de vecteurs. Grâce à l'utilisation d'un algorithme de quantification vectorielle, la DQV permet d'obtenir une prédiction de qualité identique pour *toutes* les composantes du vecteur prédit. En effet, toutes les composantes des vecteurs, qu'il s'agisse des régresseurs ou des déformations, ont le même poids dans le calcul des distances du SOM, comme c'est d'ailleurs le cas pour tout algorithme de quantification vectorielle. Autrement dit, aucune composante n'a plus d'importance que les autres, que ce soit lors de la quantification, lors de la création des clusters ou lors de la recherche du prototype le plus proche d'un certain régresseur. Les autres étapes de la DQV n'étant que des manipulations pour additionner ou soustraire des vecteurs entre eux, composante à composante, ou pour prendre la moyenne des simulations, on s'attend effectivement à obtenir le même niveau de précision pour chaque composante d'une prédiction vectorielle.

Pour résumer, on retiendra que la méthode de Double Quantification Vectorielle est en mesure de prédire aussi bien à court terme qu'à long terme, en scalaire ou en vectoriel. L'algorithme décrivant la méthode ne change pas ni en fonction de l'horizon de prédiction ni en fonction de la dimension des prédictions à fournir. Seules les constructions des régresseurs et des déformations doivent se faire de façon appropriée. La présentation de la méthode proposée dans la Section 3.3 n'est donc qu'un cas particulier de l'algorithme, dans le cas où les vecteurs d'entrée sont issus d'une série scalaire de dimension $d = 1$ et lorsque l'horizon de prédiction est fixé à $h = 1$.

3.4 Commentaires sur la Double Quantification Vectorielle

La Double Quantification Vectorielle est un modèle auto-régressif et non linéaire de prédiction de séries temporelles. Comme pour tout modèle auto-régressif, on fait l'hypothèse initiale qu'il existe une relation entre les valeurs passées et la(les) valeur(s) future(s) de la série, permettant d'obtenir une estimation d'une (ou plusieurs) valeur(s) future(s) au travers d'une modélisation de ces valeurs passées. Comme tout modèle auto-régressif, la DQV utilise des régresseurs en ce sens. Il faut noter que la taille de ces régresseurs est un élément important de la méthode. Si cette taille est trop petite, les régresseurs ne contiennent pas suffisamment d'information et le lien avec la(les) valeur(s) future(s) est perdu. Pour éviter que l'hypothèse initiale ne soit invalidée, il est donc nécessaire d'utiliser des régresseurs de taille suffisamment grande. En outre, à la différence d'un modèle $AR(p)$ ou RBFN, où ce sont les coefficients et/ou les paramètres du modèle qui permettent d'approximer la relation entre les valeurs passées et futures, le lien entre ces valeurs est modélisé dans la DQV dans les lignes de la matrice des transitions. Le principe de cette matrice est que chaque ligne représente une dynamique de transition particulière pour un ensemble de régresseurs d'un même cluster. Intuitivement, les transitions pour ces régresseurs relativement semblables entre eux doivent donc être limitées en nombre. Sous l'hypothèse initiale d'une relation entre valeurs passées et futures, on peut donc raisonnablement s'attendre à ce que certains éléments $T(i, j)$ de la matrice des transitions aient une valeur très petite, voire nulle, et ce sur chaque ligne de la matrice, ce qui rend les matrices de transitions creuses. Dans le cas contraire, on pourrait alors choisir à peu près n'importe quel prototype de déformations pour un régresseur particulier, et il n'y aurait alors aucune relation entre les valeurs passées et futures de la série. Dans ce cas également, l'hypothèse initiale se verrait invalidée. Toutefois, en pratique, dans toutes les expériences réalisées sur différentes séries temporelles, la présence d'éléments $T(i, j)$ de valeur nulle a toujours été constatée.

Les paramètres d'un modèle de prédiction obtenu par la DQV sont bien entendu les paramètres des deux cartes auto-organisées utilisées dans le modèle. En plus de ces paramètres, les valeurs n_1 et n_2, relatives aux nombres de prototypes utilisés dans les deux SOM, doivent également être déterminées. Bien évidemment, des valeurs différentes pour n_1 et n_2 conduisent à différentes partitions de l'ensemble des régresseurs et de l'ensemble des déformations, ce qui donne au final différents modèles de la série temporelle. En effet, les distributions conditionnelles de la matrice sont à chaque fois distinctes. D'un point de vue conceptuel, ces deux valeurs n_1 et n_2 discrétisent la relation entre les valeurs passées et les valeurs futures. Augmenter le nombre de lignes et de colonnes permet d'obtenir une approximation plus fine de la relation ainsi discrétisée, mais avec le risque d'un sur-apprentissage. Il y a donc un compromis à faire au niveau des valeurs n_1 et n_2 constituant la structure d'un modèle obtenu par DQV. Une des méthodes de sélection de structure de modèle, décrites dans la Section 2.2, peut être utilisée afin d'optimiser ces paramètres de structure n_1 et n_2 sur un ensemble de données particulier. Notons que la méthode de

sélection de structure devra ici être appliquée $n_1 * n_2$ fois puisqu'il faut pouvoir tester les combinaisons entre les deux valeurs.

La forme de la grille utilisée lors des deux utilisations du SOM n'est pas réellement critique, il est possible d'utiliser aussi bien des grilles unidimensionnelles que des grilles bidimensionnelles. Néanmoins, en pratique, les grilles unidimensionnelles sont préférées. La raison de cette préférence est d'ordre qualitatif : en utilisant des ficelles, on constate que les éléments non nuls de la matrice, correspondant à des déformations proches pour les régresseurs d'un même cluster, sont regroupés sur la ligne de la matrice des transitions correspondant à ce cluster. En utilisant des cartes bidimensionnelles, ce fait est plus difficilement visible. En effet, lors de la construction de la matrice des transitions, l'indice des lignes devra alors parcourir un objet en deux dimensions. Des indices de ligne successifs peuvent alors représenter des prototypes éloignés sur la grille (aux deux extrémités de deux lignes successives de la grille, par exemple), et des indices de ligne éloignés peuvent représenter des prototypes proches (deux prototypes au même niveau dans deux lignes successives de la grille). Le même raisonnement est valable pour l'indice des colonnes de la matrice. L'usage en pratique de la DQV rend donc l'utilisation des ficelles plus intéressante, bien que l'emploi de grilles bidimensionnelles soit possible.

Le SOM peut par ailleurs être avantageux par rapport à tout autre algorithme de quantification vectorielle si on envisage de regrouper les clusters obtenus après exécution de l'algorithme. En effet, seul le SOM possède la propriété de respect de la topologie, qui dit que deux vecteurs semblables sont soit dans le même cluster, soit dans des clusters voisins. Il est donc possible de regrouper des éléments de clusters voisins au sens de la grille en une sorte de "macro clusters", les vecteurs de clusters voisins étant relativement semblables. Cette approche, qui dépasse le cadre de cette thèse, revient en quelque sorte à optimiser les paramètres n_1 et n_2 de la DQV, question qui est résolue dans le cadre de ce travail par une méthode de cross-validation.

Par ailleurs, notons que l'usage de deux SOM ne peut être remplacé par un seul SOM appliqué à des vecteurs construits en concaténant simplement les régresseurs et les déformations. En effet, c'est grâce à l'utilisation de deux SOM que la matrice des transitions peut être construite. En concaténant les deux vecteurs, on perd donc la possibilité de reconstruire la matrice des transitions telle que proposée ici, toute l'information étant alors contenue dans un seul vecteur.

Enfin, mentionnons que la DQV ne fait aucune hypothèse sur le bruit contenu dans la série temporelle, à la différence de nombreux modèles de prédiction où il est fait l'hypothèse d'un bruit de distribution normale et de moyenne nulle.

3.5 Stabilité à long terme de la Double Quantification Vectorielle

Le but de cette section est de prouver théoriquement que les valeurs prédites à long terme par la méthode de Double Quantification Vectorielle ont un sens par rapport aux

valeurs passées de la série temporelle. Partant d'une série quelconque, les valeurs prédites peuvent rapidement différer des valeurs passées de la série utilisées pour l'apprentissage du modèle, surtout à long terme. En effet, chaque valeur prédite est entachée d'une erreur de prédiction. Or, les erreurs de prédiction s'accumulent à long terme, et l'impact de cette accumulation d'erreur(s) peut être très important, surtout lorsqu'on adopte une approche de prédiction récursive : on peut finalement se retrouver avec des prédictions qui sont relativement fort différentes des valeurs de la série, parfois même d'un ordre de grandeur différent. Cette propension à donner des prédictions dont la valeur explose par rapport à l'intervalle des valeurs passées de la série est appelée l'instabilité. Dans cette section, une méthode dont les prédictions sont comprises dans un intervalle borné est dite stable ; dans le cas contraire, elle est dite instable. La stabilité est une propriété attendue et nécessaire de toute méthode de prédiction, surtout dans le cadre du long terme. Sans cette stabilité, les valeurs obtenues n'apporteront malheureusement aucune information utile sur l'évolution future de la série à long terme.

Pour prouver la stabilité de la DQV à long terme, on observe l'évolution des prédictions par rapport aux valeurs initiales de la série temporelle contenue dans un intervalle noté I, les prédictions étant obtenues par approche récursive. Intuitivement, cette stabilité est attendue dans le cas de la DQV. En effet, la DQV fournit des prédictions obtenues par ajout d'une déformation à un régresseur. En toute généralité, si la déformation ajoutée au régresseur tend à donner une valeur qui sort de l'intervalle I, on peut s'attendre à ce que la déformation suivante ramène la prédiction à l'intérieur de I. En effet, les déformations sont construites sur base des régresseurs de la série, régresseurs dont les valeurs des composantes sont évidemment dans I. Ajouter une déformation à un régresseur conduit donc à un autre régresseur, dont les composantes auront des valeurs qui seront généralement dans I. Par ailleurs, considérons le cas où il existerait malgré tout une déformation $\bar{\mathbf{z}}_j$, avec j fixé, telle qu'en l'appliquant récursivement les prédictions aient tendance à sortir de l'intervalle. Rappelons que les lois conditionnelles de la matrice des transitions sont construites sur base des fréquences empiriques des transitions. Or, selon ces fréquences empiriques, la déformation $\bar{\mathbf{z}}_j$ en question est (très) peu fréquente, puisque l'intervalle I est borné. Le choix de cette déformation $\bar{\mathbf{z}}_j$ selon les différentes lois conditionnelles sera également peu fréquent, les prédictions étant ramenées et restant à l'intérieur de I tout le reste du temps. La suite de cette section est constituée d'une preuve théorique de cette intuition, due à Jean-Claude Fort, et d'une discussion sur la conclusion de cette preuve.

3.5.1 Rappels sur les chaînes de Markov

Dans cette section, quelques brefs rappels sur les chaînes de Markov sont proposés afin de définir les concepts utilisés dans la preuve de stabilité détaillée à la Section 3.5.2 suivante. Signalons que les notations habituellement utilisées dans la littérature sont reprises ici. Le lien avec les notations de la thèse est précisé au début de la preuve.

Un processus aléatoire ou stochastique est défini comme une collection de variables aléatoires $(X_0,\, X_1,\, \ldots,\, X_t,\, X_{t+1},\, \ldots)$ prenant leurs valeurs $(x_0,\, x_1,\, \ldots,\, x_t,\, x_{t+1},\, \ldots)$

dans un espace d'états Ω, supposé fini ou dénombrable. Une chaîne de Markov, ou une chaîne de Markov d'ordre 1, est un processus aléatoire qui se déplace d'un état d'un espace Ω à un autre état de Ω suivant une transition donnée par une distribution de probabilité fixée et respectant la propriété Markovienne :

$$\begin{aligned} & \mathrm{P}\left[X_{t+1} = x_{t+1} \mid X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_0 = x_0\right] \\ & \quad = \mathrm{P}\left[X_{t+1} = x_{t+1} \mid X_t = x_t\right] \\ & \quad = P(x_t, x_{t+1}), \end{aligned} \tag{3.10}$$

avec $P[. \mid .]$ dénotant la probabilité conditionnelle et $P(.,.)$ la probabilité de transition. En d'autres termes, cette propriété Markovienne énonce que le futur du processus, conditionnellement à l'état courant, est indépendant du passé. Par conséquence de cette indépendance au passé, la matrice des transitions P, de dimensions $\mid \Omega \mid \mathrm{x} \mid \Omega \mid$, est suffisante pour décrire une chaîne de Markov. Chaque ligne $P(x_t, .)$ décrit une distribution de probabilités de transition propre à l'état x_t. Une chaîne de Markov dont la matrice des transitions P ne varie pas au cours du temps est dite homogène, formellement :

$$\mathrm{P}\left[X_{t+h+1} = x_{t+1} \mid X_{t+h} = x_t\right] = \mathrm{P}\left[X_{t+1} = x_{t+1} \mid X_t = x_t\right]. \tag{3.11}$$

Dans ce cas, les probabilités de transition ne dépendent donc pas du temps, mais uniquement des états concernés.

Dans une chaîne de Markov, répéter les transitions d'un état à l'autre revient à multiplier la matrice des transitions avec elle-même. En itérant les transitions d'un état de la chaîne à l'autre, on peut, à partir d'un état x_t fixé, atteindre un certain nombre d'autres états. On dit que deux états qui peuvent être atteints en un nombre $m \geq 0$ de transitions communiquent. L'ensemble des états Ω peut être partitionné en classes d'états communiquant entre eux. Lorsqu'il n'y a qu'une seule classe, on dit que la chaîne est irréductible. En d'autres termes, la chaîne est irréductible si tout état est atteignable à partir de tout autre état par des transitions de probabilités strictement positives.

Si la limite des probabilités de transitions existe (c'est à dire si la limite de i multiplications de la matrice de transition avec elle-même existe, pour i tendant vers l'infini), alors on dit que la distribution de probabilités est une distribution stationnaire de la chaîne de Markov. Bien évidemment, une chaîne avec une distribution stationnaire est homogène dans le temps. Par ailleurs, lorsqu'une chaîne est irréductible, il existe une distribution de probabilités unique qui est également stationnaire. Enfin, une chaîne est dite ergodique lorsque la limite du nombre moyen de transitions vers un état particulier par rapport à toutes les transitions effectuées tend vers la probabilité de transition de cet état. En d'autres termes, une chaîne est ergodique si, sur un intervalle de temps assez grand, la proportion de temps passé par la chaîne dans un état particulier est proche de la probabilité que la chaîne se trouve effectivement dans cet état particulier à la fin de l'intervalle de temps considéré. Notons que l'ergodicité implique la stationnarité, une chaîne ergodique étant irréductible et apériodique.

Pour de plus amples détails sur les chaînes de Markov, le lecteur est invité à se référer aux ouvrages (13; 109; 124) consultés pour la rédaction de cette section.

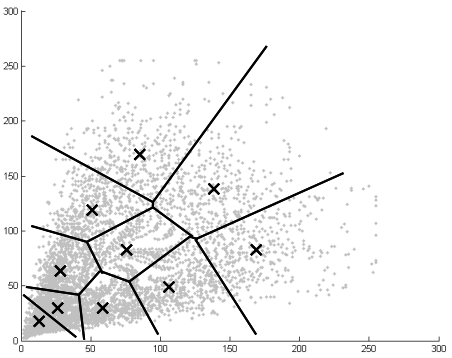

FIG. 3.1 – Distribution des régresseurs dans les différents clusters obtenus en utilisant un SOM avec dix prototypes pour la série Santa Fe A. Les régresseurs sont ici en deux dimensions. Les prototypes sont symbolisés par des croix, les clusters par des zones délimitées par des segments de droite.

3.5.2 Preuve de stabilité à long terme

La preuve consiste en trois étapes distinctes. Tout d'abord, trois lemmes techniques relatifs à la géométrie du SOM dans l'espace en deux dimensions sont démontrés. Ensuite, il est montré qu'un modèle obtenu par l'application de la DQV permet de prédire une suite de valeurs qui constituent une chaîne de Markov. Enfin, il est montré qu'une chaîne de Markov obtenue par DQV est stable à long terme ; en d'autres mots, que cette chaîne particulière possède une distribution de probabilités stationnaire. Pour ce dernier point, on montre que la chaîne de Markov est ergodique, en vérifiant les hypothèses du critère de Foster (51), ce critère constituant une condition nécessaire et suffisante d'ergodicité. Notons que la preuve de stabilité de la DQV est établie pour le cas de régresseurs en deux dimensions, soit $p = 2$.

Avant d'entrer dans le coeur de la démonstration, remarquons que les clusters c_i d'un SOM sur une distribution de données en deux dimensions, comptant au moins trois prototypes, couvrent chacun séparément moins d'un demi plan. Pour rappel, les éléments d'un cluster sont associés au prototype qui leur est le plus proche au sens de la distance utilisée, à savoir ici la distance euclidienne. Par conséquent, les points constituant les segments de frontière sont à égale distance de deux prototypes. A partir de trois prototypes, les zones couvertes par les clusters couvrent donc obligatoirement moins d'un demi-plan. La Figure 3.1 illustre ce fait sur la distribution des régresseurs en deux dimensions de la série Santa Fe A. Sur cette figure, on compte dix prototypes, représentés par des croix, et dix clusters, dont les frontières sont représentées par des segments de droite épais. On observe que la surface couverte par chaque cluster pris séparément couvre effectivement moins de la moitié du plan. Cette propriété est valable pour n'importe quel type de vecteurs en deux dimensions, donc également pour des régresseurs de données temporelles.

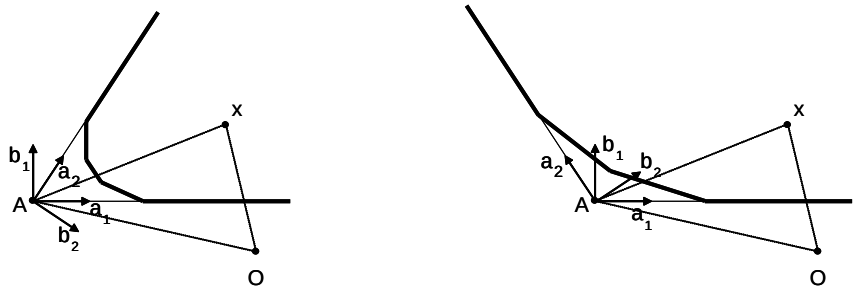

FIG. 3.2 – Illustration de la géométrie d'un cône comprenant un cluster de surface infinie. A gauche, un cône formant un angle aigu est représenté. Le cas du cône formant un angle obtus est illustré à droite. Les différentes notations sont définies dans le texte.

Sur la Figure 3.1, on distingue également deux types de clusters : les clusters dont la surface est finie et les clusters de surface infinie. Ces clusters de surface infinie peuvent également être subdivisés en deux catégories. La Figure 3.2 décrit ces deux sous-catégories, en introduisant le concept de cône dans lequel est inclus un cluster. On peut en effet avoir deux catégories de cônes : ceux formant un angle aigu et ceux formant un angle obtus.

Aussi bien pour le cas du cône avec angle aigu qu'avec angle obtus, on définit x un point de l'espace, A le sommet du cône, O l'origine de l'espace dans lequel est inclus le cône. a_1 et a_2 sont les vecteurs normés situés dans la direction des segments de droite infinis qui délimitent le cône. b_1 est défini tel que l'angle $\sphericalangle(a_1, b_1) = \frac{\pi}{2}$, et b_2 est défini tel que l'angle $\sphericalangle(b_2, a_2) = \frac{\pi}{2}$. Sur base de ces définitions, les trois lemmes suivants peuvent être démontrés.

Lemme 1.
En définissant δ_i selon :

$$\delta_i = \lim_{\|x\| \to \infty} \frac{x}{\|x\|} \cdot a_i, \tag{3.12}$$

où $\cdot$ désigne le produit scalaire, on observe que δ_1 et δ_2 sont tous deux strictement positifs dans le cas d'un angle aigu, alors qu'un seul des deux est positif lorsque l'angle est obtus.

En effet, le vecteur $\overrightarrow{Ox}$ peut s'écrire :

$$\overrightarrow{Ox} = \overrightarrow{OA} + \overrightarrow{Ax}.$$

En utilisant cette relation pour développer l'argument de la limite, relation (3.12), on a :

$$\begin{aligned}
\frac{x}{\|x\|} \cdot a_i &= \frac{\overrightarrow{Ox}}{\|x\|} \cdot a_i \\
&= \left(\frac{\overrightarrow{OA}}{\|x\|} + \frac{\overrightarrow{Ax}}{\|x\|} \right) \cdot a_i \\
&= \frac{\overrightarrow{OA} \cdot a_i}{\|x\|} + \frac{\overrightarrow{Ax} \cdot a_i}{\|x\|} \\
&= \frac{\overrightarrow{OA} \cdot a_i}{\|x\|} + \frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \frac{\|\overrightarrow{Ax}\|}{\|x\|} \cdot a_i.
\end{aligned}$$

En prenant la limite pour $\|x\| \to +\infty$, on obtient finalement :

$$\lim_{\|x\|\to+\infty} \frac{x}{\|x\|} \cdot a_i = \underbrace{\lim_{\|x\|\to+\infty} \frac{\overrightarrow{OA} \cdot a_i}{\|x\|}}_{\to 0} + \lim_{\|x\|\to+\infty} \frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \underbrace{\frac{\|\overrightarrow{Ax}\|}{\|x\|}}_{\to 1} \cdot a_i.$$

En d'autres termes, $\dfrac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \cdot a_i$ peut être borné par une constante, positive ou négative en fonction de l'angle entre les deux vecteurs dans ce produit scalaire. On obtient alors le résultat énoncé à la propriété (3.12). En observant la Figure 3.2, on constate en effet que le résultat de ce lemme est vrai à la fois pour $i = 1$ et $i = 2$ dans le cas d'un angle aigu (produit scalaire entre deux vecteurs formant un angle inférieure à 90 degrés) et qu'il est vrai au moins pour $i = 1$ ou $i = 2$, voire pour les deux, dans le cas obtus.

□

Lemme 2.

Soit C le cône dont les limites sont dans les directions données par les vecteurs a_1 et a_2. Dans le cas d'un angle aigu, tout comme dans le cas d'un angle obtus, on a :

$$\inf_{x\in C} \frac{\overrightarrow{Ax}}{\|x\|} \cdot b_i = \epsilon_i > 0. \tag{3.13}$$

En effet, le premier terme de la relation (3.13) peut être écrit selon :

$$\inf_{x\in C} \frac{\overrightarrow{Ax}}{\|x\|} \cdot b_i = \inf_{x\in C} \frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \frac{\|\overrightarrow{Ax}\|}{\|x\|} \cdot b_i.$$

Puisque $\dfrac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \cdot b_i > 0$ par construction des vecteurs b_i, on obtient donc :

$$\inf_{x\in C} \underbrace{\frac{\|\overrightarrow{Ax}\|}{\|x\|}}_{>0} \frac{\overrightarrow{Ax}}{\|\overrightarrow{Ax}\|} \cdot b_i > 0.$$

Par la définition des vecteurs b_i, le produit scalaire ci-dessus est donc toujours strictement positif. Le résultat de ce lemme est donc vrai pour $i = 1$ et $i = 2$ aussi bien dans le cas des cônes avec angle aigu que dans le cas des cônes avec angle obtus.

□

Lemme 3.

Supposons que :

$$\mathrm{E}_\lambda(\bar{\mathbf{z}}) \cdot a_1 < 0 \text{ et } \mathrm{E}_\lambda(\bar{\mathbf{z}}) \cdot a_2 < 0, \tag{3.14}$$

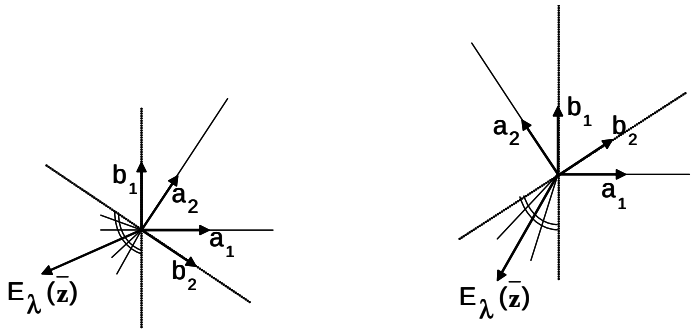

FIG. 3.3 – Lemme 3., clusters compris dans un cône formant un angle aigu, à gauche, et un angle obtus, à droite.

où $\mathrm{E}_\lambda(.)$ est l'espérance suivant la distribution empirique λ correspondant à la ligne k de la matrice des transitions, ligne correspondant à un cluster de régresseurs c_k non borné pour k fixé ; et où $\bar{\mathbf{z}}$ sont les prototypes des clusters de déformation. On pose $\gamma_i > 0$ définis selon :

$$\mathrm{E}_\lambda(\bar{\mathbf{z}}) \cdot a_i = -\gamma_i < 0. \tag{3.15}$$

Comme on peut le voir à la Figure 3.3, on a :

$$\mathrm{E}_\lambda(\bar{\mathbf{z}}) \cdot b_i < 0, \tag{3.16}$$

pour au moins $i = 1$ ou $i = 2$ dans le cas d'un angle aigu et pour $i = 1$ et $i = 2$ dans le cas d'un angle obtus. Ces deux cas de figures sont illustrés respectivement dans la partie gauche et la partie droite de la Figure 3.3.

Il faut malgré tout souligner que, stricto sensu, (3.14) est une hypothèse, ce résultat n'ayant pas été démontré formellement. Néanmoins, dans tous les cas rencontrés, il a été possible de valider expérimentalement cette hypothèse (3.14), comme illustré par exemple à la Figure 3.4 dans le cas des régresseurs et des déformations obtenus à partir de la série Santa Fe A. En effet, on y observe que la moyenne des déformations d'un cluster pointe vers l'intérieur de la distribution des régresseurs. Notons que cette Figure 3.4 n'est autre que la Figure 3.1 sur laquelle ont été ajoutés, pour chaque cluster non borné, le vecteur représentant la moyenne des déformations associées aux régresseurs du cluster considéré, cluster par cluster.

□

Après la démonstration de ces trois lemmes, prouvons maintenant que la série des valeurs prédites est une chaîne de Markov. La démonstration ayant lieu dans le cadre de la Double Quantification Vectorielle, les notations "Markoviennes" sur des variables aléatoires X_t sont remplacées par les notations des régresseurs $\mathbf{x}_t$. Pour rappel, les autres notations utilisées dans le cadre de la Double Quantification Vectorielle sont la déformation $\mathbf{z}_t$, les prototypes des clusters de régresseurs c_i et de déformations c'_j notés respectivement

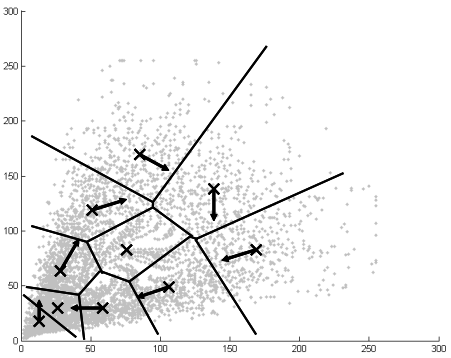

FIG. 3.4 – Distribution des régresseurs de la série Santa Fe A, en deux dimensions, dans les différents clusters obtenus en utilisant un SOM avec dix prototypes. La moyenne des déformations associées aux régresseurs de chaque cluster est symbolisée par une flèche partant du prototype du cluster considéré, représenté par une croix.

$\bar{\mathbf{x}}_i$ et $\bar{\mathbf{z}}_j$. Le symbole $\hat{}$ désigne ici également l'estimation d'une valeur inconnue, par exemple d'une prédiction.

La suite des prédictions est une chaîne de Markov

Soit $x(N)$ la dernière valeur de la série temporelle. Le régresseur correspondant est $\mathbf{x}_N$, point de départ de la simulation jusqu'à l'horizon h. Le cluster auquel appartient $\mathbf{x}_N$ est c_k dont le prototype $\bar{\mathbf{x}}_k$ est le plus proche du régresseur considéré $\mathbf{x}_N$. Par abus de langage, les notations $\bar{\mathbf{x}}_k$ et c_k sont remplacées respectivement par $\bar{\mathbf{x}}_N$ et c_N pour indiquer explicitement quels sont les éléments sélectionnés au moment considéré. La déformation qui est alors appliquée à $\mathbf{x}_N$ est $\bar{\mathbf{z}}_N$, le prototype du cluster de déformations c'_N. Les prédictions des valeurs suivantes de la série sont donc obtenues sur base de la relation (3.9), en toute généralité :

$$\begin{aligned}
\hat{\mathbf{x}}_{N+1} &= \mathbf{x}_N + \bar{\mathbf{z}}_N, \\
\hat{\mathbf{x}}_{N+2} &= \hat{\mathbf{x}}_{N+1} + \bar{\mathbf{z}}_{N+1} \\
&= \mathbf{x}_N + \bar{\mathbf{z}}_N + \bar{\mathbf{z}}_{N+1}, \\
\hat{\mathbf{x}}_{N+3} &= \hat{\mathbf{x}}_{N+2} + \bar{\mathbf{z}}_{N+2} \\
&= \mathbf{x}_N + \bar{\mathbf{z}}_N + \bar{\mathbf{z}}_{N+1} + \bar{\mathbf{z}}_{N+2}, \\
&\vdots
\end{aligned}$$

où la suite des prototypes de déformations $\bar{\mathbf{z}}_N$, $\bar{\mathbf{z}}_{N+1}$, $\bar{\mathbf{z}}_{N+2}$, ... est sélectionnée selon les lois des lignes de la matrice des transitions pour les clusters c_N, c_{N+1}, c_{N+2}, ..., aux différents instants $t = N, N+1, N+2, \ldots$. La suite des prédictions $\hat{\mathbf{x}}_{N+1}$, $\hat{\mathbf{x}}_{N+2}$, $\hat{\mathbf{x}}_{N+3}$, ... obtenue est une chaîne de Markov. Cette chaîne est homogène dans le temps, les différentes distributions de probabilité de transition décrites dans la matrice des transitions n'étant pas dépendantes du temps. Par ailleurs, cette chaîne est irréductible, tous les prototype(s) de déformations $\bar{\mathbf{z}}_N$ pouvant être atteint à partir de n'importe quel prototype

de déformations au moyen d'une (ou plusieurs) transition(s). Enfin, cette chaîne de Markov est également définie sur un ensemble dénombrable, ce dernier point découlant du fait que, la série temporelle étant limitée dans le temps, les régresseurs $\mathbf{x}_t$ sont en nombre fini. Par conséquent, les déformations $\mathbf{z}_t$, et a fortiori leurs prototypes $\bar{\mathbf{z}}_j$, sont également en nombre fini.

□

La suite des prédictions étant une chaîne de Markov, il reste à démontrer que cette chaîne est stable. Pour ce faire, on utilise le critère de Foster, qui donne les conditions nécessaires et suffisantes pour qu'une chaîne soit ergodique ; l'ergodicité d'une chaîne impliquant l'existence d'une distribution de probabilité stationnaire unique prouvant donc la stabilité de la chaîne. Cette vérification des conditions du critère de Foster est établie ici en deux dimensions ; l'avantage du cas en deux dimensions est de pouvoir utiliser le support visuel des figures des lemmes 1., 2. et 3.

Critère de Foster et stabilité de la chaîne

Le critère de Foster (51) s'énonce comme suit :
Une chaîne irréductible et dénombrable est une chaîne ergodique si et seulement si il existe $g(.)$*, une fonction positive,* $\varepsilon \geq 0$ *et* Ω*, un ensemble fini, tels que* :

$$\forall x \in \Omega : \mathrm{E}\left(g(\hat{\mathbf{x}}_{\mathrm{N}+1}) \mid \mathbf{x}_{\mathrm{N}} = x\right) < \infty, \tag{3.17}$$

$$\forall x \notin \Omega : \mathrm{E}\left(g(\hat{\mathbf{x}}_{\mathrm{N}+1}) \mid \mathbf{x}_{\mathrm{N}} = x\right) - g(x) \leq -\varepsilon. \tag{3.18}$$

Tout comme ci-dessus, les notations sont précisées pour le cas de la Double Quantification Vectorielle. Pour cette démonstration, les notations pour les variables aléatoires X_t sont donc remplacées par les notations pour les régresseurs $\mathbf{x}_t$.

Dans le cadre de la stabilité à long terme, l'interprétation de ce théorème est la suivante. On considère l'ensemble Ω comme étant l'intervalle des valeurs passées de la série temporelle. Tant que le régresseur à l'instant N se trouve dans l'ensemble Ω, soit le cas (3.17), la valeur attendue pour la prédiction en $N+1$ est finie. Lorsque le régresseur quitte cet intervalle de valeurs, soit le cas (3.18), la différence entre la valeur attendue en $N+1$ et la valeur réelle à l'instant N est négative. Or, cette différence correspond à une déformation. En d'autres termes, si la valeur attendue pour la prédiction tend à quitter l'intervalle des valeurs passées de la série, la déformation sélectionnée agit en sens inverse, et ramène donc la valeur attendue en prédiction à l'intérieur de Ω. Les valeurs prédites n'explosent donc jamais : même si on a tendance à sortir de l'intervalle des valeurs Ω, les déformations ramènent les prédictions à l'intérieur de Ω, de sorte qu'on reste toujours dans cet intervalle, même à long terme.

Par la suite, on considère que la fonction $g(.)$ utilisée dans les relations (3.17) et (3.18) est $g(.) = \|.\|^2$. De plus, on suppose que les SOM utilisés dans la DQV comptent plus de trois prototypes, de sorte que la zone de l'espace couverte par chaque cluster de ces SOM couvre moins d'un demi plan. Enfin, étant donné que la chaîne de Markov est homogène, il est suffisant d'étudier le comportement lors de la transition $\bar{\mathbf{z}}_N$ entre $\mathbf{x}_N$ et $\hat{\mathbf{x}}_{N+1}$. En

effet, le même raisonnement peut être appliqué pour tout autre instant dans le temps. La preuve du critère de Foster va être effectuée en deux étapes, pour chacune des deux relations (3.17) et (3.18).

Considérons le cas où $\|\mathbf{x}_N\| < R_N$, où R_N peut être n'importe quelle constante. Dans ce cas, par inégalité triangulaire, on peut écrire :

$$\begin{aligned} \mathrm{E}(\|\hat{\mathbf{x}}_{N+1}\|) &< R_N + \|\bar{\mathbf{z}}_N\| \\ &\leq R_N + \max_j(\|\bar{\mathbf{z}}_j\|). \end{aligned} \tag{3.19}$$

Comme les déformations $\bar{\mathbf{z}}_j$ sont en nombre fini, le maximum de leur norme est également fini ; ce qui prouve la première relation (3.17) dans le cas où la norme de $\mathbf{x}_N$ est finie. Ce cas de la norme de $\mathbf{x}_N$ finie correspond au cas des clusters bornés, tels qu'on en voit deux sur la Figure 3.1.

Le second cas concerne donc les éléments $\|\mathbf{x}_N\| \to +\infty$. Ce cas ne peut arriver que pour des clusters non bornés, qu'ils soient contenus dans un cône d'angle aigu ou obtus. Le reste de la preuve est dédiée spécifiquement au cas des clusters non bornés.

Soit c_N le cluster dont le prototype $\bar{\mathbf{x}}_N$ est le plus proche de $\mathbf{x}_N$. Considérons maintenant la distribution des transitions pour la ligne de la matrice des transitions correspondant à ce prototype $\bar{\mathbf{x}}_N$. On a :

$$\begin{aligned} \mathrm{E}(g(\hat{\mathbf{x}}_{N+1}) \mid \mathbf{x}_N = x) - g(x) &= \mathrm{E}(g(\mathbf{x}_N + \bar{\mathbf{z}}_N) \mid \mathbf{x}_N = x) - g(x) \\ &= \mathrm{E}(\|\mathbf{x}_N + \bar{\mathbf{z}}_N\|^2 \mid \mathbf{x}_N = x) - \|x\|^2 \\ &= \mathrm{E}_{\lambda_N}(\|\mathbf{x}_N + \bar{\mathbf{z}}_N\|^2) - \|\mathbf{x}_N\|^2 \\ &= \mathrm{E}_{\lambda_N}(\|\mathbf{x}_N\|^2) + 2\mathrm{E}_{\lambda_N}(\mathbf{x}_N \cdot \bar{\mathbf{z}}_N) + \mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2) - \|\mathbf{x}_N\|^2 \\ &= \|\mathbf{x}_N\|^2 + 2\mathrm{E}_{\lambda_N}(\mathbf{x}_N \cdot \bar{\mathbf{z}}_N) + \mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2) - \|\mathbf{x}_N\|^2 \\ &= 2\mathbf{x}_N \cdot \mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) + \mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2), \end{aligned}$$

où λ_N est la distribution empirique, dans la matrice des transitions, correspondant au cluster non borné c_N contenant le régresseur $\mathbf{x}_N$ et $g(.) = \|.\|^2$. Pour résumer, on obtient donc :

$$\mathrm{E}(g(\hat{\mathbf{x}}_{N+1}) \mid \mathbf{x}_N = x) - g(x) = 2\mathbf{x}_N \cdot \mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) + \mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2),$$

qui peut se réécrire comme suit, en utilisant le fait que $\mathbf{x}_N = x$ et en multipliant les deux termes du membre de droite par $\frac{\|x\|}{\|x\|} = 1$:

$$\mathrm{E}(g(\hat{\mathbf{x}}_{N+1}) \mid \mathbf{x}_N = x) - g(x) = 2\|x\| \left[\frac{x \cdot \mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)}{\|x\|} + \frac{\mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2)}{2\|x\|} \right]. \tag{3.20}$$

Observons maintenant les deux termes entre crochets de cette dernière relation (3.20). Pour le second terme, sachant que $\|\bar{\mathbf{z}}_N\|^2$ est fini, on a :

$$\sup_{x \in c_N} \mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2) < M_N < \infty.$$

Par conséquent, pour un $\alpha_N > 0$ et pour $\|x\| > \dfrac{M_N}{\alpha_N}$, on a :

$$\frac{1}{\|x\|}\mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2) < \alpha_N. \tag{3.21}$$

Concernant le premier terme de (3.20), par les lemmes 1. et 3., et sous l'hypothèse (3.14), il est possible de sélectionner soit $i = 1$ ou $i = 2$ tels que :

$$\begin{cases} \lim\limits_{\|x\|\to+\infty} \dfrac{x}{\|x\|} \cdot a_i = \delta_i > 0, \\ \\ \mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot b_i < 0. \end{cases} \tag{3.22}$$

Supposons par exemple que $i = 2$ satisfasse ces deux conditions (3.22) ; le raisonnement étant le même pour le cas $i = 1$ si ces conditions ne sont pas vérifiées pour $i = 2$. En supposant, donc, que $i = 2$, on décompose le terme $\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)$ de l'équation (3.20) dans une base décrite par les vecteurs orthogonaux (b_2, a_2) selon :

$$\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) = (\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot a_2)a_2 + (\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot b_2)b_2, \tag{3.23}$$

et donc :

$$\frac{x}{\|x\|} \cdot \mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) = (\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot a_2)\left(\frac{x}{\|x\|} \cdot a_2\right) + (\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot b_2)\left(\frac{x}{\|x\|} \cdot b_2\right). \tag{3.24}$$

Détaillons à présent les différents termes composants le membre de droite de (3.24).

Par le lemme 1., on sait que $\lim\limits_{\|x\|\to+\infty} \dfrac{x}{\|x\|} \cdot a_2 = \delta_2 > 0$. Donc, on peut écrire :

$$\frac{x}{\|x\|} \cdot a_2 > \frac{\delta_2}{2}, \tag{3.25}$$

pour $\|x\|$ suffisamment grand.

Par ailleurs, selon le lemme 3., $\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot a_2 = -\gamma_2 < 0$. De plus, suivant les conditions (3.22) ci-dessus, $\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) \cdot b_2 < 0$. Par conséquent, on peut écrire :

$$\frac{x}{\|x\|} \cdot b_2 = \left(\frac{\overrightarrow{OA}}{\|x\|} + \frac{\overrightarrow{Ax}}{\|x\|}\right) \cdot b_2. \tag{3.26}$$

Pour x suffisamment grand appartenant à un cluster non borné, c'est à dire pour $\|x\| \to +\infty$, on a $\dfrac{\overrightarrow{OA}}{\|x\|} \cdot b_2 \to 0$. De plus, par le lemme 2., on a $\dfrac{\overrightarrow{Ax}}{\|x\|} \cdot b_2 \geq \epsilon_2 > 0$. On réécrit donc la relation (3.26) en la simplifiant suivant :

$$\frac{x}{\|x\|} \cdot b_2 \geq \frac{\epsilon_2}{2} > 0, \tag{3.27}$$

lorsque $\|x\|$ est suffisamment grand.

En développant la relation (3.24), on obtient donc, après simplification :

$$\begin{aligned}\frac{x}{\|x\|}\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) &\leq \underbrace{(\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)\cdot a_2)}_{=-\gamma_2 \text{ par (3.15)}}\underbrace{\left(\frac{x}{\|x\|}\cdot a_2\right)}_{>\frac{\delta_2}{2} \text{ par (3.25)}} + \underbrace{(\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)\cdot b_2)}_{<0 \text{ par (3.22)}}\underbrace{\left(\frac{x}{\|x\|}\cdot b_2\right)}_{\geq\frac{\epsilon_2}{2} \text{ par (3.27)}} \\ &< -\gamma_2\frac{\delta_2}{2}, \end{aligned} \tag{3.28}$$

résultat obtenu pour $\|x\|$ suffisamment grand, en d'autres mots pour $\|x\|$ plus grand que n'importe quelle constante K_N^2 ce qui est noté $\|x\| > K_N^2$.

Tout le développement ci-dessus ayant conduit à la relation finale (3.28) a été réalisé sous l'hypothèse que $i = 2$. Si le cas $i = 2$ ne permettait pas de vérifier les conditions (3.22), alors il suffit de prendre $i = 1$, et d'appliquer le même résultat pour arriver à la relation finale :

$$\frac{x}{\|x\|}\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N) < -\gamma_1\frac{\delta_1}{2}. \tag{3.29}$$

Ici aussi, ce résultat est valide lorsque $\|x\|$ est suffisamment grand, soit pour $\|x\| > K_N^1$.

On peut maintenant simplifier l'expression (3.20), en utilisant les résultats obtenus pour les deux termes, soit les relations (3.21) d'une part et (3.28) ou (3.29) d'autre part :

$$\begin{aligned}\mathrm{E}(g(\hat{\mathbf{x}}_{N+1}) \mid \mathbf{x}_N = x) - g(x) &= 2\|x\|\left[\frac{x\cdot\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)}{\|x\|} + \frac{\mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2)}{2\|x\|}\right] \\ &< 2\|x\|\left[-\alpha_N + \frac{1}{2}\alpha_N\right] \\ &= -2\|x\|\frac{\alpha_N}{2} \end{aligned} \tag{3.30}$$

où $\|x\|$ est suffisamment grand, ce qui est noté $\|x\| > K_N$ avec $K_N = \max(K_N^1, K_N^2)$, et où la constante α_N dans la relation (3.21) est choisie telle que $\alpha_N = \min(\frac{\gamma_1\delta_1}{2}, \frac{\gamma_2\delta_2}{2})$.

Tout le développement effectué pour obtenir le résultat (3.30) a été réalisé pour un seul cluster, le cluster c_N auquel $\mathbf{x}_N$ appartient. Par conséquent, les constantes α_N et K_N choisies dans le développement pour obtenir ce résultat sont dépendantes de ce cluster c_N. En toute généralité, il faut considérer non pas un seul cluster non borné, mais tous les clusters c_i du SOM qui sont non bornés. On définit donc des constantes valables pour tous les clusters c_i non bornés telles que $\alpha = \inf_{c_i}\alpha_i$ et $K = \sup_{c_i} K_i$ avec $i \in \mathrm{I}$ l'ensemble des indices des clusters non bornés d'une carte auto-organisée. Sur base de ces constantes choisies en tenant compte de tous les clusters non bornés, on a finalement :

$$\forall\|x\| \geq K \;:\; \frac{x\cdot\mathrm{E}_{\lambda_N}(\bar{\mathbf{z}}_N)}{\|x\|} + \frac{\mathrm{E}_{\lambda_N}(\|\bar{\mathbf{z}}_N\|^2)}{2\|x\|} < -\frac{\alpha}{2} < 0. \tag{3.31}$$

En substituant ce dernier résultat (3.31) dans la relation (3.20), on obtient finalement :

$$\mathrm{E}(g(\hat{\mathbf{x}}_{N+1}) \mid \mathbf{x}_N = x) - g(x) < -\alpha\|x\|, \tag{3.32}$$

où le membre de droite de tends vers $-\infty < -\varepsilon$ pour $\|x\| \to +\infty$.

Pour pouvoir conclure, il reste à définir l'ensemble Ω qui doit être utilisé, afin de prouver les relations (3.17) et (3.18) suivant les développements proposés ici. On pose :

$$\Omega = (\bigcup_{i \in \mathrm{I}} c_i) \bigcup \{\mathbf{x}_N |\ \|\mathbf{x}_N\| < K\}, \tag{3.33}$$

où, pour rappel, I est l'ensemble des indices des clusters non bornés d'une carte auto-organisée. Dès lors, en utilisant la définition de l'ensemble Ω (3.33), tous les développements ci-dessus permettent de prouver les deux relations (3.17) et (3.18) du critère de Foster. Ce critère étant une condition nécessaire et suffisante, la chaîne de Markov qui vérifie ce critère est donc ergodique. La conséquence de ce résultat est que la chaîne des prédictions, ergodique, admet une loi de distribution stationnaire unique, autrement dit elle est stable.

□

3.5.3 Portée du résultat de stabilité

Le résultat obtenu dans la Section 3.5.2 précédente permet de valider théoriquement la procédure de simulation de la DQV utilisée afin de prédire à long terme. En effet, grâce à ce résultat théorique, on a la garantie que les prédictions fournies par la DQV n'auront jamais tendance à tendre vers l'infini. Au contraire, les valeurs prédites par simulation resteront à l'intérieur de l'intervalle qui contient l'ensemble des valeurs de la série temporelle, et elles y seront ramenées si jamais elles avaient tendance à sortir de cet intervalle. Par ailleurs, les simulations à long terme étant répétées, il est possible d'obtenir des informations qualitatives sur l'évolution de la série temporelle, telles que des statistiques sur le comportement de la série à long terme.

Le résultat obtenu dans la Section 3.5.2 précédente est également utile par rapport aux différentes approches de prédiction à long terme décrites dans la Section 2.3. La preuve de la stabilité a été détaillée dans le cas de régresseurs en deux dimensions, pour une transition. Le raisonnement est exactement le même pour n'importe quelle transition ultérieure, il suffit de modifier les indices N et $N+1$ utilisés dans les différentes étapes de la démonstration. Par ailleurs, il est possible de remplacer les indices utilisés dans le cas de la prédiction à un pas par des indices pour une prédiction à un horizon quelconque $h > 1$. Il est alors possible d'établir aisément un résultat semblable dans le cas de la prédiction directe à long terme, en appliquant exactement le même raisonnement.

Concernant la généralisation de la preuve à des vecteurs de dimension $p > 2$, le résultat n'est pas établi. Il est possible de démontrer que la suite des prédictions est une chaîne de Markov, pour des régresseurs de dimension quelconque, aussi bien dans le cas de la prédiction scalaire que pour la prédiction de vecteurs. La démonstration du critère de Foster pourrait sans doute être réalisée aussi pour des régresseurs de dimension quelconque, aussi bien dans le scalaire que vectoriel, mais à condition que les résultats obtenus dans les trois lemmes techniques sur la géométrie du SOM soient généralisables à n'importe quelle dimension. Or, les résultats de ces lemmes ne sont pas directement généralisables pour des dimensions $p > 2$. Observons par exemple ce qui se passe en trois dimensions.

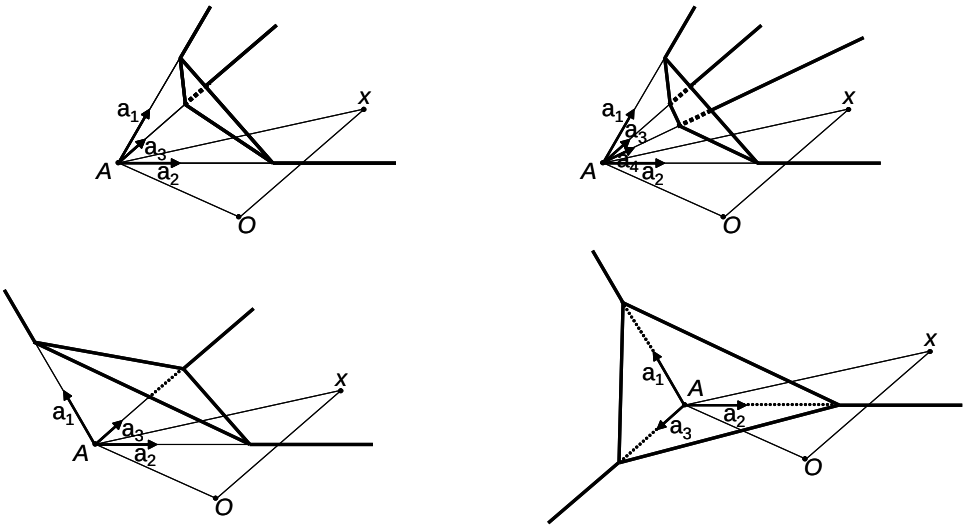

FIG. 3.5 – Cônes en trois dimensions.

En utilisant au moins trois prototypes, il est possible de définir des clusters en trois dimensions qui couvrent moins d'un demi espace. Dans ce cas, la zone couverte par les clusters en trois dimensions peut de nouveau être inscrite dans des cônes tridimensionnels de sommet A. Toutefois, il n'est plus possible de classer ces cônes tridimensionnels entre ceux qui forment un angle aigu ou un angle obtus : il existe en fait une grande variété de cônes différents, en fonction des intersections des plans définissant les limites des clusters et du nombre de ces plans. La Figure 3.5 illustre le cas de quelques uns de ces cônes tridimensionnels. Les vecteurs a_i définissant le sommet A sont situés sur les droites résultant des intersections de plans.

Sur base des graphes de la Figure 3.5, on peut constater que le Lemme 1. reste vrai en trois dimensions également. Par contre, il n'est pas possible d'établir les Lemmes 2. et 3., puisqu'il existe une infinités de vecteurs b perpendiculaires à chaque vecteur a_i. En effet, pour chaque vecteur a_i, il existe un faisceau de plans passant par la droite supportant ce vecteur. Autrement dit, il existe une infinité de plans passant par a_i. Or, pour chaque plan de ce faisceau, on peut définir un vecteur b perpendiculaire à a_i dans ce plan. Le problème est que, suivant le plan considéré et le vecteur b obtenu, il est parfois possible de vérifier les Lemmes 2. et 3., parfois pas.

Pour tenter de résoudre ce problème de choix, on pourrait projeter tout point x dans un plan formé par deux vecteurs a_i. On se ramène alors au cas en deux dimensions. On peut alors répéter la projection pour tous les plans définissant les limites du cône tridimensionnel. Malheureusement, avec cette approche, on arrive à prouver les trois lemmes pour n'importe quelle projection d'un point x de l'espace en trois dimensions, mais pas pour le point x lui-même. Le résultat ne peut donc être obtenu en étendant simplement le raisonnement effectué en deux dimensions. Pour établir le critère de Foster en dimen-

sion trois et éventuellement plus, il faudrait pouvoir démontrer des résultats géométriques équivalents aux Lemmes 1. à 3. dans l'espace de dimension considérée.

Enfin, dans la section 3.3.3, il a été expliqué comment la méthode DQV peut être utilisée en prédiction à court terme et à long terme, tant en scalaire qu'en vectoriel. Sachant maintenant que les prédictions à long terme en récursif et en direct sont stables, on peut se poser la question de savoir s'il existe une approche supérieure à une autre. Une réponse à cette question permettrait de savoir s'il vaut mieux utiliser du récursif ou du direct en prédiction à long terme. Dans le cadre de l'application de la DQV à la série CATS, résumée à la section 3.6.3, une comparaison empirique des prédictions récursive et directe a été réalisée. Une discussion théorique générale, indépendante du modèle de DQV, est proposée à ce sujet en Section 3.7.

3.6 Applications

Dans cette section, la Double Quantification Vectorielle est utilisée afin de prédire plusieurs séries temporelles. Tant dans le cas scalaire que dans le cas vectoriel, l'accent est mis sur la prédiction à long terme, mais la prédiction à court terme est également illustrée. La première application de la DQV illustre la prédiction d'une série scalaire au moyen d'un benchmark classique en prédiction de séries, la série Santa Fe A (177). La deuxième application illustre le cas de la prédiction vectorielle dans le cadre d'une application concrète, à savoir un problème de consommation électrique. La troisième et dernière application a pour but d'illustrer les performances de la méthode en comparaison avec d'autres méthodes de prédiction de séries temporelles, sur base de résultats obtenus par la DQV lors de la compétition de prédiction CATS (104).

3.6.1 Santa Fe A

Pour cette première application, la série Santa Fe A (177) est de nouveau utilisée. Cette série illustre l'application de la DQV dans le cas d'une série scalaire pour de la prédiction à un pas, soit $h = 1$, répétée pour du long terme suivant une approche itérative. Pour cette application, la sélection des paramètres de structure n_1 et n_2 est faite par cross-validation, sur base du critère de moyenne des moindres carrés MSE (2.13).

L'intérêt principal de la méthode DQV n'est pas au niveau de la prédiction de valeurs numériques, mais au niveau de la prédiction de valeurs qualitatives sur les tendances à long terme. Néanmoins, dans un premier temps, la méthode peut être utilisée pour effectuer de la prédiction à un pas. En utilisant le jeu des 1 000 données de la compétition Santa Fe A, soit les conditions réelles de la compétition, la MSE réalisée par la méthode DQV sur les 100 données suivantes lui aurait permis de se classer douzième.

Dans un deuxième temps, disposant au total de 10 000 valeurs de la série, on peut utiliser un plus grand nombre de ces données pour approximer de manière plus précise les différentes lois de probabilités de la matrice des transitions. Dans ce but, on choisit

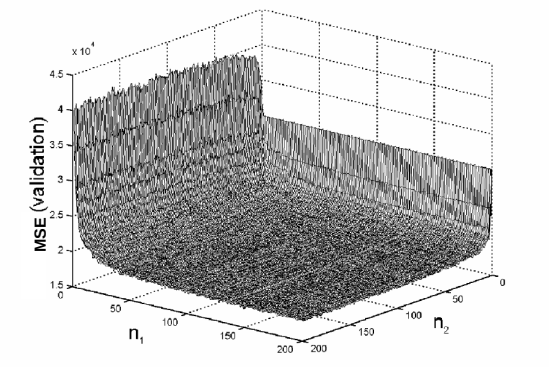

FIG. 3.6 – Graphe de l'estimation de l'erreur de généralisation par cross-validation pour les 40 000 structures de modèles testées, série Santa Fe A.

arbitrairement d'utiliser les 6 000 premières données de la série comme ensemble d'apprentissage ; les 2 000 suivantes servant d'ensemble de validation. Un ensemble de test comportant les 100 valeurs suivantes a ensuite été utilisé. Après sélection de la structure optimale sur base de l'ensemble de validation, les 8 000 premières données ont été réunies dans un nouvel ensemble utilisé pour l'apprentissage d'un modèle avec les valeurs n_1 et n_2 optimales. C'est ce dernier modèle, appris sur un ensemble de données 8 000 données de la série, qui a été utilisé lors de la phase de test. Pour cette application, on construit les régresseurs selon (150) :

$$\mathbf{x}_t = (x(t), x(t-1), x(t-2), x(t-3), x(t-5), x(t-6)) \,. \tag{3.34}$$

Pour résumer, on applique donc la DQV a un ensemble de données, avec $d = 1$, $p = 6$ et $h = 100$. Comme $d = 1$, la version scalaire décrite dans la Section 3.3 est appliquée. Notons que, sur base de résultats antérieurs, la composante $x(t-4)$ ne se trouve pas dans le régresseur (150).

Des cartes auto-organisées avec des grilles en ficelle ont été utilisées, chacune des deux grilles comptant respectivement de 1 à 200 prototypes. Toutes les combinaisons possibles pour les paramètres n_1 et n_2 ont été testées, soit 40 000 structures de modèle différentes. L'erreur de généralisation $\hat{E}_{\text{cross-validation}}(s)$ est ici la moyenne de $R = 100$ répétitions de la procédure apprentissage/validation. Cette erreur $\hat{E}_{\text{cross-validation}}(s)$ des 40 000 modèles de structure s est illustrée à la Figure 3.6. Notons que, d'une part, le graphe de l'erreur de généralisation paraît particulièrement lisse et que, d'autre part, ce graphe comporte une large zone relativement plate. En fait, même dans cette zone, il y a de nombreuses différences locales. La meilleure structure de modèles qui est sélectionnée est donc celle qui réalise numériquement le minimum de $\hat{E}_{\text{cross-validation}}(s)$. Cette structure choisie parmi tous les modèles testés compte 179 prototypes pour la première grille SOM et 161 pour la seconde.

Comme indiqué ci-dessus, un modèle avec $n_1 = 179$ et $n_2 = 161$ prototypes est utilisé sur l'ensemble d'apprentissage des 8 000 premières données de la série. Après cet ultime

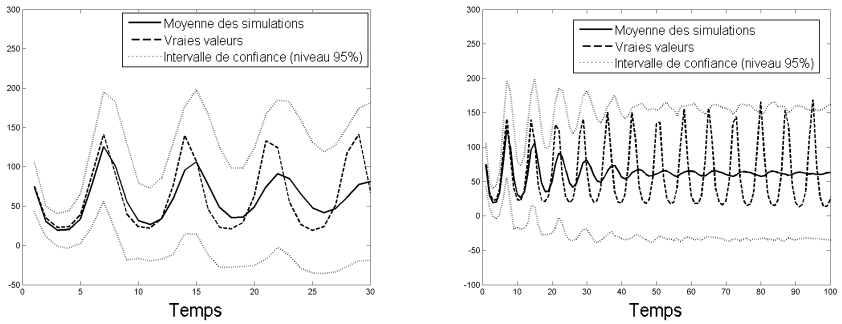

FIG. 3.7 – Comparaison entre la moyenne des 1 000 simulations (trait plein) et les vraies valeurs de la série Santa Fe A (tirets). Les lignes extérieures correspondent à l'intervalle de confiance au niveau 95%. Les graphes de gauche et de droite présentent les résultats pour les horizons $h = 30$ et $h = 100$, respectivement.

apprentissage, 1 000 simulations de la procédure de prédiction à un pas sont effectuées, jusqu'à l'horizon 100. Les moyennes de ces 1 000 simulations pour les cent différents pas de temps sont les prédictions fournies par la méthode DQV. La Figure 3.7 présente les valeurs prédites jusqu'aux horizons $h = 30$ et $h = 100$, respectivement à gauche et à droite. Sur cette figure, on observe que la méthode DQV est en mesure de prédire avec une certaine fiabilité numérique une vingtaine de valeurs de la série temporelle.

Au delà des prédictions scalaires, ce sont les informations fournies par la méthode sur le long terme qui constituent l'intérêt principal de la méthode. Grâce à la répétition de la procédure de simulations, la DQV permet de calculer des statistiques sur l'évolution future de la série temporelle. Ces statistiques sont visibles à la Figure 3.7 où on voit, premièrement, que la moyenne de la série reste stable sur l'intervalle de temps considéré et que, deuxièmement, les valeurs de la série restent dans le même intervalle de valeurs, ce qui est indiqué par l'espace compris entre les limites de l'intervalle de confiance. Et effectivement, malgré les variations plus ou moins cycliques de la série Santa Fe A, cette série reste stable sur les 100 valeurs suivantes et la majorité de ces 100 valeurs sont effectivement comprises dans l'intervalle de confiance.

La Figure 3.8 présente une centaine de simulations parmi les 1 000, sélectionnées au hasard. On peut y voir, d'une courbe à l'autre, l'effet du tirage des déformations suivant les lois reprises dans la matrice des transitions, les courbes des différentes simulations se croisant de plus en plus au fur et à mesure que le temps avance. Néanmoins, on observe également la grande stabilité des prédictions pour les dix premières valeurs, malgré le caractère stochastique du modèle.

A titre d'exemple, la Table 3.1 donne les lois conditionnelles obtenues pour un modèle beaucoup plus simple qui ne comporte que $n_1 = 6$ et $n_2 = 8$ prototypes. Cette table est proposée afin d'illustrer le fait que la matrice des transitions est effectivement creuse : comme attendu, et discuté dans la section 3.3.1, on constate la présence d'un certain

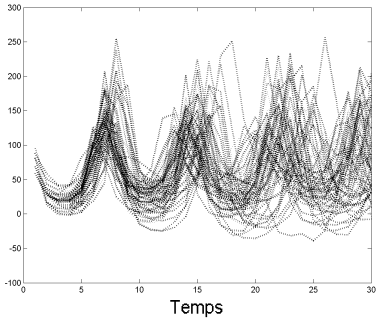

FIG. 3.8 – 100 simulations de la série Santa Fe A, pour l'horizon $h = 30$. Chaque courbe représente une simulation obtenue en utilisant le modèle stochastique comptant $n_1 = 179$ et $n_2 = 161$ prototypes.

0.12	0	0	0	0	0	0.23	0.65
0.67	0.30	0	0	0	0	0.02	0.01
0.05	0.55	0.40	0	0	0	0	0
0.03	0	0.30	0.54	0.13	0	0	0
0	0	0	0	0.50	0.48	0.02	0
0.06	0	0	0	0.0	0.34	0.56	0.04

TAB. 3.1 – Exemple de matrice des transitions, pour un modèle simple comptant $n_1 = 6$ et $n_2 = 8$ prototypes.

nombre de zéros dans la matrice des transitions. On peut remarquer que les éléments non nuls de la matrice des transitions sont généralement groupés. Ce fait est probablement une conséquence de l'utilisation d'une ficelle comme structure de grille du SOM. Cette caractéristique de la matrice des transitions pourrait être utilisée pour regrouper ensemble des clusters et jouer de la sorte sur le nombre de clusters, tant au niveau des régresseurs que des déformations, mais cette possibilité n'est pas explorée ici. Enfin, notons que, dans cette table, la somme des éléments de chaque ligne vaut bien 1.

3.6.2 Série polonaise électrique

Les résultats proposés dans cette section illustrent l'utilisation de la DQV dans le cas vectoriel, sur des vecteurs de données temporelles (3.7). La série utilisée est la série de la consommation électrique en Pologne (126). Cette série contenant les valeurs de la consommation, heure par heure, est représentée graphiquement à la Figure 3.9 et décrite en Annexe A. Cette série a été utilisée dans un cadre de prédictions à court terme (126), (103) ou selon une approche vectorielle, en combinant le SOM avec des modèles linéaires et non linéaires (30). Le but de ces travaux était de pouvoir prédire le vecteur des 24 valeurs suivantes, représentant les consommations horaires de la journée suivante. Dans cet

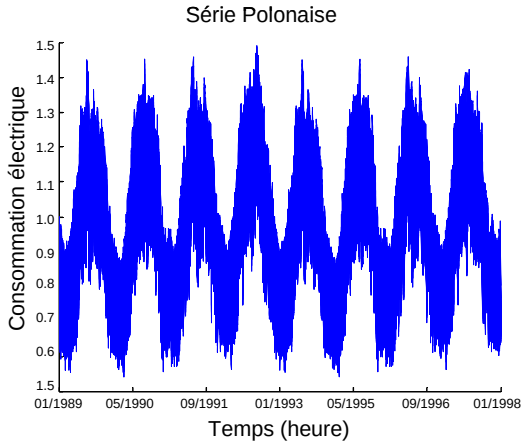

FIG. 3.9 – La série Polonaise des consommations électriques horaires.

exemple, on applique donc la DQV en vectoriel avec $d = 24$, et $h > 1$. Outre l'obtention de valeurs numériques, l'intérêt de cette application est d'illustrer les capacités de prédiction dans le cas vectoriel, et l'obtention d'informations qualitatives complémentaires à long terme.

Le régresseur optimal étant inconnu, de multiples combinaisons ont été essayées, conduisant à la sélection des vecteurs aux instants t, $t-1$, $t-2$, $t-6$ et $t-7$, chaque vecteur correspondant à une journée entière. En d'autres termes, on utilise pour cet exemple les régresseurs :

$$\mathbf{x}_t = (\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), \mathbf{x}(t-6), \mathbf{x}(t-7)) \tag{3.35}$$

Les 3000 données vectorielles $\mathbf{x}(t) = (x(t), x(t-1), \ldots, x(t-23))$ en dimension 24 sont donc devenues 2992 données vectorielles en dimension 120. Ces dernières ont été divisées en trois ensembles d'apprentissage (2 000 premières données vectorielles), de validation (800 suivantes) et de test (les 192 données restantes). Une procédure de validation a été appliquée avec le critère de moyenne des moindres carrés MSE (2.13) afin de sélectionner le modèle de complexité optimale, en faisant varier n_1 et n_2 de 5 à 200 par incréments de 5. Le test n'est pas exhaustif dans ce cas-ci pour des raisons de temps calcul, cette procédure de sélection de structure ayant également servi à la sélection des éléments du régresseur (3.35). Les résultats obtenus pour les différents modèles sont illustrés à la Figure 3.10. On peut observer, en comparant cette Figure avec la Figure 3.6, l'effet des répétitions qui sont réalisées en cross-validation par rapport à la validation utilisée ici : le graphe de l'erreur est beaucoup moins lisse dans ce cas-ci que pour la Figure 3.6. Sur base de ces valeurs de MSE, la structure de modèle sélectionnée utilise $n_1 = 160$ et $n_2 = 140$ prototypes.

Comme pour l'exemple précédent, un nouvel apprentissage est effectué pour le modèle de structure optimale, en utilisant les 2 800 données qui servaient précédemment d'ensembles d'apprentissage et de validation. Sur base de ce modèle, 1 000 simulations sont effectuées. La Figure 3.11 présente les résultats obtenus par la DQV dans cet exemple de

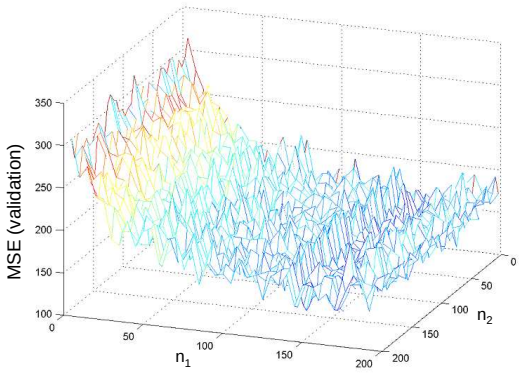

FIG. 3.10 – Graphe de l'estimation de l'erreur de généralisation par validation pour la série des consommations électriques.

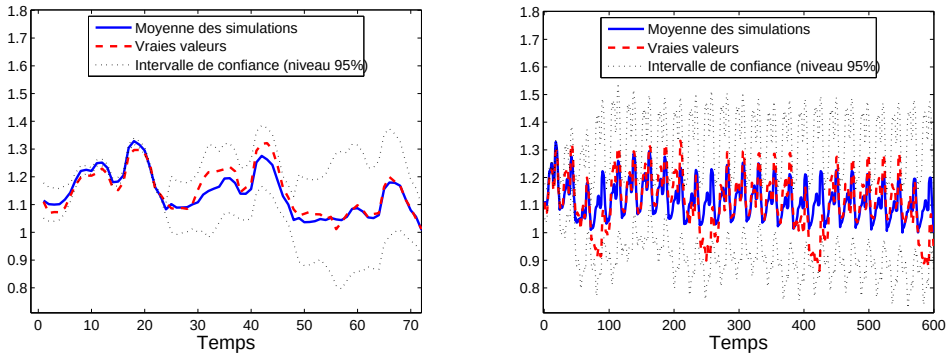

FIG. 3.11 – Gauche : comparaison entre la moyenne des 1 000 simulations (trait plein) et les vraies valeurs de la série des consommations électriques (tirets) pour l'horizon $h = 3$ jours. Les lignes extérieures correspondent à l'intervalle de confiance au niveau 95%. Droite : même figure pour l'horizon $h = 25$ jours.

prédiction vectorielle. La partie gauche de la figure présente la moyenne de 1 000 simulations comparée aux vraies valeurs de la série jusqu'à l'horizon $h = 3$ jours. On peut y observer le comportement de la DQV dans le cadre de la prédiction vectorielle à court terme, le modèle obtenu est capable de réaliser de donnes prédictions numériques, sur toutes les composantes des vecteurs prédits, avec la même précision sur les différentes composantes. La partie droite de la figure illustre l'évolution de la moyenne des 1 000 simulations jusqu'à un horizon $h = 25$ jours.

Comme pour l'exemple précédent, différentes informations qualitatives peuvent être obtenues, même dans le cas de prédictions vectorielles. Par exemple, les intervalles de confiance à 95% sont également représentés. De plus, dans le graphe de gauche de la Figure 3.11, on voit que la moyenne des prédictions à long terme suit la même périodicité que la

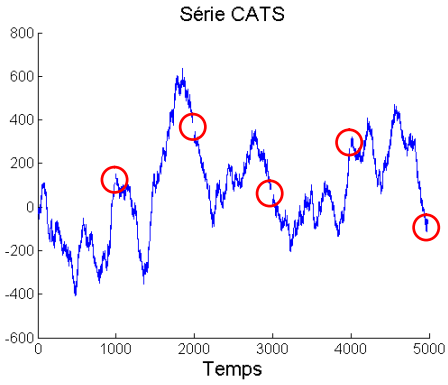

FIG. 3.12 – La série de la compétition CATS. Les cinq blocs de vingt valeurs manquantes sont entourés.

véritable série temporelle, les deux pics quotidiens étant correctement reproduits à court et à long terme. Par ailleurs, les intervalles de confiance décrivent des bornes qui évoluent aussi périodiquement. Ces intervalles de confiance sont, dans ce cas-ci, relativement étroits, indiquant que l'évolution de la série autour de sa moyenne est relativement faible, d'un jour à l'autre. Toutes ces informations qualitatives peuvent être obtenues grâce à la procédure de simulation, qui constitue une caractéristique de la DQV.

3.6.3 CATS

La méthode de Double Quantification Vectorielle a été utilisée lors de la compétition de prédiction de la série temporelle CATS organisée en 2004 dans le cadre de la conférence IJCNN (104).

Brièvement, le but de la compétition CATS est de fournir les 100 prédictions pour 100 valeurs manquantes dans la série temporelle, réparties en cinq blocs de 20 valeurs manquantes. La série temporelle est présentée graphiquement à la Figure 3.12, où les blocs de valeurs manquantes sont entourés. De plus amples détails sur les blocs de valeurs manquantes sont proposés en Annexe A.

Cette série n'ayant fait l'objet d'aucune étude auparavant, et ayant été mise à disposition des participants à la compétition sans information particulière quant à son origine, une analyse détaillée a été effectuée, afin d'estimer la dimension de la structure de la série temporelle et d'en déduire la taille p des régresseurs et des déformations utilisés par la DQV (145). La dimension de corrélation (64) a été estimée pour les 4900 données de la série, sans considérer les valeurs manquantes. Etant donnés les résultats obtenus et l'allure générale de la série, l'hypothèse d'une série financière a été considérée. La série a alors été stationnarisée suivant deux approches :

– calcul de la série différenciée :

$$x_d(t) = x(t+1) - x(t), \tag{3.36}$$

– calcul de la série des returns

$$x_r(t) = \frac{x(t+1) - x(t)}{x(t)}. \tag{3.37}$$

Pour ces deux séries, la dimension de corrélation a également été estimée au moyen de la procédure décrite dans (64) et résumée dans (18). Toutes les expériences décrites ci-dessous ont également été effectuées pour ces deux autres séries temporelles, sans donner de meilleurs résultats en prédiction. Vu le contexte de compétition, seules les prédictions obtenues avec la série initiale ont été soumises aux organisateurs. La méthodologie qui a permis d'obtenir ces prédictions est décrite ici.

Dans le cadre de cette application, outre les valeurs des paramètres n_1 et n_2 de la méthode, il faut décider de l'horizon de prédiction h à utiliser. En effet, la DQV étant en mesure de prédire plusieurs valeurs en une seule étape, il est possible de simuler la série à l'horizon $h = 20$ au moyen de vingt prédictions scalaires, ou alors de simuler la série au moyen de dix prédictions de vecteurs à deux dimensions, ou une seule prédiction d'un vecteur de dimension vingt, sans oublier les quelques possibilités intermédiaires. En d'autres mots, il faut trouver un compromis entre la valeur de d et la précision des prédictions obtenues, jusqu'à l'horizon $h = 20$. Pour optimiser ce nombre, ainsi que les deux paramètres n_1 et n_2, une procédure de cross-validation est utilisée. L'application de la DQV à la série de la compétition CATS a donc été l'occasion de comparer expérimentalement les performances obtenues avec une approche récursive ou une approche directe en prédiction à long terme.

Pour cette série, des ensembles d'apprentissage et de validation ont été créés comme suit, en reproduisant le problème de la compétition. Quinze blocs de vingt valeurs ont été enlevés aléatoirement à l'intérieur de la série temporelle, avec cependant une contrainte : ces blocs ne pouvaient pas coïncider avec les valeurs manquantes de la série initiale ni se chevaucher. Les 300 valeurs enlevées étant en fait connues, elles peuvent servir d'ensemble de validation. De plus, afin d'éviter tout biais dans l'estimation de l'erreur de généralisation dû à la création de ces quinze ensembles de valeurs manquantes, l'opération a été répétée 20 fois. L'estimation finale de l'erreur de généralisation est obtenue sur base du critère MSE en prenant la moyenne sur les 20 créations d'ensemble de validation comptant chacun 300 valeurs. Toutes les structures de modèles pour n_1 et n_2, variant de 5 à 100 par incrément de 5, ont été testées. Comme pour les autres applications, le modèle de structure n_1 et n_2 réalisant le minimum de l'erreur de généralisation est sélectionné comme étant le meilleur modèle. Toute cette procédure a été répétée pour chacune des valeurs possibles du paramètre supplémentaire d. Au final, un modèle comportant $n_1 = 50$ et $n_2 = 5$ prototypes, utilisé pour des régresseurs de dimension $d = 2$ et $p = 3$, a été utilisé pour prédire les 100 valeurs manquantes. Un nouvel apprentissage est alors réalisé pour un modèle de cette structure, sur l'ensemble des 4900 données. Pour éviter tout problème dû à l'initialisation des prototypes de régresseurs ou de déformations, vingt apprentissages différents ont été répétés sur les 4900 données, le modèle réalisant la plus petite erreur

d'apprentissage étant finalement conservé. Notons qu'au final, une approche intermédiaire entre prédiction récursive et directe a été retenue, puisque $d = 2$ a été sélectionnée, par cross-validation.

Puisque le problème était présenté dans un contexte de compétition, deux heuristiques supplémentaires ont été mises en place afin d'améliorer les résultats de la DQV en vue du classement final.

La première heuristique consiste à utiliser le meilleur modèle obtenu dans le sens chronologique inverse afin d'obtenir d'autres prédictions. Bien qu'une autre sélection de structure de modèle aurait idéalement dû être réalisée sur la série dans l'ordre chronologique inverse, le modèle obtenu sur l'ordre chronologique a été utilisé. Notons que cette première heuristique est valable pour les quatre premiers blocs de valeurs manquantes uniquement.

La seconde heuristique était de prédire jusqu'à un horizon $h = 21$ au lieu de se limiter à l'horizon $h = 20$. De la sorte, il est possible, pour les quatre premiers blocs de valeurs manquantes, de comparer la $21^{\text{ème}}$ valeur prédite avec la première valeur réelle. Puisqu'on itère la prédiction, on s'attend à observer une différence entre la $21^{\text{ème}}$ prédiction et la véritable valeur. Il est alors possible de corriger légèrement cette erreur en multipliant chacune des vingt prédictions par un coefficient. Le coefficient est défini différemment pour chacune des vingt prédictions, et est choisi proportionnellement à l'écart dans le temps entre la prédiction et la $21^{\text{ème}}$ prédiction. Le but est de faire en sorte que ce ne soit pas la seule $20^{\text{ème}}$ prédiction qui supporte à elle seule l'entièreté de la correction apportée aux valeurs. Concrètement, la première prédiction n'est pas corrigée, la deuxième prédiction est corrigée en lui soustrayant 5% de la différence entre la $21^{\text{ème}}$ valeur prédite et la vraie valeur correspondante, la troisième prédiction est corrigée en lui soustrayant 10% de cette différence, la quatrième en soustrayant 15% de la différence, etc., jusqu'à la $21^{\text{ème}}$ à laquelle on enlève 100% de cette différence afin de la faire coïncider avec la vraie valeur. Bien entendu, pour les quatre premiers blocs de valeurs manquantes, les corrections sont appliquées, dans les deux sens, chronologique et chronologique inverse. Par contre, aucune heuristique n'est applicable dans le cas du dernier bloc de valeurs manquantes et seule la moyenne des 100 simulations peut être utilisée dans ce cas.

Enfin, signalons que toutes les prédictions, que ce soit dans l'ordre chronologique ou dans l'ordre chronologique inverse, sont obtenues au moyen de 100 simulations jusqu'à l'horizon de prédiction final. La Figure 3.13 présente les résultats obtenus avec le meilleur modèle sélectionné, en manipulant les prédictions suivant les deux heuristiques décrites ci-dessus, et soumis aux organisateurs de la compétition.

Comme mentionné précédemment, toute la méthodologie décrite ici a été appliquée aux deux séries temporelles des différences et des returns, mais sans fournir de meilleurs résultats (145). Seuls les résultats illustrés à la Figure 3.13 ont donc été utilisés lors du classement final de la compétition.

Deux critères étaient utilisés pour ce classement, le premier mesurant l'erreur quadratique sur les cinq blocs de valeurs manquantes, le second sur les quatre premiers blocs

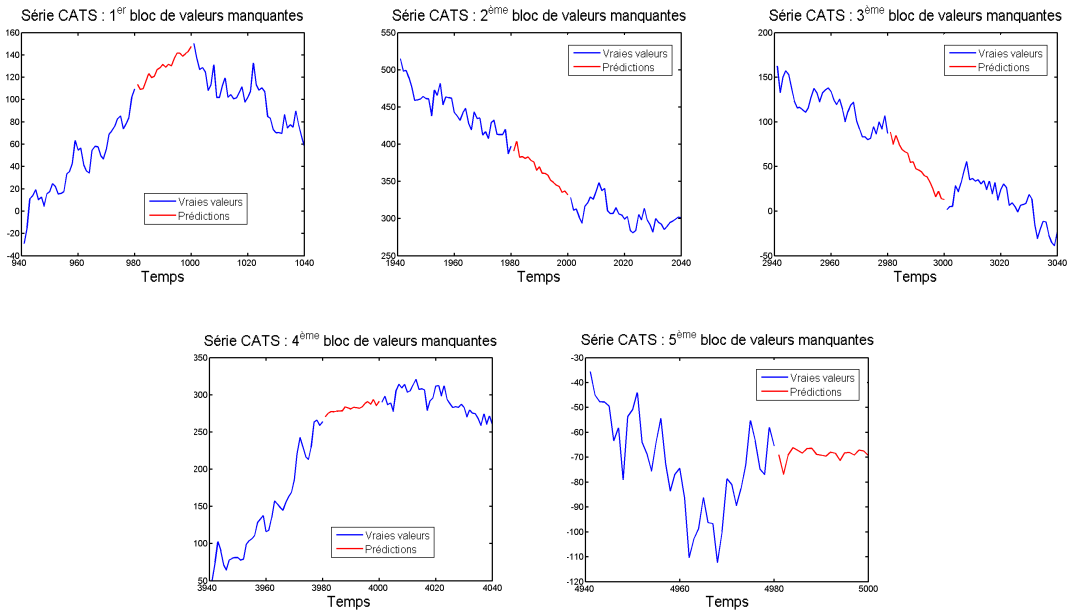

FIG. 3.13 – Prédictions obtenues par la DQV sur la série CATS, avec deux heuristiques spécifiques, pour le premier bloc de valeurs manquantes jusqu'au cinquième respectivement de gauche à droite et de haut en bas. Pour les quatre premiers blocs de valeurs manquantes, les vingt prédictions sont disjointes des vraies valeurs qui les entourent à gauche et à droite, sauf pour le cinquième où les vraies valeurs ne figurent que sur la partie de gauche du graphe.

seulement. Selon le second critère, la DQV s'est finalement classée quatrième sur 24 participants dont 17 classés (104). Concernant le premier critère, la DQV figure à la septième position (104). Le classement moins bon obtenu selon le premier critère peut s'expliquer en partie par l'utilisation des deux heuristiques qui ont permis de corriger les prédictions obtenues pour les quatre premiers blocs de valeurs manquantes, ce qui ne fût pas le cas pour le dernier bloc.

3.7 Prédiction récursive, prédiction directe : discussion sur les performances à long terme

Lors de l'application de la DQV à la série CATS, les approches récursive et directe ont été testées afin de sélectionner la meilleure méthodologie de prédiction à long terme dans le cadre de la compétition CATS. Au final, une approche intermédiaire a été retenue, entre la prédiction de 20 valeurs scalaires par approche récursive et la prédiction directe d'un vecteur de 20 valeurs : la prédiction par approche récursive d'un vecteur de deux valeurs. On peut donc se demander s'il existe, en toute généralité, une approche supérieure lorsqu'on essaie de prédire à long terme. Une réponse théorique à cette question apporterait une information très importante en pratique pour tout problème de prédiction à long

terme. Il serait en effet possible de savoir quelle méthodologie de prédiction utiliser en fonction de l'horizon considéré. Dans cette section, une formalisation du problème de prédiction à long terme est proposée, les erreurs obtenues par approches itérative et directe sont définies de façon rigoureuse. La conclusion de la formalisation détaillée ci-dessous est qu'il n'est pas possible de comparer les erreurs obtenues à long terme. Le développement proposé ne permet donc pas de décider si une des deux approches est supérieure à l'autre.

Au Chapitre 2, un modèle a été défini de façon générale selon la relation (2.11) rappelée ici :

$$\hat{y}(t) = f_s\left(\mathbf{x}(t), \theta_s(\mathcal{D})\right). \tag{3.38}$$

Pour rappel, dans ce chapitre, il a également été expliqué comment les méthodes de sélection de structure de modèle de type validation et/ou cross-validation divisent l'ensemble $\mathcal{D}$ en deux ensembles d'apprentissage $\mathcal{A}$ et de validation $\mathcal{V}$, respectivement de taille $N_\mathcal{A}$ et $N_\mathcal{V}$ avec $N_\mathcal{A} + N_\mathcal{V} = N$.

Dans cette section, il est fait l'hypothèse que les modèles utilisés, que ce soit en prédiction à court terme ou en prédiction à long terme, sont optimaux, c'est à dire de structure optimale et avec des paramètres dont la valeur est optimale. Autrement dit, on note un modèle optimal selon :

$$f_{s^\star}\left(\mathbf{x}(t), \theta^\star_{s^\star}(\mathcal{D})\right), \tag{3.39}$$

où le symbole * désigne l'optimalité. Cette optimalité peut être résumée comme suit :

$$s^\star = \arg\min_s \sum_{t=1}^{N_\mathcal{V}} \frac{\left(f_s\left(\mathbf{x}(t), \theta^\star_s(\mathcal{A})\right) - y(t)\right)^2}{N_\mathcal{V}}, \tag{3.40}$$

avec $\theta^\star_s(.)$ obtenu pour une structure s fixée selon :

$$\theta^\star_s(.) = \arg\min_{\theta_s(.)} \sum_{t=1}^{N_\mathcal{A}} \frac{\left(f_s\left(\mathbf{x}(t), \theta_s(\mathcal{A})\right) - y(t)\right)^2}{N_\mathcal{A}}. \tag{3.41}$$

En d'autres termes, $\theta^\star_s(.)$ minimise l'erreur d'apprentissage, selon le critère des moindres carrés MSE, alors que $s^\star$ minimise l'erreur de validation, toujours selon le critère MSE.

Considérons à présent le cas de la prédiction récursive (2.31). Un modèle optimal pour de la prédiction à long terme par approche récursive est en fait un modèle de prédiction à un pas utilisé de façon répétée. Un modèle optimal de prédiction à un pas se note donc :

$$f^1_{s^\star}\left(\mathbf{x}(t), \theta^\star_{s^\star}(\mathcal{A})\right). \tag{3.42}$$

De manière similaire, on définit le modèle optimal pour de la prédiction à un horizon h quelconque selon l'approche directe (2.32) par :

$$f^h_{s^\star}\left(\mathbf{x}(t), \theta^\star_{s^\star}(\mathcal{A})\right). \tag{3.43}$$

Pour essayer de déterminer s'il existe une approche préférable dans tous les cas, on observe à présent les erreurs commises par ces deux modèles lorsqu'ils sont utilisés pour

prédire à un horizon $h > 1$ fixé. Considérons un instant t pour lequel la valeur $y(t+h)$ à l'horizon h est connue. On suppose néanmoins que cette valeur $y(t+h)$ n'a pas été utilisée ni pour l'apprentissage ni pour la validation. $y(t+h)$ est donc une donnée de l'ensemble de test. L'erreur obtenue par les deux modèles s'écrit d'une part :

$$\begin{aligned}
e_1 &= \left(f^1_{s^\star}\left(x(t), x(t-1), \ldots, x(t-p+1), \theta^\star_{s^\star}(\mathcal{A})\right) - y(t+1)\right)^2, \\
e_2 &= \left(f^1_{s^\star}\left(\hat{x}(t+1), x(t), \ldots, x(t-p+2), \theta^\star_{s^\star}(\mathcal{A})\right) - y(t+2)\right)^2, \\
e_3 &= \left(f^1_{s^\star}\left(\hat{x}(t+2), \hat{x}(t+1), \ldots, x(t-p+3), \theta^\star_{s^\star}(\mathcal{A})\right) - y(t+3)\right)^2, \\
&\vdots \\
e_h &= \left(f^1_{s^\star}\left(\hat{x}(t+h-1), \hat{x}(t+h-2), \ldots, \hat{x}(t-p+h), \theta^\star_{s^\star}(\mathcal{A})\right) - y(t+h)\right)^2,
\end{aligned} \tag{3.44}$$

où les composantes du régresseur ont été décrites explicitement et $h > p$; et d'autre part :

$$e'_h = \left(f^h_{s^\star}\left(\mathbf{x}(t), \theta^\star_{s^\star}(\mathcal{A})\right) - y(t+h)\right)^2. \tag{3.45}$$

Pour savoir si l'approche récursive est supérieure à l'approche directe, ou inversement, il faut pouvoir comparer e'_h et e_h. Malheureusement, une comparaison directe n'est pas possible, les grandeurs étant tout simplement incomparables entre elles : de par leur définition les grandeurs sont conceptuellement différentes (même s'il s'agit d'une erreur de moindre carrés dans les deux cas). Par conséquent, la question ne peut être tranchée suivant l'approche proposée ci-dessus.

Cette question de la performance des modèles de prédiction à long terme a déjà fait l'objet de quelques études dans la littérature. Il est par exemple expliqué que la prédiction directe à long terme finit par ne donner qu'une constante, la moyenne des données, qui permet certes de minimiser l'erreur de généralisation mais ne permet plus de faire une prédiction de qualité suffisante (88). Il est également indiqué dans (88) que l'approche itérative a l'avantage de produire une série temporelle qui, même si elle s'éloigne de plus en plus de la vraie série lorsqu'on réinjecte des valeurs prédites entachées d'erreur, possède les mêmes propriétés statistiques (intervalle de valeur, moyenne, variance, ...) que la véritable série. L'approche récursive dans le cas de la prédiction à long terme est déconseillée dans (167) lorsque le modèle considéré est non linéaire, la prédiction à long terme par un modèle direct étant plus adaptée dans ce cas. Cependant, il est également affirmé qu'en pratique, l'approche récursive peut se montrer supérieure à l'approche directe, notamment lorsque le modèle n'est pas correctement construit ou lorsque le bruit dans les données est important (167). L'idéal pourrait donc venir d'une approche intermédiaire, qui combine à la fois les approches récursive et directe. La méthode proposée dans (154) combine justement ces deux approches de prédiction comme suit : le régresseur utilisé par un modèle de prédiction directe à un horizon de prédiction h fixé est construit sur base du régresseur utilisé par un autre modèle direct à un horizon $h-1$ et de la prédiction obtenue par ce modèle de prédiction directe à un horizon $h-1$; les régresseurs augmentant donc évidemment lorsque l'horizon h augmente. Cette approche méthodologique est en tous cas

très intéressante d'un point de vue théorique, mais surtout très utile sur le plan pratique puisqu'elle combine les avantages des deux approches de prédiction, directe et récursive. Elle a néanmoins le désavantage non négligeable de voir la taille des régresseurs augmenter avec l'horizon de prédiction h, ce qui se traduit par un modèle comportant de plus en plus de paramètres, et dont les valeurs sont de plus en plus délicates à optimiser. La question de la supériorité d'une approche par rapport à l'autre est donc encore une question ouverte dans la littérature, et la formalisation du problème proposée ici ne permet pas non plus de conclure, ni dans un sens ni dans l'autre.

3.8 Méthodes proches de la Double Quantification Vectorielle

La méthode DQV se base sur une double application d'un algorithme de quantification vectorielle à des ensembles de données vectorielles et sur la construction d'une matrice des transitions. Ces deux caractéristiques de la DQV sont également à la base de deux autres méthodes d'analyse et de prédiction de séries temporelles présentées dans cette section. L'intérêt de cette section est de montrer la richesse des étapes de double quantification et de construction d'une matrice des transitions, en illustrant concrètement qu'il est possible de construire d'autres méthodes que la DQV avec ces deux "outils de base". Dans un premier temps, l'idée d'une double utilisation d'un algorithme de quantification vectorielle est combinée avec une matrice des transitions et des modèles RBFN locaux dans les différents clusters ainsi construits. Dans un deuxième temps, une unique quantification vectorielle est combinée avec deux matrices de transitions. La description qui va suivre des deux méthodes originales sera relativement brève, et principalement axée sur la manière dont on peut manipuler ces outils de base que sont la double quantification et la matrice des transitions sont utilisés. Ces deux méthodes ont fait l'objet de publications auxquelles le lecteur peut se référer.

3.8.1 Double Quantification et RBFN locaux

La méthode de double quantification avec modèles RBFN locaux (34; 35) est relativement proche de la DQV, à quelques différences près expliquées ici. Cette méthode a été développée spécifiquement pour la prédiction de séries financières.

Conceptuellement, l'idée de base est de modéliser la dynamique de la série au moyen d'un ensemble de régresseurs $\mathbf{x}_t$ définis comme précédemment selon la relation (2.30), à savoir $\mathbf{x}_t = (x(t), x(t-1), \ldots, x(t-p+1))$, et d'un ensemble de régresseurs *étendus* $\mathbf{x'}_t$. Ces régresseurs étendus sont obtenus par concaténation de la valeur suivante de la série temporelle au régresseur, selon :

$$\mathbf{x'}_t = (x(t+1), x(t), x(t-1), \ldots, x(t-p+1)) \,. \tag{3.46}$$

Notons qu'ici aussi, pour chaque régresseur $\mathbf{x}_t$, il existe un et un seul régresseur étendu $\mathbf{x'}_t$ qui peut lui être associé, et réciproquement. L'algorithme SOM est ensuite utilisé pour

partitionner respectivement les ensembles de régresseurs et de régresseurs étendus. On construit alors une table comparable à la matrice des transitions, table dans laquelle on reprend la proportion de régresseurs étendus $\mathbf{x'}_t$ appartenant à un cluster de la seconde carte auto-organisées, sachant à quel cluster de la première carte appartient le régresseur $\mathbf{x}_t$ qui lui est associé. Par ailleurs, puisque les régresseurs étendus contiennent par construction la valeur de sortie attendue, on peut utiliser un modèle supervisé, par exemple un RBFN, pour apprendre localement la relation entrées-sortie, à l'intérieur de chaque cluster de régresseurs étendus.

Dans cette méthode, la prédiction est obtenue directement, sans simulation, au moyen d'une somme pondérée par les fréquences empiriques des différentes prédictions obtenues par les RBFN locaux. Partant d'un régresseur $\mathbf{x}_t$ au temps t, on cherche le cluster c_k du prototype $\bar{\mathbf{x}}_k$ le plus proche de $\mathbf{x}_t$ parmi tous les prototypes $\bar{\mathbf{x}}_i$, $1 \leq i \leq N$. Pour les régresseurs appartenant à c_k, on connaît les régresseurs étendus qui leur sont associés et qui sont situés dans quelques clusters de régresseurs étendus. Les modèles RBFN de ces quelques clusters sont utilisés afin de prédire, chacun séparément, la valeur suivante de la série. Les prédictions fournies par les RBFN locaux sont alors pondérées par les fréquences de transitions du cluster c_k vers les quelques clusters de régresseurs étendus qui peuvent être atteint à partir des régresseurs de c_k, ce qui constitue la prédiction finale du modèle.

Par rapport à la DQV, les différences principales sont donc :

- la création de régresseurs étendus à la place des déformations,
- l'utilisation de modèles RBFN locaux,
- la prédiction par somme pondérée des valeurs obtenues par les RBFN locaux, sans procédure de simulation.

Notons que, dans cette méthode, outre les tailles respectives n_1 et n_2 des SOM, il faut déterminer le nombre m de noyaux Gaussiens qui doivent être utilisés, et ce pour chacun des $n_1 * n_2$ modèles RBFN utilisés. Ce qui dans le cas de la DQV impliquait déjà une double application d'une méthode de sélection de structure en demande une triple dans le cadre de cette méthode.

L'illustration des capacités de cette méthode de modélisation et de prédiction de séries temporelles dans le cadre de la prédiction à un pas de la série financière du DAX30 est fournie dans (34; 35).

3.8.2 Double Matrice des Transitions

Cette méthode, dont le but est de développer un outil permettant de mettre en évidence l'inefficience des marchés financiers sur des intervalles de temps très courts, utilise non pas deux quantifications vectorielles et une matrice des transitions, mais une seule quantification vectorielle et deux matrices des transitions.

Conceptuellement, après quantification par l'algorithme SOM de l'ensemble des régresseurs obtenus d'une série temporelle, on construit deux matrices de transitions : une matrice théorique et une matrice empirique. La principale différence entre ces deux matrices est que la matrice théorique ne tient pas compte de la dépendance temporelle entre les

régresseurs, ce que fait la matrice empirique. En observant les distributions de probabilités de transitions contenues dans ces deux matrices, on peut montrer qu'il existe un certain déterminisme dans la matrice empirique, alors que la matrice théorique n'en présente pratiquement aucun. Cette différence de comportement dû à la prise en compte de la dépendance temporelle est interprétée comme une preuve empirique de l'existence d'une certaine dynamique dans l'évolution de la série. La mise en évidence de l'existence d'une telle dynamique tend à prouver, dans le cadre de séries temporelles financières, l'inefficience des marchés financiers sur des intervalles de temps très courts.

Plus précisément, partant d'une série de valeurs scalaires, on construit les régresseurs $\mathbf{x}_t$, selon la relation (2.30). On applique ensuite l'algorithme SOM avec une carte par exemple rectangulaire, ce qui permet ensuite de partitionner les données au moyen des prototypes $\bar{\mathbf{x}}_i$ et des clusters c_i ainsi obtenus, $1 \leq i \leq n$. Pour chaque prototype i de la grille rectangulaire, i étant fixé, on définit une matrice théorique locale $TH_i(.)$ des transitions entre ce prototype i et les prototypes situés dans son voisinage Voisinage($\bar{\mathbf{x}}_i$), la limite de voisinage étant ici fixée à 1. On considère donc les prototypes à une distance de 1 sur la grille du SOM, ainsi que le prototype $\bar{\mathbf{x}}_i$ lui-même. Formellement, chaque élément j d'une matrice théorique locale des transitions de prototype i se définit selon la relation :

$$TH_i(j) = \frac{\#\left\{\mathbf{x}_t \in c_j \mid \bar{\mathbf{x}}_j \in \text{Voisinage}(\bar{\mathbf{x}}_i)\right\}}{\#\left\{\mathbf{x}_t \in \text{Voisinage}(\bar{\mathbf{x}}_i)\right\}}. \tag{3.47}$$

En d'autres termes, les valeurs $TH_i(j)$ dénotent les fréquences de transitions théoriques entre les régresseurs $\mathbf{x}_t$ du cluster c_i et les régresseurs $\mathbf{x}_t$ situés dans les clusters c_j, $1 \leq j \leq n$, dont les prototypes sont des voisins de $\bar{\mathbf{x}}_i$ situés à une distance de 1 sur la grille du SOM.

Afin de simplifier la représentation des éléments d'une matrice $TH_i(.)$, ceux-ci sont représentés dans une matrice 3 * 3 en respect de la position respective des prototypes $\bar{\mathbf{x}}_j$ sur la grille du SOM par rapport au prototype $\bar{\mathbf{x}}_i$, en utilisant des voisinages carrés. Le prototype $\bar{\mathbf{x}}_i$ est positionné au centre de cette matrice 3 * 3. La matrice théorique finale des transitions TH est définie comme la concaténation des matrices théoriques locales des transitions $TH_i(.)$, en respectant la topologie du SOM, comme par exemple :

$$TH = \begin{pmatrix} TH_1(.) & TH_3(.) & TH_5(.) & TH_7(.) \\ TH_2(.) & TH_4(.) & TH_6(.) & TH_8(.) \end{pmatrix}, \tag{3.48}$$

pour une carte auto-organisées utilisant une grille rectangulaire de 2 * 4 prototypes.

De manière similaire, on construit des matrices empiriques locales $EM_i(.)$ des transitions. Pour chaque prototype i de la grille rectangulaire, i étant fixé, on calcule maintenant les éléments $EM_i(j)$ en tenant compte des dépendances temporelles entre $\mathbf{x}_t$ et $\mathbf{x}_{t+1}$ selon :

$$EM_i(j) = \frac{\#\left\{\mathbf{x}_{t+1} \in c_j \mid \mathbf{x}_t \in c_i\right\}}{\#\left\{\mathbf{x}_t \in c_i\right\}}. \tag{3.49}$$

Les valeurs $EM_i(j)$ dénotent donc les fréquences de transitions empiriques quand, partant d'un régresseur $\mathbf{x}_t$ appartenant au cluster c_i, on se retrouve dans un des clusters c_j voisins

de c_i auquel appartient le régresseur suivant $\mathbf{x}_{t+1}$. Pour cette matrice également, la limite de voisinage est fixée à 1. Les matrices $EM_i(.)$ sont elles aussi de dimension 3 * 3, et elles sont regroupées par concaténation, tout comme pour les matrices théoriques locales, dans une matrice EM, suivant la topologie de la grille utilisée.

L'idée de la méthode est de comparer visuellement la distribution des fréquences de transitions obtenues dans le cas théorique et dans le cas empirique. Considérons le cas d'un régresseur $\mathbf{x}_t$ à un instant t. La transition vers l'instant $t+1$ va conduire au régresseur $\mathbf{x}_{t+1}$. S'il y a une tendance, ou un déterminisme, dans l'évolution de la série temporelle, la transition de tout régresseur de forme comparable à $\mathbf{x}_t$ vers son régresseur suivant devrait conduire vers le cluster contenant les régresseurs semblables à $\mathbf{x}_{t+1}$, par le respect de la topologie de l'algorithme SOM. Tout comme pour la DQV, on devrait donc avoir des fréquences importantes pour certaines transitions et nulle pour d'autres. Par contre, en l'absence de tendance claire dans l'évolution de la série, les transitions devraient être plus ou moins identiquement distribuées. Intuitivement, ce raisonnement se justifie par le fait que, dans la matrice théorique des transitions, on ne tient pas compte des dépendances temporelles, uniquement des fréquences d'appartenance aux clusters. Si l'algorithme SOM a correctement convergé, ces fréquences d'appartenance aux clusters devraient être sensiblement les mêmes. Par contre, le fait de tenir compte des dépendances temporelles dans la construction de la matrice empirique des transitions devrait permettre de mettre en évidence la tendance d'évolution existant dans la série, s'il y en a une.

La Figure 3.14 présente les matrices TH et EM obtenues pour la série Santa Fe A, respectivement à gauche et à droite, en utilisant des cartes carrées de dimension 4 * 4. Les 16 matrices locales sont, dans les deux figures, symbolisées par 16 grandes astérisques associées à chaque prototype $\bar{\mathbf{x}}_i$, $1 \leq i \leq 16$ et organisées selon la topologie de la carte. Chaque grande astérisque couvre une matrice théorique locale, ou empirique locale, de dimension 3 * 3. L'importance relative des fréquences théoriques et empiriques de transitions vers les régresseurs des clusters voisins est symbolisée par un niveau de gris, dont l'échelle est reprise sur la droite de chaque matrice. Bien évidemment, les transitions partant d'un prototype sur le bord de la grille vers l'extérieur de la grille sont nulles puisqu'il n'y a pas de voisin dans cette direction ; les cases correspondantes sont donc blanches.

Grâce à ces deux matrices, on constate la présence d'un déterminisme clair dans la série Santa Fe A. En effet, selon la matrice théorique, pour à peu près n'importe quel prototype i, la transition de $\mathbf{x}_t$ appartenant au cluster c_i vers $\mathbf{x}_{t+1}$ peut conduire à n'importe quel cluster c_j dont le prototype est dans le voisinage du prototype i, avec à peu près la même fréquence théorique. Cette affirmation est soutenue par la représentation visuelle de la matrice théorique des transitions, dont la couleur est d'un gris plus ou moins uniforme. Par contre, pour la matrice empirique, les fréquences de transitions ne sont plus du tout uniformes, ce qui peut être aisément observé par la grande différence des niveaux de gris entre les différents éléments de chaque matrice 3 * 3. Dans certains cas (coins supérieur gauche et inférieur gauche), il n'y a même qu'un seul type de transitions possible, vers un voisin unique. La comparaison entre les matrices théorique et empirique des transitions de

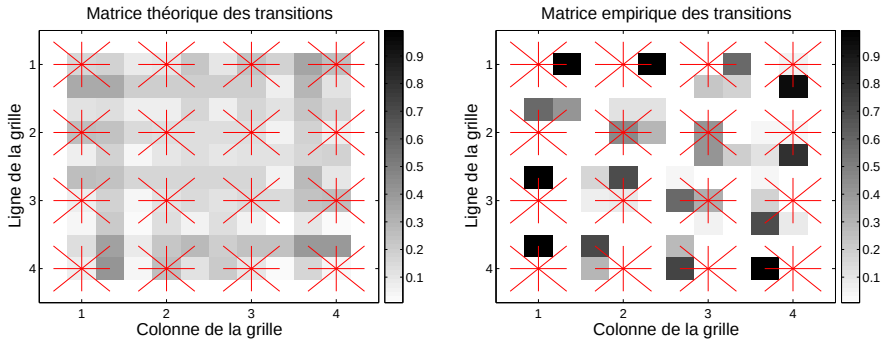

FIG. 3.14 – Matrices théorique et empirique des transitions obtenues pour la série Santa Fe A.

la Figure 3.14 montre donc qu'il y a une dynamique d'évolution réelle pour la série Santa Fe A.

Cette méthode de détection de tendance dans l'évolution d'une série temporelle a été appliquée à des séries financières issues du marché des titres de la bourse de Paris (22). Plusieurs séries temporelles, correspondant aux offres de vente et d'achat de certaines actions sur des intervalles de temps relativement courts ont été construites. En appliquant la méthode de double matrice des transitions, il a été permis de montrer que la présence d'un déterminisme dans toutes les séries considérées, pour des intervalles de temps courts, ce qui permet de montrer empiriquement l'inefficience des marchés financiers. Cette conclusion, obtenue pour le marché des titres de la bourse de Paris, complète au moyen d'une approche non linéaire les résultats publiés dans la littérature par (26) et obtenus au moyen de modèles linéaires sur le marché NYSE de New York.

3.9 Résumé du chapitre

Dans ce chapitre, la méthode de Double Quantification Vectorielle a été introduite. Cette méthode permet de modéliser et de prédire une série temporelle en se basant sur l'algorithme SOM. L'utilisation des SOM dans le contexte temporel a été discutée.

Ensuite, la phase d'apprentissage de la DQV et l'utilisation du modèle obtenu en prédiction ont été décrites dans le cas simple des séries temporelles scalaires, avec prédiction de la valeur suivante.

Les capacités de la méthode à long terme ont été discutées, ce qui a permis de généraliser la méthode au cas vectoriel et de montrer toute la souplesse de la DQV. Il est en effet possible de prédire très facilement aussi bien à un horizon $h = 1$ qu'à tout autre horizon de prédiction $h > 1$, de façon récursive ou directe, aussi bien en scalaire qu'en vectoriel. Mentionnons surtout que toutes ces options sont possibles sans aucune modification particulière de l'algorithme.

La preuve théorique de la stabilité de la méthode DQV dans le cadre de la prédiction à long terme a ensuite été proposée. Cette preuve se base sur les chaînes de Markov et est détaillée ici pour le cas de régresseurs en deux dimensions, lors de prédiction à long terme obtenues par approche récursive. L'idée de la preuve est de prouver que la suite des prédictions est une chaîne de Markov et que cette chaîne est ergodique, cette propriété impliquant la stabilité. Le résultat de cette démonstration a été discuté dans le contexte de la prédiction récursive et directe, où il s'applique également. Il a alors été expliqué que la généralisation du développement de la preuve pour des régresseurs de dimension supérieure à deux n'est pas immédiate et demande pour pouvoir être établie, de prouver des résultats équivalents aux Lemmes 2. et 3.

Plusieurs applications ont permis de montrer la grande souplesse d'utilisation de la méthode DQV. La prédiction scalaire a d'abord été illustrée en utilisant la série Santa Fe A ; la prédiction vectorielle a été étudiée au moyen d'un problème de prédiction de la consommation électrique. Les résultats obtenus lors de la compétition de prédiction CATS ont également été présentés, permettant de situer les performances de la méthode par rapport à d'autres méthodes de prédiction.

La DQV permettant de prédire en itératif et en direct, il a été discuté de façon théorique si l'une ou l'autre approche pouvait être supérieure, en toute généralité, pour de la prédiction à long terme. La conclusion de cette discussion est qu'il n'y a pas de méthode supérieure dans tous les cas, et que l'approche de prédiction à long terme doit être choisie en fonction de la série et des objectifs de l'application.

Enfin, en complément à la DQV, la méthode de double quantification avec modèles RBFN locaux a également été présentée. Cette méthode est conceptuellement très proche de la DQV, la principale différence résidant dans l'utilisation de modèles locaux qui nécessitent de manipuler un autre type de régresseurs et qui permettent d'obtenir une prédiction sans procédure de simulation. Par ailleurs, une méthode utilisant une matrice des transitions théorique et une matrice des transitions empirique a été plus longuement décrite. Il a été détaillé comment cette méthode utilise une représentation graphique des probabilités de transition, suivant la topologie des grilles de l'algorithme SOM, pour détecter la présence de déterminisme dans l'évolution d'une série temporelle. Grâce à cette méthode, il a été possible de montrer expérimentalement l'inefficience du marché des titres de la bourse de Paris.

Chapitre 4

L'importance de la sélection du délai dans la construction de régresseurs

Ce chapitre aborde brièvement la question de l'analyse de séries temporelles et souligne l'importance de cette étape dans la construction d'un modèle. La question est précisée dans un cas concret, à savoir la sélection du délai dans la construction de régresseurs. L'impact de cette sélection du délai est étudié dans le cadre de l'application des algorithmes de quantification vectorielle à un ensemble de régresseurs. Des critères de comparaison sont introduits afin de pouvoir caractériser les distributions de prototypes obtenues lors de différentes applications des algorithmes de quantification vectorielle. Des résultats expérimentaux mettent en évidence l'importance d'une analyse et de la sélection du délai, en comparant les cas des régresseurs construits respectivement soit avec un délai fixé à un soit avec un délai quelconque. Une contribution de ce chapitre est une heuristique géométrique qui permet de sélectionner très intuitivement un délai quelconque. Une autre contribution de ce chapitre est l'utilisation de cette heuristique géométrique dans un débat actuellement en cours et portant sur le caractère significatif de l'application des algorithmes de clustering à des données de type régresseurs issus de séries temporelles. Un résumé clôture ce chapitre en introduisant le Chapitre 5, où l'accent est mis sur la sélection du délai.

4.1 Analyse de séries temporelles

En toute généralité, l'analyse de données peut être déterminante lors de la construction d'un modèle. En effet, une analyse permet de prendre connaissance des données à disposition et d'en retirer un certain nombre d'informations. Habituellement, l'étape d'analyse permet par exemple d'appréhender l'ensemble des données au travers de quelques statistiques, comme la moyenne, l'écart-type, l'intervalle de valeurs, etc. L'analyse permet aussi de détecter l'éventuelle présence de données manquantes, d'observer des "outliers", d'estimer l'importance du bruit dans les données lorsque c'est possible, etc.

Dans le contexte temporel, analyser une série, c'est notamment observer si la série est stationnaire ou non, c'est détecter une tendance saisonnière, etc. Toutes ces opérations sont des manipulations relativement classiques permettant de se familiariser avec un ensemble de valeurs temporelles. Par ailleurs, ces opérations indiquent quel(s) prétraitement(s) il est utile, voire parfois indispensable, d'appliquer aux données. Dans certains cas, l'analyse influence même le choix du modèle qui sera utilisé parmi les nombreux modèles possibles, rendant plus opportun suivant les cas l'emploi d'un modèle linéaire, ou d'un modèle avec données exogènes, ou d'un modèle avec sortie vectorielle, etc.

Au-delà de ces opérations proposant déjà une série d'informations sur la série considérée, d'autres informations peuvent être obtenues par une analyse non linéaire des séries temporelles (88). Ce nom recouvre un ensemble de techniques issues de la physique et de l'étude des systèmes dynamiques. Certaines de ces techniques sont détaillées dans le chapitre 5.

Le but de ce chapitre est de souligner l'importance de l'étape d'analyse dans la construction d'un modèle. La question du choix du délai entre variables d'un régresseur, sélectionné au moyen d'une analyse de la série, et de l'impact de l'utilisation d'un délai sur le modèle obtenu est au coeur de ce chapitre. Différentes techniques d'analyse, et plus précisément de sélection de délai, sont présentées au chapitre suivant.

4.2 Construction de régresseurs et choix du délai

Dans le chapitre 2, un modèle a été défini de façon générale suivant la relation (2.11) rappelée ici :

$$\hat{y}(t) = f_s\left(\mathbf{x}_t, \theta_s(\mathcal{D})\right), \tag{4.1}$$

où $\theta_s(\mathcal{D})$ représente les différents paramètres du modèle f de structure s dont les valeurs respectives ont été déterminées sur l'ensemble $\mathcal{D}$ lors de l'apprentissage. $\mathbf{x}_t$ est le régresseur et $\hat{y}(t)$ est l'estimation calculée par le modèle de la valeur $y(t)$ attendue en sortie.

Dans le cadre de la modélisation de séries temporelles scalaires, cette définition générale peut être précisée suivant la relation (2.29), généralisée ici pour un délai τ quelconque, mais fixe, entre deux valeurs successives du régresseur :

$$\hat{x}(t+1) = f_s\left(x(t), x(t-\tau), \ldots, x(t-\tau*(p-1)), \theta_s(\mathcal{D})\right). \tag{4.2}$$

Le vecteur des composantes reprises dans le modèle est le régresseur avec délai :

$$\mathbf{x}_t = \left(x(t), x(t-\tau), \ldots, x(t-\tau*(p-1)\right). \tag{4.3}$$

La dépendance au délai τ n'est pas indiquée explicitement dans la notation $\mathbf{x}_t$ du régresseur avec délai (4.3). En effet, même pour les régresseurs utilisés jusqu'ici, et définis suivant (2.30), un délai $\tau = 1$ était déjà utilisé. (2.30) n'est donc qu'un cas particulier du cas général (4.3) décrit ici.

Notons qu'en pratique, une série temporelle est une suite de valeurs obtenues à partir d'un système évoluant au cours du temps, par mesures successives à une période

d'échantillonage T fixée. T correspond donc à l'intervalle de temps écoulé entre deux mesures de la série temporelle. Le délai τ est quant à lui l'intervalle de temps entre deux valeurs comprises dans le régresseur. Dans tout modèle utilisant des régresseurs, le délai τ est choisi comme étant un multiple de T.

Jusqu'à présent, le paramètre p a été choisi au cas par cas, en fonction de la série temporelle considérée. De même, le délai τ entre deux valeurs reprises dans le régresseur, a été fixé à 1 dans la relation (2.29). Pourtant, lors des applications de la DQV décrites à la section 3.6 du chapitre précédent, tant la taille p du régresseur que la valeur de ce délai τ ont été fixées suivant des connaissances a priori comme dans le cas de la série Santa Fe A (sections 2.2.4 et 3.6.1), ou sélectionnées par cross-validation dans le cas de la série de la consommation électrique (section 3.6.2), ou encore en laissant τ fixé à un comme pour la série CATS (section 3.6.3). Aucun détail n'a été donné jusqu'à présent sur les méthodes permettant d'obtenir une estimation de ces valeurs p et τ. Le chapitre 5 discute du choix des valeurs de p et, plus en détails, de τ. Dans ce chapitre, l'importance de l'utilisation d'un délai τ dans la construction des régresseurs est mise en évidence au travers d'un cas concret qui fait actuellement l'objet d'un débat dans la communauté scientifique.

4.3 Le clustering de régresseurs est-il significatif ?

Un article publié récemment pose la question du caractère significatif des algorithmes de clustering lorsqu'ils sont appliqués à des données de type régresseurs issus de séries temporelles (94). La thèse de cet article est que, dans ces applications, le clustering n'est pas significatif. Cet article remet donc clairement en cause l'application d'une méthode couramment utilisée dans le contexte de séries et de données temporelles.

Avant de contribuer au débat en cours, il est utile de préciser ce que recouvrent les termes de "clustering" et de "significatif". Le clustering recouvre un ensemble de techniques d'analyse de données utilisées pour classer de façon non supervisée des données, généralement vectorielles, dans des groupes. Le clustering est donc un moyen de partitionner un ensemble de données, la partition étant obtenue en regroupant ensemble des données qui partagent certaines caractéristiques, généralement une similarité ou une proximité selon un certain critère, généralement une mesure de distance (83). La quantification vectorielle, introduite au chapitre 2, est en fait une manière de faire du clustering. En effet, la quantification vectorielle conduit à une partition des données en clusters, ce qui est équivalent à un clustering. La principale différence entre quantification vectorielle et clustering réside dans le fait que l'obtention de données caractéristiques, telles que les prototypes, et la partition des données de départ constituent un but en soi en clustering, alors qu'il ne s'agit que d'un moyen de réduire le volume de données en minimisant la perte d'information dans le cas de la quantification vectorielle.

Dans (94), il est étudié dans quelle mesure le clustering de sous-séquences de séries temporelles, à savoir de régresseurs, est significatif. Les auteurs de (94) ont analysé le comportement des méthodes de clustering en comparant des distributions de prototypes obtenus

après clustering de régresseurs issus de séries temporelles distinctes. Ils ont constaté que les distributions de prototypes se ressemblent fortement, d'une série temporelle à l'autre, et ce bien que les séries temporelles soient distinctes (94). Cette observation a conduit les auteurs à conclure que les prototypes obtenus lors du clustering de régresseurs d'une série temporelle S_1 pouvaient être utilisés pour partitionner les régresseurs d'une autre série S_2, aussi fidèlement que s'ils avaient été obtenus directement à partir des régresseurs de cette seconde série S_2. En d'autres termes, quelle que soit la série temporelle considérée, les prototypes obtenus sont, dans leur ensemble, à peu près toujours les mêmes. Il n'y a donc dans ce cas aucun intérêt à appliquer des méthodes de clustering à des régresseurs ; le clustering n'est pas significatif (94).

La méthodologie introduite par (94) est reprise ici afin de comprendre ce qui a conduit les auteurs de cet article à la conclusion du caractère non significatif des méthodes de clustering. Le but est de démentir l'affirmation principale de cet article sur base de la même méthodologie : le clustering de régresseurs issus de séries temporelles est bel et bien significatif. Ce démenti permet alors de contribuer au débat en cours autour de cette publication, pour clarifier la situation. Les critères de comparaison, introduits dans (94), sont donc repris ici et appliqués à plusieurs séries temporelles. Grâce aux résultats expérimentaux, un rappel non négligeable sur le plan méthodologique est possible : souligner, sur base d'exemples concrets, l'importance d'une analyse des données préalable à la construction d'un modèle, en montrant par la pratique l'intérêt de construire des régresseurs en utilisant un délai. Une autre contribution est le critère géométrique proposé à la section 4.6 dans le cadre de la sélection du délai.

En pratique, il existe un certain nombre d'algorithmes de clustering, tels que le K-means (77), le K-medoid (77), les méthodes de clustering hiérarchique (77) ou encore les méthodes de quantification vectorielle (65; 77) comme le Competitive Learning ou le Self-Organizing Map présenté au chapitre 2, entre autres. Une classification des différentes méthodes de clustering est proposée dans (83) et un état de l'art des applications du clustering spécifiquement dans le cadre des séries temporelles est proposé dans (110). Dans le contexte temporel, les algorithmes de clustering sont par exemple utilisés pour obtenir une partition des données temporelles considérées, soit en comparant des séries entières entre elles, soit en comparant des sous-séquences d'une série temporelle (94), ou régresseurs. Le contenu de ce chapitre se situe dans le cadre des sous-séquences uniquement.

Dans la suite de ce chapitre, l'algorithme SOM, qui permet de quantifier vectoriellement des données, est utilisé pour étudier le caractère significatif du clustering. C'est donc avec cet outil, devenu familier au cours des pages, que l'importance du délai dans la construction des régresseurs va être mise en évidence.

4.4 Critères de comparaison pour distributions de prototypes

Pour pouvoir détecter dans quelle mesure des distributions de prototypes sont distinctes l'une de l'autre, il faut pouvoir comparer entre elles ces distributions. Conceptuellement, deux ensembles de prototypes issus de deux séries temporelles distinctes doivent être plus différents entre eux que ne le sont deux ensembles de prototypes obtenus à partir d'une même série, en appliquant à deux reprises un algorithme de clustering, comme le SOM, avec différentes initialisations. La mesure à la base des deux critères définis ci-dessous permet de formaliser cette idée.

Considérons deux séries temporelles S_1 et S_2, de longueurs respectives N_1 et N_2. Il n'est pas nécessaire que N_1 et N_2 soient identiques afin de pouvoir comparer les séries. Par contre, il est indispensable que les séries temporelles soient normalisées afin de ne pas fausser les comparaisons à cause, simplement, d'un facteur d'échelle. Dans la suite de cette section, les séries temporelles utilisées, qui sont toutes scalaires, sont toutes normalisées selon :

$$x_{norm}(t) = (x(t) - \mu_S)/\sigma_S, \tag{4.4}$$

où μ_S et σ_S désignent la moyenne et l'écart type des valeurs d'une série. Après normalisation de S_1 et de S_2, les régresseurs sont construits en tant que suite de valeurs normalisées $x_{norm}(t)$. On dispose alors de deux ensembles de régresseurs, notés $\mathcal{X}_1 = \{\mathbf{x}_i\}, p \leq i \leq N_1$ et $\mathcal{X}_2 = \{\mathbf{x}'_j\}, p \leq j \leq N_2$ contenant respectivement les régresseurs obtenus à partir des deux séries. Bien évidemment, la dimension p des régresseurs doit être la même pour les deux ensembles $\mathcal{X}_1$ et $\mathcal{X}_2$, sans quoi aucune comparaison entre ces vecteurs n'est possible.

En appliquant un algorithme de quantification, tel que le SOM, séparément sur $\mathcal{X}_1$ et $\mathcal{X}_2$, on obtient deux ensembles de prototypes $\mathcal{P}_1 = \{\bar{\mathbf{x}}_i\}, 1 \leq i \leq n$, et $\mathcal{P}_2 = \{\bar{\mathbf{x}}'_j\}, 1 \leq j \leq n$. Naturellement, le nombre n de prototypes dans chacun des deux ensembles $\mathcal{P}_1$ et $\mathcal{P}_2$ doit lui aussi être le même, sans quoi les comparaisons entre $\mathcal{P}_1$ et $\mathcal{P}_2$ ne seraient tout simplement pas correctes.

Pour comparer les distributions respectives de deux ensembles de prototypes $\mathcal{P}_1$ et $\mathcal{P}_2$, on définit une mesure $POS(.,.)$ (94; 146) qui se base sur la localisation des prototypes de $\mathcal{P}_1$ et $\mathcal{P}_2$ dans un espace de dimension p :

$$POS(\mathcal{P}_1, \mathcal{P}_2) = \sum_{i=1}^{n} \min_j \left(dist(\bar{\mathbf{x}}_i, \bar{\mathbf{x}}'_j) \right) \quad \text{avec} \quad 1 \leq j \leq n. \tag{4.5}$$

Cette mesure $POS(.,.)$ est donc définie comme la somme des distances entre chacun des prototypes de l'ensemble $\mathcal{P}_1$ et le prototype de $\mathcal{P}_2$ qui en est le plus proche, selon la distance $dist(.,.)$, qui est ici la distance euclidienne.

Notons que la mesure définie dans (4.5) n'est pas une relation un à un entre les prototypes des deux ensembles $\mathcal{P}_1$ and $\mathcal{P}_2$ (94; 146). En effet, certains prototypes de $\mathcal{P}_2$ peuvent être sélectionnés plusieurs fois, d'autres peuvent ne jamais être sélectionnés. Il est possible d'ajouter une contrainte à la mesure (4.5) pour forcer une relation un à un,

en ne sélectionnant qu'une et une seule fois chaque prototype de $\mathcal{P}_2$. Toutefois, il a été constaté que l'ajout d'une telle contrainte ne change pas grand chose, en pratique, quant aux conclusions sur le caractère significatif du clustering (94; 146).

Pour éviter tout biais dû à l'initialisation de l'algorithme, la détermination de la position des prototypes menant au clustering des régresseurs est répétée un certain nombre de fois K. On obtient dès lors deux ensembles d'ensembles de régresseurs pour les deux séries temporelles considérées S_1 et S_2, définis selon :

$$\mathfrak{P}_1 = \{\mathcal{P}_1^k \mid \mathcal{P}_1^k = \{\bar{\mathbf{x}}_i{}^k\}\}, \text{ avec } 1 \leq i \leq n \text{ et } 1 \leq k \leq K; \text{ et} \tag{4.6}$$

$$\mathfrak{P}_2 = \{\mathcal{P}_2^l \mid \mathcal{P}_2^l = \{\bar{\mathbf{x}}'_j{}^l\}\}, \text{ avec } 1 \leq j \leq n \text{ et } 1 \leq l \leq K. \tag{4.7}$$

Sur base de la mesure (4.5) et des ensembles $\mathfrak{P}_1$ et $\mathfrak{P}_2$, on peut maintenant définir deux critères qui permettent de caractériser d'une part la ressemblance d'une série temporelle avec elle même et d'autre part la différence entre deux séries temporelles. La différence entre ensembles de prototypes obtenus d'une même série temporelle S_1 au travers de plusieurs applications d'un algorithme de clustering est définie suivant :

$$within(\mathfrak{P}_1) = \sum_{k=1}^{K}\sum_{l=1}^{K} POS(\mathcal{P}_1^k, \mathcal{P}_1^l). \tag{4.8}$$

De manière similaire, la différence entre des ensembles de prototypes issus de deux séries temporelles distinctes S_1 et S_2, en utilisant à plusieurs reprises un algorithme de clustering, est définie suivant :

$$between(\mathfrak{P}_1, \mathfrak{P}_2) = \sum_{k=1}^{K}\sum_{l=1}^{K} POS(\mathcal{P}_1^k, \mathcal{P}_2^l). \tag{4.9}$$

Ces deux critères (4.8) et (4.9) peuvent être utilisés pour étudier le caractère significatif du clustering de séries temporelles. En effet, grâce à ces deux critères, il est possible de mesurer dans quelle mesure une distribution de prototypes est spécifique à une série temporelle ou si, au contraire, elle peut également caractériser une autre série temporelle distincte de la première. Intuitivement, rien ne ressemble plus à une distribution de prototypes obtenue sur base des régresseurs d'une série qu'une autre distribution de prototypes obtenue sur base des mêmes régresseurs, même avec une initialisation différente de l'algorithme utilisé. On peut donc s'attendre à ce que les valeurs du critère (4.8) soient relativement faibles. Par contre, pour des distributions de prototypes obtenues de deux ensembles de régresseurs issus de deux séries temporelles différentes, rien n'indique a priori si ces distributions de prototypes sont semblables ou significativement différentes. Intuitivement, si les séries sont différentes, on peut s'attendre à ce que les distributions de régresseurs, et donc in fine les distributions de prototypes, soient elles aussi différentes. Le but est donc d'observer si les valeurs du critère (4.9) sont relativement plus importantes que celles obtenues pour le critère (4.8). Dans ce cas de figure, les distributions de prototypes obtenues par clustering sont propres à chaque série temporelle. Le clustering obtenu est alors significatif. Dans le cas contraire, le clustering n'est pas en mesure de détecter les différences existantes entre des séries temporelles distinctes ; dans ce cas, il faut conclure au caractère non significatif du clustering.

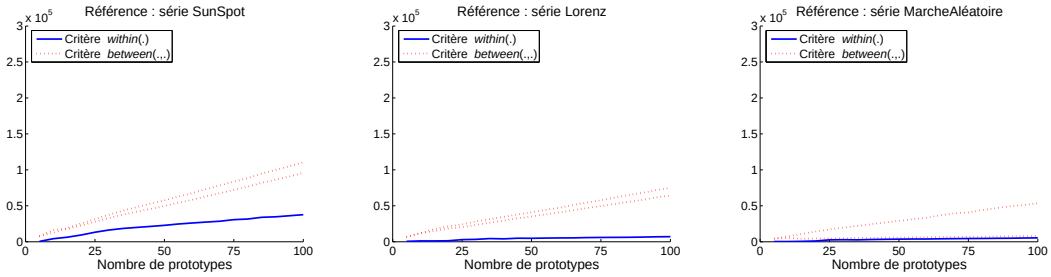

FIG. 4.1 – Comparaison des valeurs obtenues pour les critères $within(.)$ et $between(.,.)$ sur trois séries temporelles distinctes considérées chacune comme référence.

4.5 Comparaison de distributions de prototypes

Dans cette section, les critères (4.8) et (4.9) sont utilisés afin de comparer des ensembles de prototypes obtenus à partir de séries temporelles distinctes. Les résultats présentés dans cette section sont obtenus en utilisant l'algorithme SOM comme outil de clustering. Trois séries temporelles sont utilisées : une série réelle, la série SunSpot des taches solaires décrites au Chapitre 1 ; une série artificielle, générée à partir du système d'équations différentielles de Lorenz et une série en MarcheAléatoire, générée à partir d'un bruit blanc Gaussien, décrites en annexe A. Afin d'observer l'impact de la taille des clusters sur le caractère significatif du clustering de régresseurs, l'algorithme SOM est utilisé avec un nombre croissant de prototypes, variant de 5 à 100 par incrément de 5. Pour ces premiers résultats, la dimension p est fixée arbitrairement à 3, les régresseurs étant donc constitués comme des sous-séquences de la série (94), ce qui en trois dimensions correspond à :

$$\mathbf{x}_t = (x(t), x(t-1), x(t-2)), \text{ avec } 3 \leq t \leq N. \tag{4.10}$$

Les graphes de la Figure 4.1 présentent les résultats obtenus pour ces trois séries. Pour chaque graphe, une série temporelle est considérée comme série de référence, les deux autres séries temporelles étant alors comparées à cette série de référence. En d'autres termes, pour le graphe de gauche de la Figure 4.1, où la série SunSpot des taches solaires est considérée comme référence, il a été calculé $within(\mathfrak{P}(\text{Sunspot}))$, $between(\mathfrak{P}(\text{Sunspot}), \mathfrak{P}(\text{Lorenz}))$, et $between(\mathfrak{P}(\text{Sunspot}), \mathfrak{P}(\text{MarcheAléatoire}))$ où $\mathfrak{P}(S)$ désigne l'ensemble des ensembles de prototypes obtenus lors des $K = 50$ applications de l'algorithme SOM à la série S. Les graphes au centre et à droite de la Figure 4.1 présentent les résultats obtenus lorsque les séries Lorenz et MarcheAléatoire sont considérées comme série de référence.

Sur la Figure 4.1, on observe premièrement que le nombre de clusters a une influence sur la valeur des critères $within(.)$ et $between(.,.)$. En effet, en augmentant le nombre de prototypes utilisés, il devient naturellement plus facile de comparer des distributions de régresseurs entre elles, des différences locales apparaissant d'une distribution à l'autre au niveau de la position des prototypes. Intuitivement, ces différences locales sont dues à des caractéristiques propres à chaque série et qui ne peuvent être prises en compte qu'avec

un nombre suffisamment élevé de prototypes. Ces petites différences sont en effet perdues lors de la réduction d'information due à la quantification vectorielle lorsque le nombre de prototypes est trop faible.

Par ailleurs, comme on pouvait s'y attendre, on constate sur les graphes de la Figure 4.1 que les valeurs du critère $within(.)$ sont inférieures aux valeurs du critère $between(.,.)$, et ce quelle que soit la série temporelle considérée comme référence. Le point le plus important de cette figure est qu'il est déjà possible, dans la plupart des cas, de distinguer des distributions de régresseurs, ce qui contredit l'affirmation dans (94) et tend à prouver de manière qualitative que le clustering de régresseur est significatif. Il existe néanmoins un cas, sur la Figure 4.1 qui pourrait porter à confusion. Lorsque la série de référence est la série en MarcheAléatoire, une des deux courbes du critère $between(.,.)$ est très proche de la courbe du critère $within(.)$. Autrement dit, il existe des cas où, en appliquant un algorithme de quantification vectorielle à des ensembles de régresseurs obtenus à partir de deux séries temporelles différentes, les distributions de prototypes sont plus difficilement comparables entre elles. Dans ce type de situation, on pourrait alors être amené à conclure que le clustering n'est pas significatif : les prototypes obtenus ici pour la distribution des régresseurs de la série en MarcheAléatoire pourraient être utilisés pour caractériser la distribution des régresseurs de la série Lorenz. Or les séries temporelles sont nettement distinctes, si on s'en réfère aux figures de l'annexe A où ces séries sont illustrées graphiquement.

Pour comprendre ce qui semble être une contradiction, il est utile d'observer les distributions de régresseurs obtenus comme sous-séquences des différentes séries, c'est à dire construits en prenant une suite de valeurs successives de la série, séparées par un délai $\tau = 1$, pour le moment. La Figure 4.2 propose une représentation graphique de la distribution des régresseurs pour les trois séries SunSpot, Lorenz et MarcheAléatoire utilisée ci-dessus, en deux et trois dimensions. La difficulté de la comparaison des distributions de régresseurs apparaît sur cette Figure 4.2. On peut en effet comprendre que, dans certains cas, il est difficile de comparer ces distributions puisqu'elles occupent approximativement la même portion de l'espace, que ce soit en deux ou en trois dimensions. Pire, cette portion de l'espace, limitée autour de la diagonale principale, occupe une place relative qui diminue de plus en plus au fur et à mesure que la dimension de l'espace augmente. Il est donc possible que cette difficulté à comparer des distributions de régresseurs soit plus fréquente lorsque la taille p des régresseurs augmente. En l'état actuel, la seule chose que permettent de mesurer les critères $within(.)$ et $between(.,.)$, ce sont les largeurs respectives des distributions de régresseurs autour de la diagonale principale. Notons que ces largeurs sont tout de même suffisamment différentes pour permettre de distinguer, sauf dans quelques cas, les différences intrinsèques d'une série à l'autre, comme le montre la Figure 4.1. Pour éliminer les situations où il est relativement plus difficile de comparer des distributions de prototypes, il est nécessaire de forcer les distributions de régresseurs à occuper une portion de l'espace plus importante qu'une zone limitée autour de la diagonale principale.

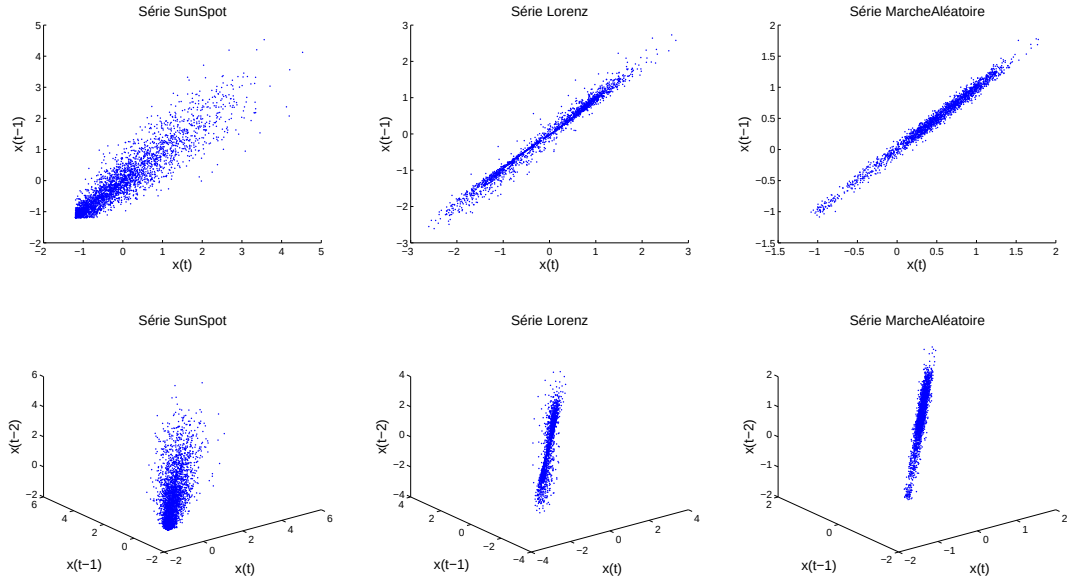

FIG. 4.2 – Distribution des régresseurs pour les trois séries considérées. En haut : régresseurs en deux dimensions ; en bas : régresseurs en trois dimensions.

4.6 Distance à la Diagonale

Les représentations des distributions de régresseurs de la Figure 4.2 constituent un moyen d'appréhender, ou d'analyser, des séries temporelles. L'analyse peut être complétée par la sélection d'un délai τ différent de 1. Sélectionner un délai différent de 1 permet de construire des régresseurs dont la distribution occupe une portion de l'espace plus large qu'une zone limitée autour de la diagonale principale.

Pour ce faire, il suffit de choisir un délai τ approprié, et de construire des régresseurs "dépliés" dans l'espace selon :

$$\mathbf{x}_t = (x(t), x(t-\tau), x(t-2*\tau)), \text{ avec } (2*\tau)+1 \leq t \leq N. \tag{4.11}$$

Notons que ce dépliage des régresseurs correspond en fait en la reconstruction de la série temporelle dans son espace de phase, dont il sera question plus en détails dans la section 5.1 du chapitre suivant. La sélection du délai par autocorrélation ou par information mutuelle y sera notamment présentée. Dans le cadre de ce chapitre, une méthode alternative et très intuitive est proposée pour sélectionner le délai, le critère de la Distance à la Diagonale. Ce critère est proposé pour mesurer l'importance du dépliage d'une distribution de régresseurs, afin que celle-ci soit aussi différente que possible de la portion de l'espace située autour de la diagonale principale de cet espace.

Formellement, la Distance à la Diagonale DD($\mathcal{P}$) d'un ensemble de régresseurs $\mathcal{P}$ se

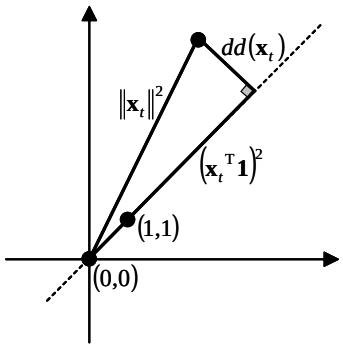

FIG. 4.3 – Représentation graphique du critère de calcul de la Distance à la Diagonale, dans le cas $p = 2$. La diagonale principale est ici représentée en pointillés.

définit selon :

$$\mathrm{DD}(\mathcal{P}) = \sum_{t=(2*\tau)+1}^{N} dd(\mathbf{x}_t) = \sum_{t=(2*\tau)+1}^{N} \|\mathbf{x}_t\|^2 - \left((\mathbf{x}_t)^{\mathrm{T}}\mathbf{1}\right)^2, \tag{4.12}$$

où $\mathbf{1}^{\mathrm{T}} = (1, 1, ..., 1)$ est un vecteur de dimension p, avec $^{\mathrm{T}}$ la notation habituelle pour la transposée. Ce critère $dd(\mathbf{x}_t)$ n'est en fait rien d'autre qu'une application de la formule générale de la distance d'un point $\mathbf{x}_t$ à une droite passant par deux points v_1 et v_2, en dimension p, et qui s'écrit :

$$dd(\mathbf{x}_t) = \frac{\|v_1 - \mathbf{x}_t\|^2\|v_2 - v_1\|^2 - \left((v_1 - \mathbf{x}_t)^{\mathrm{T}}(v_2 - v_1)\right)^2}{\|v_2 - v_1\|^2}. \tag{4.13}$$

Dans le cas du critère de Distance à la Diagonale, les vecteurs v_1 et v_2 sont choisis tels que ${v_1}^{\mathrm{T}} = (0, 0, ..., 0)$ et ${v_2}^{\mathrm{T}} = (1, 1, ..., 1)$, tous deux des vecteurs de dimension p.

La formule (4.12) est illustrée à la Figure 4.3, dans le cas $p = 2$, pour un point $\mathbf{x}_t$ donné. Comme l'illustre ce schéma, ce critère permet de mesurer à quel point un régresseur est éloigné de la diagonale principale.

Notons que l'idée de ce critère est comparable aux approches de (134) où il est défini une mesure de "force de la reconstruction du signal" en sommant les différences entre tous les régresseurs décalés dans le temps, pour différents délais, ou encore de (17) où les valeurs du "facteur de remplissage" obtenues pour différents délais sont comparées, le facteur de remplissage mesurant le volume moyen de parallélépipèdes en p dimensions dont les sommets sont des régresseurs choisis aléatoirement dans l'espace. Ici également, le but est de remplir le plus possible l'espace de phase, si ce n'est qu'on essaie ici que ce remplissage de l'espace de phase soit, autant que possible, différent de la diagonale principale de l'espace de phase considéré.

L'intérêt principal de ce critère de Distance à la Diagonale réside dans le fait qu'il est défini sur base de vecteurs représentant la position de points dans un espace de dimension p, ce qui lui permet d'être naturellement utilisé avec des régresseurs de dimension

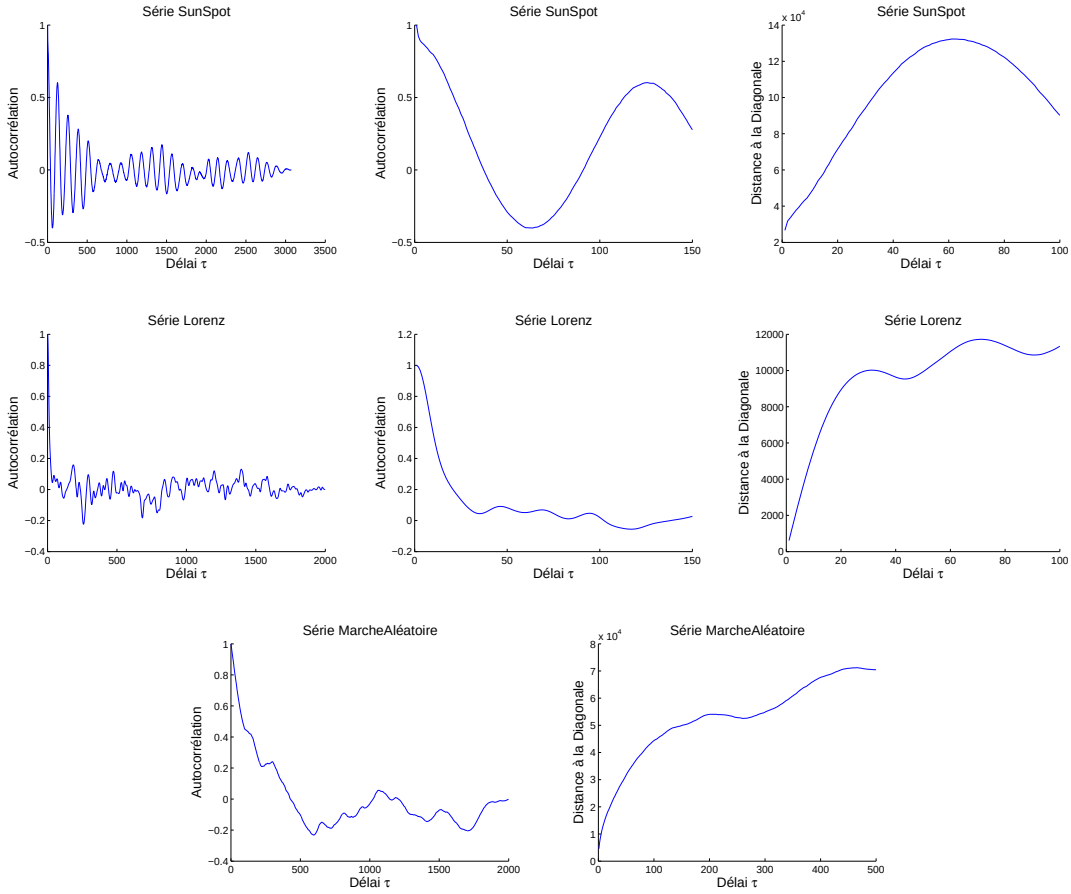

FIG. 4.4 – Comparaison des délais sélectionnés par autocorrélation et Distance à la Diagonale. Pour les séries SunSpot et Lorenz, le graphe au centre est un agrandissement du graphe d'autocorrélation pour les 150 premiers délais.

p quelconque. La Figure 4.4 présente les courbes obtenues par critère de Distance à la Diagonale, pour pouvoir sélectionner le délai pour les trois séries considérées ici. A fin de comparaison, les courbes obtenues par autocorrélation, dont il sera brièvement question à la section 5.2.1 du chapitre suivant, sont également proposées. Il est donc possible de comparer les valeurs de délai fournies par ces deux critères sur des exemples concrets.

En pratique, sur base de graphes tels que ceux présentés à la Figure 4.4, la sélection du délai se fait comme suit. Le but étant de maximiser la portion de l'espace occupée par la distribution des régresseurs, on cherche à sélectionner un maximum de la courbe. Sélectionner le délai tel que la valeur du critère de Distance à la Diagonale lui correspondant est le maximum global du graphe n'est cependant pas un bon choix. En effet, un tel délai est généralement trop éloigné dans le temps et conduit à créer des régresseurs dépliés particulièrement longs, ce qui n'a bien souvent plus beaucoup de sens par rapport à l'évolution physique du processus. Une bonne règle serait donc de considérer un délai ayant une valeur suffisamment haute du critère de Distance à la Diagonale. Cette règle est com-

Série temporelle	Critère	Délai
SunSpot	Autocorrélation	37
SunSpot	Distance à la Diagonale	55
Lorenz	Autocorrélation	35
Lorenz	Distance à la Diagonale	25
MarcheAléatoire	Autocorrélation	220
MarcheAléatoire	Distance à la Diagonale	100

TAB. 4.1 – Délais sélectionnés par autocorrélation et Distance à la Diagonale, pour les trois séries considérées ; régresseurs en deux dimensions.

Série temporelle	Délai
SunSpot	40
Lorenz	20
MarcheAléatoire	100

TAB. 4.2 – Délais sélectionnés par Distance à la Diagonale, pour des régresseurs en trois dimensions.

parable à l'approche classique avec le critère d'autocorrélation, où on sélectionne le premier délai dont la valeur est proche de zéro, soit un minimum local et non un minimum global. Les délais sélectionnés par autocorrélation et Distance à la Diagonale sont repris dans le Tableau 4.1. Les valeurs obtenues pour le critère de Distance à la Diagonale seul, dans le cas de régresseurs en trois dimensions, sont repris dans le Tableau 4.2. Les distributions de régresseurs construits en utilisant le délai fourni par Distance à la Diagonale, en deux et trois dimensions, sont illustrées respectivement dans les parties supérieure et inférieure de la Figure 4.5. Ces distributions permettent d'observer une reconstruction propre à chaque série temporelle. Ces reconstructions n'étaient pas visibles à la Figure 4.2, la distribution des régresseurs étant repliée sur elle-même. Par la suite, des régresseurs construits avec délai sont également appelés des régresseurs dépliés.

Un premier commentaire concerne le cas de la série MarcheAléatoire. Par définition d'une marche aléatoire, il n'y a aucune raison de prendre un délai supérieur à un dans un but de prédiction. Toutefois, le but ici n'est pas de prédire mais d'identifier des distributions de prototypes positionnés suivant les distributions de régresseurs propres à chaque série temporelle, à fin de comparaison. Il est donc choisi de prendre un délai $\tau > 1$ pour pouvoir déplier les régresseurs et voir apparaître une reconstruction structurée dans la distribution de ces régresseurs, même dans le cas de la série en MarcheAléatoire, comme l'illustrent les graphes de la Figure 4.5.

Un deuxième commentaire concerne le tableau 4.1. Dans ce tableau, on constate que le délai sélectionné par Distance à la Diagonale est différent de celui sélectionné par autocorrélation. Il y a toutefois une certaine cohérence entre les deux critères : le délai obtenu par Distance à la Diagonale se trouve à chaque fois dans une région du graphe de

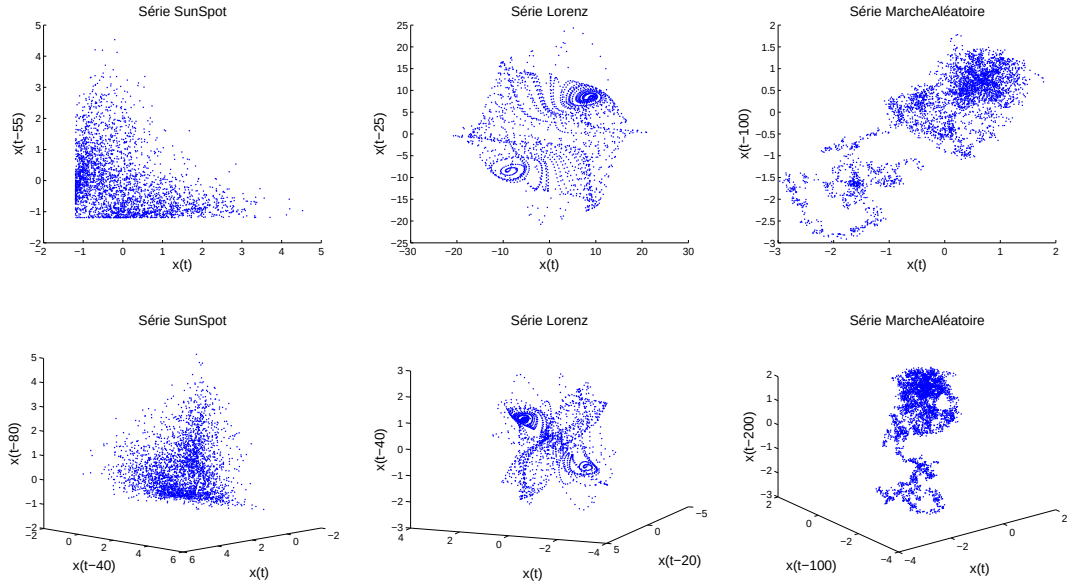

FIG. 4.5 – Distribution des régresseurs dépliés pour les trois séries considérées, en deux (haut) et trois (bas) dimensions. Les délais ont été sélectionnés par Distance à la Diagonale.

l'autocorrélation où celle-ci varie peu en valeur absolue. Prendre une valeur légèrement supérieure ou inférieure a donc un impact limité tant qu'on reste dans cette région du graphe où l'autocorrélation varie peu. Par ailleurs, il a été constaté dans certains cas que les délais sélectionnés par autocorrélation et Distance à la Diagonale peuvent être les mêmes, comme par exemple pour des séries générées suivant des modèles linéaires de type $AR(p)$ et $ARMA(p,q)$, ou encore pour des séries où les régresseurs sont déjà correctement dépliés en utilisant un délai $\tau = 1$ (146).

Enfin, en comparant les Tableaux 4.1 et 4.2, on constate une diminution du délai τ lorsque p augmente. Ce fait est très intéressant, et va à l'encontre des pratiques habituelles selon lesquelles on choisit un délai unique en deux dimensions et on prend ensuite des multiples de ce délai pour les autres dimensions. Le délai est toujours constant ici, et seuls des multiples sont considérés. Toutefois, le calcul de Distance à la Diagonale est effectué sur trois variables au lieu de deux pour le tableau 4.2 où on constate une diminution du délai choisi. On peut interpréter ce phénomène comme suit. Si l'information nécessaire pour déplier correctement une distribution en deux dimensions est disponible avec un délai $\tau = 25$, par exemple, il paraît naturel que cette information soit également présente avec un délai de $\tau = 25$ en trois dimensions et plus. Il n'est donc pas nécessaire de continuer à reculer dans le temps par pas de 25, mais d'obtenir un délai qui, multiplié par $p-1$ (le nombre de fois où il est utilisé pour construire le régresseur), approxime la valeur de $\tau = 25$ trouvée en deux dimensions. On peut donc considérer que la valeur sélectionnée quand $p = 2$ constitue une borne pour le cas de régresseurs de dimension $p > 2$. La sélection du délai en tenant compte de plus de deux variables est au coeur du chapitre suivant.

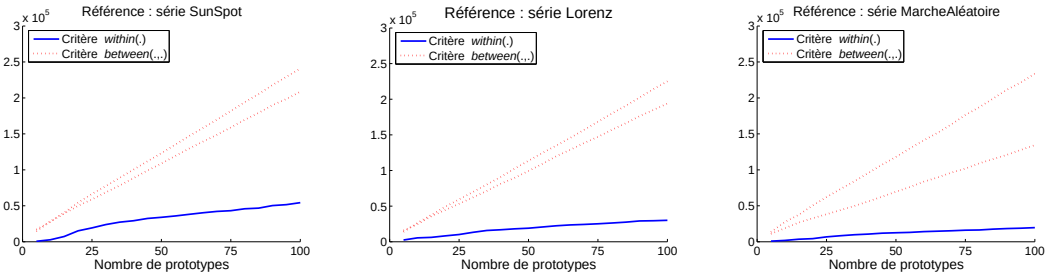

FIG. 4.6 – Comparaison des valeurs obtenues pour les critères $within(.)$ et $between(.,.)$. Par comparaison avec la figure 4.1, les graphes ci-dessus présentent les résultats obtenus lorsque les régresseurs en trois dimensions ont été dépliés en utilisant un délai fixe.

4.7 Sélection de délai et comparaison de distributions de prototypes

Dans cette section, les expériences de la section 4.5 sont à nouveau effectuées, mais avec des régresseurs dépliés en utilisant un délai sélectionné par Distance à la Diagonale.

En appliquant les critères $within(.)$ et $between(.,.)$ aux régresseurs dépliés en dimension $p = 3$, avec $K = 50$, on obtient les graphes de la Figure 4.6. En comparant avec la Figure 4.1, on constate une certaine amélioration, au moins sur le plan qualitatif, dans la comparaison entre les distributions de prototypes, et ce surtout dans le cas de la série MarcheAléatoire où une courbe $between(.,.)$ était très proche de la courbe $within(.)$.

Si la comparaison de distributions de prototypes pouvait être, dans certains cas, sensiblement plus difficile lorsque les régresseurs étaient construits avec un délai de un, comme dans le cas de la Figure 4.1, il n'y a plus aucune situation problématique dans le cas de régresseurs construits avec un délai sélectionné de manière appropriée, comme l'illustre la Figure 4.6. En effet, les courbes $within(.)$ et $between(.,.)$ sont à présents clairement distinctes, quelle que soit la série considérée comme référence. Il n'y a donc aucune raison pour remettre en cause le caractère significatif du clustering.

Il reste à observer que l'utilisation d'un délai correctement sélectionné permet également de distinguer plus facilement des distributions de régresseurs lorsque p augmente. Par construction, la distribution obtenue en utilisant la Distance à la Diagonale est aussi distincte que possible de la zone autour de la diagonale principale de l'espace. On s'attend donc à ce que ce dépliage facilite la comparaison de distributions de prototypes pour n'importe quelle dimension p, tout comme pour $p = 3$. La méthodologie de sélection du délai par Distance à la Diagonale est donc appliquée pour des régresseurs avec $p = 5$ et $p = 10$, pour les trois séries considérées ici. Les délais correspondants sont repris aux Tableaux 4.3 et 4.4. Les graphes des critères $within(.)$ et $between(.,.)$ sont repris aux Figures 4.7 et 4.8, avec une comparaison entre régresseurs dépliés et régresseurs avec délai $\tau = 1$.

Dans les Tableaux 4.3 et 4.4, on observe que la valeur du délai continue de diminuer lorsque que la dimension des régresseurs augmente, ce qui est cohérent avec l'interprétation

Série temporelle	Délai
SunSpot	20
Lorenz	20
MarcheAléatoire	50

TAB. 4.3 – Délais sélectionnés par Distance à la Diagonale, régresseurs de dimension $p = 5$.

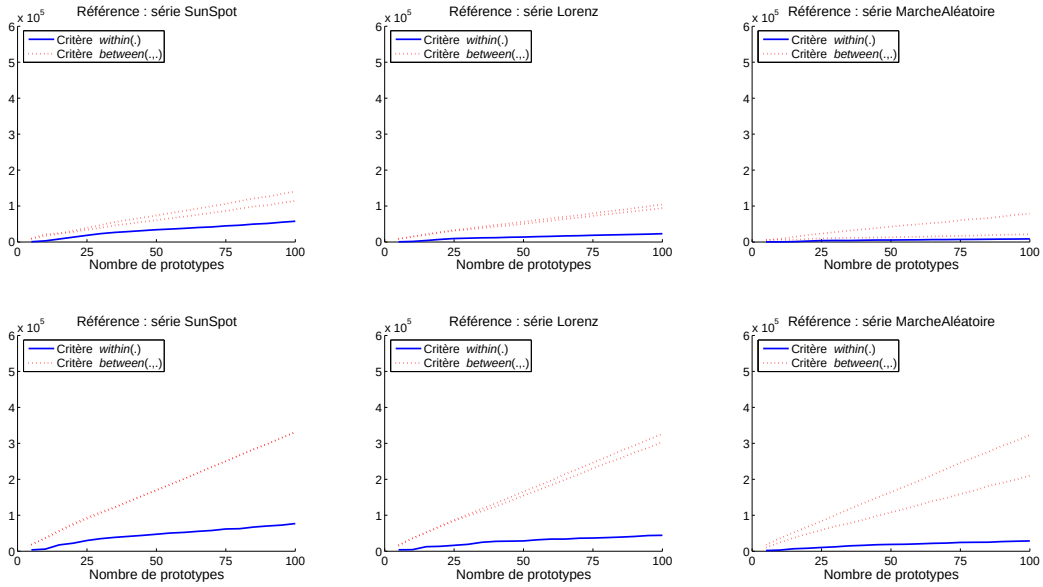

FIG. 4.7 – Comparaison des valeurs obtenues pour les critères *within*(.) et *between*(.,.), pour des régresseurs de dimension $p = 5$. Les graphes du haut illustrent le cas des régresseurs contigus, ceux du bas les régresseurs dépliés.

de la valeur du délai en deux dimensions, en tant que borne sur l'information disponible pour la construction des régresseurs. Nous reviendrons sur ce constat au Chapitre 5.

De plus, on constate sur les Figures 4.7 et 4.8 qu'il est de plus en plus aisé de distinguer des séries temporelles sur base de distributions de prototypes, même en diminuant le délai utilisé dans la construction des régresseurs quand p augmente. Il suffit de constater que la pente des graphes en 10 dimensions est nettement supérieure à celle en 5 dimensions, elle-même supérieure à celle des graphes en 3 dimensions vu le changement d'échelle. En effet, l'échelle des ordonnées des Figures 4.1 et 4.6 a été divisée par deux par rapport aux échelles des Figures 4.7 et 4.8, qui sont elles identiques, pour des raisons de lisibilité du cas de la série en MarcheAléatoire de la Figure 4.1.

On peut conclure cette section en rappelant que le clustering de régresseurs issus de séries temporelles est définitivement significatif, et ce d'autant plus lorsque les régresseurs ont été correctement dépliés, quelle que soit la dimension p des vecteurs régresseurs, et quelles que soient les séries considérées.

Série temporelle	Délai
SunSpot	10
Lorenz	10
MarcheAléatoire	20

TAB. 4.4 – Délais sélectionnés par Distance à la Diagonale, régresseurs de dimension $p = 10$.

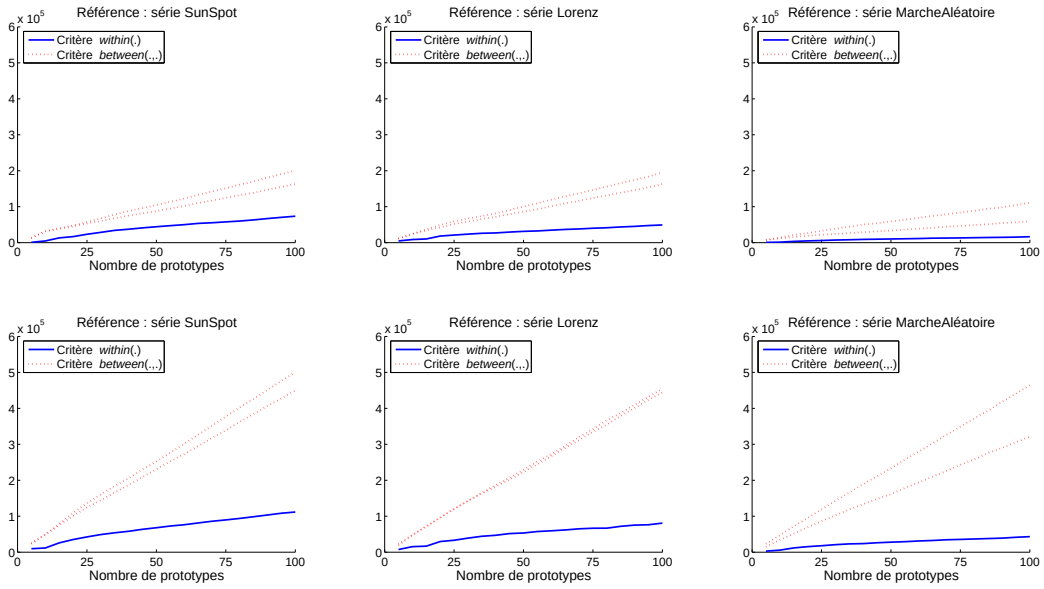

FIG. 4.8 – Comparaison des valeurs obtenues pour les critères $within(.)$ et $between(.,.)$, pour des régresseurs de dimension $p = 10$ contigus (haut) et dépliés (bas).

4.8 Caractère significatif du clustering : enseignements

L'étude du caractère significatif du clustering proposée à la section précédente et illustrée sur trois séries distinctes permet de conclure que le clustering de régresseurs, ou de sous-séquences d'une série temporelle, est significatif pour autant que les régresseurs aient été correctement dépliés dans l'espace de dimension p. Cette conclusion sur le caractère significatif du clustering, détaillée par ailleurs dans (61; 146), rejoint les conclusions de divers travaux.

Dans (161), le clustering de sous-séquences est plutôt vu comme "coriace" et non pas comme non significatif, et ce de par le caractère auto-similaire des séries temporelles : même en considérant des échelles de temps différentes, les séries temporelles restent semblables à elles-mêmes, ce que (161) appelle le caractère auto-similaire ou fractal des séries temporelles. (43) utilise une approche avec des outils de clustering à noyaux Gaussiens basés sur la densité de distributions des régresseurs pour obtenir un clustering significatif. (94) avance comme argument en faveur du caractère non significatif que les prototypes

produits par K-means sont généralement de forme sinusoïdale, quelle que soit la série temporelle considérée. (81) montre de manière théorique que ces prototypes sont effectivement de forme sinusoïdale, et propose d'utiliser des sous-séquences décalées de manière quelconque dans le temps pour éviter d'obtenir de tels prototypes sinusoïdaux. Mentionnons aussi (25), qui utilise également le K-means et la reconstruction des sous-séquences dans l'espace de phase pour montrer que le caractère non significatif du clustering peut provenir de l'utilisation de la distance Euclidienne dans les algorithmes, lorsque la dimension des vecteurs est élevée. Notons que le choix du SOM comme outil de clustering est justifié comme alternative au K-means (8). Au delà du clustering par SOM, des résultats comparables à ceux présentés dans ce chapitre ont été obtenus avec d'autres algorithmes de clustering, tels le K-means et le Competitive Learning, et d'autres séries temporelles (146). Quel que soit l'algorithme de clustering utilisé, les résultats obtenus conduisent toujours à la même conclusion : le clustering de régresseurs dépliés est significatif (146).

L'affirmation du caractère non significatif du clustering trouve sans doute son origine dans plusieurs erreurs méthodologiques, dont certaines sont passées en revue ici. La première réside dans les expériences réalisées pour mettre en évidence le caractère non significatif. Dans (94), les comparaisons de prototypes effectuées concernent des prototypes obtenus de deux séries temporelles uniquement, à savoir d'une part une série financière, et d'autre part une série de type marche aléatoire. La dynamique d'une série financière est relativement proche d'une marche aléatoire, rendant dès le départ difficile toute comparaison de distributions de régresseurs et de prototypes.

Par ailleurs, les valeurs des critères $within(.)$ et $between(.,.)$ ne sont pas représentées séparément dans (94), mais sous forme de rapport. L'idée est de dire que si le rapport de la valeur de $within(.)$ par rapport à la valeur de $between(.,.)$ est proche de zéro, alors le clustering est significatif. Par contre, plus la valeur de ce rapport est grande, moins le clustering de séries temporelles est significatif. Or, prendre le rapport de ces deux valeurs, revient à comparer les pentes des droites visibles sur les Figures 4.1, 4.6, 4.7 et 4.8. On voit nettement sur ces figures que la valeur du rapport ne peut en aucun cas s'approcher de zéro : il faudrait pour cela plusieurs ordres de grandeurs entre les valeurs des critères $within(.)$ et $between(.,.)$.

De plus, la définition des critères est basée sur des distributions de prototypes. Or, dans (94), le but est de trouver des prototypes spécifiques dans un but de clustering. Dans les expériences présentées, une réponse à une question de clustering (des prototypes spécifiques pour chaque série) est cherchée au travers de résultats obtenus par quantification vectorielle (une distribution de prototypes). Comparer des prototypes spécifiques pour observer dans quelle mesure ils sont distincts (et par conséquent obtenir des clusters eux aussi distincts) aurait donc été plus approprié. Cependant, même de telles comparaisons entre prototypes doivent être réalisées avec soin : on sait que la forme générale des prototypes est sinusoïdale lorsque les régresseurs ne sont pas décalés de manière quelconque dans le temps (81).

Enfin, il y a dans (94) un manque de toute analyse des séries utilisées. Un simple graphe de la distribution des régresseurs tant de la série financière que de la marche

aléatoire, comme ceux proposés dans la Figure 4.2, aurait probablement permis aux auteurs de prendre conscience du problème sous-jacent : les distributions étaient sans aucun doute fortement semblables. L'utilisation d'un délai aurait permis d'éviter dans tous les cas l'écueil dans lequel se sont retrouvés les auteurs, comme le montrent les résultats expérimentaux présentés dans ce chapitre.

Au-delà de la contribution apportée dans le débat en cours sur le caractère significatif ou non des méthodes de clustering appliquées à des régresseurs, un enseignement à retenir de cette étude est donc l'importance de l'analyse, qui est plus précisément illustrée ici par l'importance de la sélection du délai. Analyser des données est une règle de bonne pratique qui existe depuis longtemps, et les nombreuses méthodes d'analyse existantes montrent que cette question de l'analyse a déjà été abordée, à de nombreuses reprises, dans la littérature. Et pourtant, il semble opportun de rappeler cette nécessité de l'analyse, lorsqu'on se retrouve confronté à des affirmations telles que le caractère non significatif du clustering de régresseurs.

En plus de la contribution au débat en cours, une autre contribution est la définition du critère de Distance à la Diagonale, une heuristique géométrique proposée pour sélectionner le délai τ utilisé dans la construction des régresseurs. Grâce à sa définition dans le cas général, sur base de vecteurs, le critère de Distance à la Diagonale peut être naturellement utilisé avec un nombre quelconque de variables. Le chapitre suivant revient sur ce critère, en le comparant aux méthodes classiques de sélection de variables que sont l'autocorrélation et l'information mutuelle. Cependant, ces méthodes sont généralement définies sur base de deux variables uniquement. Il faut donc, dans un premier temps, généraliser les approches de sélection du délai par autocorrélation ou information mutuelle à plus de deux variables.

4.9 Résumé du chapitre

Dans ce chapitre, l'étape d'analyse de séries temporelles est abordée au moyen d'un exemple concret, la sélection du délai utilisé dans la construction des régresseurs.

L'importance de l'utilisation de ce délai est illustrée au travers de l'étude du caractère significatif de l'application des méthodes de clustering à des régresseurs. Cette étude a été l'occasion de contribuer à un débat en cours en affirmant que des régresseurs correctement dépliés, c'est à dire construits en utilisant un délai approprié, permettent de soutenir l'affirmation selon laquelle l'application des algorithmes de clustering à des régresseurs issus de séries temporelles est une démarche significative.

De plus, une méthodologie de sélection du délai utilisant le critère de Distance à la Diagonale a été proposée. Le critère de Distance à la Diagonale est une heuristique géométrique permettant de sélectionner un délai en remplissant l'espace de manière à ce que la distribution des régresseurs couvre une portion de l'espace la plus différente possible d'une zone concentrée autour de la diagonale principale de cet espace. L'étude de cette contribution est poursuivie dans le Chapitre 5, où il est également proposé une généralisation de deux méthodes classiques de sélection du délai, l'autocorrélation et l'information mutuelle.

Chapitre 5

Sélection du délai en haute dimension

Dans une première partie de ce chapitre, basée sur la théorie des systèmes dynamiques, les principaux concepts et résultats théoriques sont rappelés, de même que quelques grandeurs caractéristiques des séries temporelles. Les méthodes de sélection du délai utilisé dans la reconstruction de l'espace de phase sont ensuite décrites, à savoir l'autocorrélation et l'information mutuelle. Le besoin d'une généralisation de ces méthodes, où il est utilisé plus de deux variables, est introduit au travers d'un exemple concret illustrant la limite des critères définis en utilisant deux variables seulement. Une définition des critères d'autocorrélation et d'information mutuelle adaptée au cas général, en haute dimension, est ensuite proposée. L'utilisation en haute dimension du critère de Distance à la Diagonale, introduit au Chapitre 4, est étudiée. Une comparaison entre les approches basées sur l'information mutuelle et la Distance à la Diagonale est alors proposée, les observations étant discutées en rapport avec la théorie de la reconstruction de l'espace de phase. Cette discussion est suivie du résumé de ce chapitre.

5.1 Reconstruction de l'espace de phase : la théorie

La formulation générale des régresseurs proposée dans la définition (4.3), où un délai est utilisé, correspond en fait à la reconstruction d'une série temporelle dans son espace de phase, ou embedding dans la littérature anglaise (140; 165). Ce terme d'embedding peut se traduire par immersion qui, bien que peu utilisé même dans la littérature française, sera utilisé ici.

Quelques résultats théoriques rendent possible la reconstruction d'une série temporelle dans son espace de phase. Afin de comprendre ces théorèmes, quelques définitions sont tout d'abord nécessaires.

5.1.1 Définitions

Soient A et B deux ensembles et f une application qui va de A dans B. On dit que f est un difféomorphisme lorsque f une fonction bijective de classe $\mathcal{C}^1$.

On dit que l'application différentiable $f : A \to B$ est une immersion si $f(A) \subseteq B$ et si l'application $g : A \to f(A)$, défini par $g(a) = f(a)\ \forall a \in A$, est un difféomorphisme.

Une variété est l'ensemble des éléments d'un espace. Une variété de dimension d est une variété dont chacun des points peut être décrit pas d coordonnées. Lorsque l'entièreté des éléments d'un espace se trouve dans un sous-espace de dimension $d_i < d$, on dit alors que la dimension intrinsèque de l'ensemble des éléments est d_i.

Un système dynamique est un système dont l'évolution dans le temps est caractérisée dans un espace de phase, un espace de phase étant un espace dont chaque point décrit un état possible du système. On dit qu'une variété est invariante par rapport à un système dynamique lorsque tous les états du système dynamique sont dans la variété.

5.1.2 Théorèmes fondamentaux

Plusieurs résultats fournissent un cadre théorique à la construction de régresseurs qui conduisent à la reconstruction d'une série temporelle dans son espace de phase. Dans cette section, ces résultats sont abordés et leur portée discutée.

Le résultat le plus connu est sans doute le théorème de Takens (165), qui s'énonce comme suit :
Si A est une variété de dimension d_i dans $\mathbb{R}^k$ et que A est invariante par rapport à un système dynamique G, si $p > 2d_i$ et si F qui va de $\mathbb{R}^k$ vers $\mathbb{R}^p$ est une reconstruction du système basée sur une fonction de mesurage H et une période d'échantillonnage T, alors F est une bijection.

Notons que la reconstruction du système basée sur des valeurs obtenues d'une série temporelle selon une fonction de mesurage H et une période d'échantillonnage T correspond en fait simplement à la construction de régresseurs avec délai.

La portée de ce théorème est la suivante : Si on considère une série temporelle S pour laquelle on construit des régresseurs selon (4.3), avec $p > 2d_i$, alors la reconstruction dans l'espace de phase, de dimension p, de la série est en bijection avec le système dynamique initial, de dimension d_i, dont est mesurée la série. Ce résultat est d'une grande importance en pratique, lorsqu'on manipule effectivement des régresseurs de série temporelle : à partir du moment où le nombre de composantes utilisées dans un régresseur est suffisamment grande, on a la garantie que la reconstruction est équivalente au système dynamique initial. Notons que ce résultat ne garantit pas que la reconstruction et le système dynamique soient strictement identiques. On a uniquement l'assurance que la reconstruction et le système initial sont équivalents, grâce à une relation un à un pour tous les points de ces deux ensembles de points situés dans des espaces de dimensions différentes. Ce résultat est en fait suffisant lorsqu'on reconstruit l'espace de phase, que ce soit pour comparer des reconstructions, comme au chapitre 4, ou encore dans un but de prédiction, comme il en est question dans la partie de ce chapitre décrivant les résultats expérimentaux.

Le théorème de Takens, parfois attribué aussi à Mañé (115) dans la littérature, est en fait une particularisation d'un résultat démontré par Whitney dans un cadre beaucoup plus général (178). Ce résultat de Whitney énonce notamment que toute variété différentiable de dimension d_i peut être reconstruite dans un espace de dimension $2d_i+1$ au moyen d'une immersion. L'intérêt du théorème de Takens est de démontrer la validité de ce résultat dans un contexte temporel, lorsque l'immersion est une reconstruction de l'espace de phase par régresseurs avec délai. Cependant, ces deux résultats sont limités dans leur portée aux seuls cas où la dimension d_i est entière. Dans (140), des généralisations des résultats de Whitney et de Takens sont proposées, pour le cas où d_i est quelconque ; dès lors que d_i peut ne pas être entier, des structures de dimension d_i fractale peuvent aussi être considérées. Grâce aux résultats théoriques dans (140), on a donc la garantie en pratique que, pour autant que $p > 2d_i$, la reconstruction dans l'espace de phase de n'importe quelle série temporelle va être équivalente au système dynamique initial, que la dimension d_i soit entière ou non. Il faut noter qu'en pratique on ne dispose généralement pas de fonctions continues mais d'un nombre limité de valeurs, qui plus est entachées de bruit. Les résultats démontrés dans (140; 165) ont été étudiés expérimentalement dans (21) pour le cas de données bruitées. Il y est proposé un traitement approprié des séries temporelles bruitées tel que le bruit et les erreurs d'estimation sont pris en compte lors de la reconstruction de l'espace de phase. Cette étude montre de manière empirique que les résultats théoriques dans (140) restent valables en pratique, même avec un nombre fini de données, et même lorsque ces données sont bruitées.

Notons qu'une valeur est au centre de tous les théorèmes cités ici : la dimension intrinsèque d_i, de laquelle il est possible de déduire la dimension p de l'espace de phase. Quelques méthodes permettant d'estimer cette dimension intrinsèque d_i sont décrites ci-dessous. Par contre, il n'est pas dit comment on peut déterminer la valeur du délai τ à utiliser. Les résultats étant démontrés sans aucune mention à une quelconque valeur de τ, le plus simple est donc de sélectionner $\tau = 1$. Cependant, en pratique, on ne dispose que d'un nombre fini de valeurs de la série temporelle. On est donc bien loin du cas de fonctions continues, d'autant plus qu'il y a généralement une certaine quantité de bruit dans les valeurs mesurées. Le choix d'un délai τ adéquat peut donc améliorer sensiblement la reconstruction dans l'espace de phase, de même que l'estimation de plusieurs statistiques non linéaires telles que les exposants de Lyapunov ou l'intégrale de corrélation (88). Le choix du délai est au coeur de ce chapitre. Mais avant d'étudier cette question, quelques méthodes d'estimation de la dimension intrinsèque, permettant d'obtenir in fine la valeur de p, sont décrites.

5.1.3 Méthodes d'estimation de la dimension intrinsèque

En pratique, pour estimer de façon opérationnelle la dimension intrinsèque d_i, plusieurs méthodes peuvent être utilisées. Une description complète de ces méthodes dépasse le cadre de cette thèse et figure par ailleurs dans la plupart des ouvrages de référence dans

le domaine, comme par exemple (3; 88). Une description relativement brève de quelques unes de ces méthodes est proposée ici.

La dimension de corrélation (64) est une méthode dont la caractéristique principale est le calcul de l'intégrale de corrélation, calcul qui s'effectue comme suit. Considérons des points dans un espace de phase en dimension d. En chaque point, on positionne une boule de rayon fixé. On calcule ensuite toutes les distances entre tous les points, pris deux à deux, et on prend le rapport du nombre de distances inférieures au rayon des boules sur le nombre total de distances qui peut être calculé. Ce rapport est l'intégrale de corrélation pour une dimension donnée et un rayon donné. Le calcul de l'intégrale de corrélation est alors répété pour différentes dimensions d et pour des rayons de plus en plus petits. Les valeurs de la dimension de corrélation sont obtenues en prenant le rapport entre le logarithme de l'intégrale de corrélation et le logarithme du rayon des boules. Ces valeurs sont représentées sur un graphe log-log de l'intégrale de corrélation en fonction du rayon des boules, sur lequel on construit différentes courbes représentant chacune une dimension différente. L'estimation de la dimension intrinsèque d_i est choisie dans la partie du graphe où les différentes courbes saturent, c'est à dire dans la partie du graphe où les différentes courbes tendent à se superposer. La dimension intrinsèque est la pente des courbes situées dans cette partie du graphe et qui évoluent quasi linéairement. La procédure est résumée et appliquée de manière concrète dans (18) et étendue dans (56) ou (166), entre autres.

Une autre méthode couramment utilisée est le Box-Couting (3; 88). Dans cette méthode, un espace de dimension d est découpé en boîtes de sorte qu'on compte le nombre de boîtes qui contiennent au moins un élément de l'espace considéré. L'idée est alors de recommencer ces opérations de comptage en faisant décroître progressivement la taille des boîtes. En représentant sur un graphe log-log le nombre de boîtes non vides en fonction de l'inverse de la taille des boîtes, on obtient une série de valeurs qui peuvent être interpolées linéairement. La pente de la droite ainsi obtenue représente l'estimation de la dimension intrinsèque d_i. La procédure est décrite dans (42), entre autres.

La méthode des faux voisins (91) utilise une autre approche. Partant de régresseurs dans un espace de dimension d, la distribution des régresseurs est dépliée dans un espace de dimension $d+1$ en considérant des régresseurs de dimension $d+1$. L'idée est d'observer l'influence de l'ajout d'une composante aux régresseurs lorsqu'on calcule des distances entre points dans un espace de dimension d puis dans un espace de dimension $d+1$. En pratique, on observe les différences entre la distance entre un point et son $\mathrm{r}^{\text{ième}}$ voisin en dimension d et la distance entre ces mêmes points en dimension $d+1$. Si la différence entre les distances calculées en d et $d+1$ dimensions est importante, cela signifie que des points proches, sans doute à cause d'une projection dans un espace de dimension d trop petite, ont été éloignés l'un de l'autre dans un espace de dimension $d+1$. Ces faux voisins ont donc été séparés en augmentant d'une composante la taille des régresseurs, la distribution des régresseurs a été dépliée en passant de la dimension d à la dimension $d+1$. Le principe de la méthode est alors de poursuivre les ajouts de composantes, une à une, jusqu'à ce que les différences entre les distances en dimensions d et $d+1$ soient

inférieures à un seuil de 10, comme proposé par les auteurs. Dans ce cas, on considère qu'il n'y a plus de faux voisins et que la distribution des régresseurs est correctement dépliée, en d'autres termes que la reconstruction dans l'espace de phase est correcte. La dimension minimale permettant d'éviter le cas des faux voisins est alors choisie comme estimation de la dimension intrinsèque d_i. Une extension de la méthode résumée ici, ainsi qu'une application de cette méthode sont proposées dans (92).

La méthode de dimension minimum (19) peut être considérée comme une variante de la méthode des faux voisins. Des distances entre points en dimension d et en dimension $d+1$ sont également calculées, mais c'est le rapport entre les moyennes respectives de ces distances, en dimension d et en dimension $d+1$, qui est observé. L'estimation de la dimension intrinsèque d_i est alors obtenue pour la plus petite dimension d telle que le rapport des moyennes de distances entre points se stabilise, même lorsque la dimension continue d'augmenter.

Toutes ces méthodes permettent de déduire une valeur pour la dimension p de l'espace de phase, à partir de la connaissance de la dimension intrinsèque d_i. Le reste de ce chapitre est consacré à la sélection du délai τ. Les deux méthodes les plus couramment utilisées sont décrites en détails dans la section suivante.

5.2 Sélection du délai

Pour avoir une intuition de l'impact du délai sur la construction d'un régresseur, discutons le cas d'une série temporelle mesurée à une fréquence d'échantillonnage très élevée. Dans cette situation, le processus dispose de peu de temps pour évoluer de façon significative entre deux mesures. Par conséquent, des valeurs successives de la série sont fortement corrélées entre elles : l'information contenue dans une valeur est fortement semblable à celle contenue dans les valeurs précédent de même que suivante de la série. En utilisant un délai lors de la construction des régresseurs d'une série de ce type, il est possible de sélectionner des valeurs avec un contenu informatif utile en vue de la construction d'un modèle, par exemple.

En réalité, choisir un délai doit faire l'objet d'un compromis. Si le délai choisi est trop court, les valeurs sélectionnées sont trop fortement corrélées entre elles, elles sont redondantes (21; 114). A l'opposé, si le délai est trop grand, les valeurs sélectionnées deviennent faiblement corrélées avec la sortie, jusqu'à être indépendantes de cette sortie dans le pire des cas. Le contenu informatif de ces variables est alors très faible, elles n'ont plus beaucoup de pertinence (voire aucune) par rapport à la sortie (21; 114), qu'elles soient ou non redondantes entre elles. Le but est donc de choisir un délai qui sépare des valeurs telles qu'elles sont relativement indépendantes entre elles, tout en étant encore suffisamment corrélées avec la sortie. La question est donc de savoir comment choisir un tel délai afin que les valeurs de la série sélectionnées dans le régresseur apportent chacune une information utile qui n'est pas fournie par les autres composantes du régresseur.

En pratique, deux méthodologies sont souvent utilisées pour sélectionner le délai τ. Ces méthodes sont deux approches basées d'une part sur la fonction d'autocorrélation

et d'autre part sur l'information mutuelle. On peut voir ces deux méthodes comme une version linéaire et une version non linéaire d'une même approche : dans les deux cas, un délai unique est sélectionné et le régresseur est ensuite construit en utilisant $p-1$ multiples de ce délai. Mentionnons qu'il existe également des approches basées sur la transformée de Fourrier afin de sélectionner le délai, comme résumé dans (88) entre autres, mais ces méthodes ne sont pas abordées ici.

Notons enfin que, dans cette section tout comme dans tout le reste de ce chapitre, toutes les séries temporelles utilisées à fin d'illustration ont été pré-traitées comme suit : les séries ont été stationnarisées, le cas échéant désaisonnalisées et, dans tous les cas, normalisées.

5.2.1 Sélection linéaire du délai par autocorrélation

De manière classique, la fonction d'autocorrélation $C(\tau)$, pour un délai τ fixé, d'une série temporelle scalaire $S = \{x(t) \mid 1 \leq t \leq N\}$ est définie selon :

$$C(\tau) = \langle x(t)x(t-\tau)\rangle = \sum_{t=\tau}^{N-1} x(t)x(t-\tau) \text{ avec } 0 \leq \tau \leq M, \tag{5.1}$$

où $M \leq N-1$ est le délai maximum. Suivant sa définition, l'autocorrélation est une mesure linéaire de la dépendance entre deux variables séparées par un délai τ. En pratique, le calcul de l'autocorrélation est répété pour une suite de valeurs croissantes de τ. Le graphe de la valeur d'autocorrélation en fonction du délai τ est ensuite dessiné.

Dans la littérature, plusieurs heuristiques ont été proposées pour sélectionner le délai sur base du graphe de la fonction d'autocorrélation : lorsque la valeur de la fonction d'autocorrélation correspondante est proche de zéro, lorsque la pente de la fonction d'autocorrélation est décroissante selon $1/e$, lorsque la courbe d'autocorrélation présente un point d'inflexion, etc (134; 157). Par la suite, le critère retenu est de sélectionner le premier délai tel que la valeur de la fonction d'autocorrélation correspondante est proche de zéro. Ce choix a l'avantage de sélectionner un délai entre deux variables tel que ces variables sont peu corrélées entre elles. La combinaison de ces deux variables fournit donc un contenu informatif pertinent. Cette approche est d'ailleurs la plus couramment utilisée dans la littérature.

Un graphe de la fonction d'autocorrélation pour une série temporelle générée artificiellement à partir du système d'équations différentielles de Lorenz (5; 88; 89), décrite dans l'annexe A, est proposé à la Figure 5.1. On voit sur cette figure que la valeur de la fonction d'autocorrélation décroît au fur et à mesure que le délai augmente, ce qui illustre la redondance entre deux valeurs proches dans le temps ($1 \leq \tau \leq 5$), et le manque de pertinence entre deux valeurs trop éloignées ($\tau \geq 60$). Un délai réalisant le compromis entre redondance et pertinence est situé dans l'intervalle entre $\tau = 20$ et $\tau = 40$. En sélectionnant le délai dont la valeur d'autocorrélation est la plus proche de zéro, on sélectionne $\tau = 35$, ce qui correspond au premier minimum local du graphe de la Figure 5.1. La reconstruction obtenue dans l'espace de phase en deux dimensions est présentée à la Figure 5.2.

FIG. 5.1 – Graphe de la fonction d'autocorrélation pour une série générée selon le système d'équations de Lorenz, pour les 100 premiers délais.

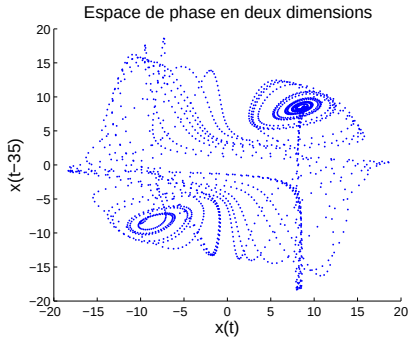

FIG. 5.2 – Représentation dans l'espace de phase en deux dimensions de la reconstruction de la série Lorenz, pour un délai $\tau = 35$ sélectionné par autocorrélation.

Une approche plus générale mesurant les dépendances non linéaires entre variables est l'information mutuelle.

5.2.2 Sélection non linéaire du délai par information mutuelle

L'information mutuelle est une mesure non paramétrique de la dépendance entre variables, capable de détecter une relation quelconque entre les variables considérées, lorsqu'il en existe une. L'information mutuelle a été introduite dans le cadre de la théorie de l'information de Shannon.

L'information mutuelle IM(.,.) entre deux variables $x(t)$ et $x(t-\tau)$ peut être définie selon (32) :

$$\mathrm{IM}(x(t), x(t-\tau)) = \int \mathrm{P}\left[x(t), x(t-\tau)\right] \log \left(\frac{\mathrm{P}\left[x(t), x(t-\tau)\right]}{\mathrm{P}\left[x(t)\right] \mathrm{P}\left[x(t-\tau)\right]} \right) \mathrm{d}x(t)\ \mathrm{d}x(t-\tau), \quad (5.2)$$

où P[.] désigne la probabilité. Conceptuellement, ce critère permet de mesurer la quantité d'information à propos de $x(t)$ qui peut être obtenue grâce à la connaissance de $x(t-\tau)$.

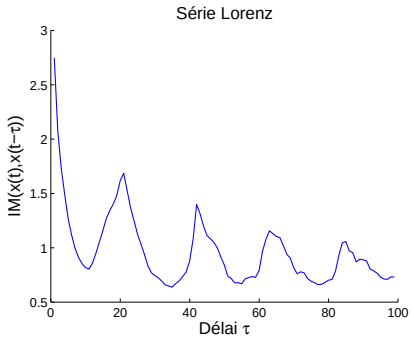

FIG. 5.3 – Graphe de l'information mutuelle en fonction du délai pour les 100 premiers délais, série Lorenz.

Il a été proposé d'utiliser l'information mutuelle en lieu et place de la fonction d'autocorrélation, comme critère non linéaire de sélection du délai (58). Ici encore, le but est de sélectionner une variable suffisamment indépendante des autres, mais un minimum corrélée avec la sortie, afin d'apporter une information utile par exemple dans la construction du régresseur. Cette variable ne doit cependant pas être totalement indépendante de la sortie car elle n'apporterait dans ce cas aucune information utile (3). Il est donc question, une fois encore, d'un compromis entre redondance et pertinence.

Pour ce critère également, l'information mutuelle est calculée entre deux variables $x(t)$ et $x(t-\tau)$ pour des valeurs de τ croissantes. Le graphe de l'information mutuelle en fonction du délai τ est alors dessiné. Le critère de sélection du délai proposé est de prendre le délai qui correspond au premier minimum local de l'information mutuelle (3; 58).

Un graphe de l'information mutuelle pour la série Lorenz utilisée dans la section précédente est proposé à la Figure 5.3, pour des délais allant de 1 à 100. Sur cette figure, on constate également que la valeur de l'information mutuelle décroît au fur et à mesure que le délai augmente, illustrant à nouveau le compromis à réaliser entre redondance ($1 \leq \tau \leq 5$) et manque de pertinence ($\tau \geq 80$). Selon le critère de (58), le délai réalisant un bon compromis est ici $\tau = 11$. La reconstruction obtenue dans l'espace de phase est présentée à la Figure 5.4. Par comparaison avec la Figure 5.2, on constate que la reconstruction de la série temporelle est plus facilement visible dans le cas de la reconstruction avec délai choisi par information mutuelle, le délai choisi permettant de construire des régresseurs plus compacts dans le temps ; ce constat est cohérent avec d'autres observations reprises dans (58).

5.2.3 Vers une généralisation de l'autocorrélation et de l'information mutuelle

Il a été montré dans (3; 58) que la sélection du délai par information mutuelle est plus efficace que son alternative linéaire par autocorrélation. En effet, le délai sélectionné par information mutuelle est généralement plus petit que celui obtenu par autocorrélation.

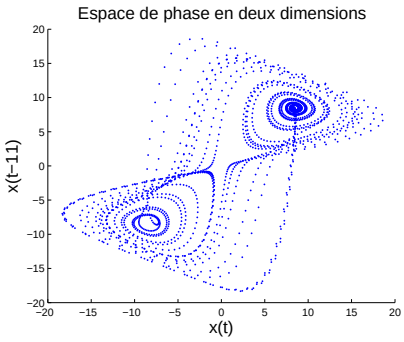

FIG. 5.4 – Représentation dans l'espace de phase en deux dimensions de la reconstruction de la série Lorenz, pour un délai $\tau = 11$ sélectionné par information mutuelle.

Ce fait permet d'obtenir des régresseurs avec un bon contenu informatif mais aussi, et surtout, plus compacts dans le temps que s'ils avaient été construits en utilisant un délai sélectionné par autocorrélation. L'exemple de la série temporelle Lorenz et la comparaison des reconstructions obtenues dans des espaces de phase en deux dimensions, proposées aux Figures 5.2 et 5.4 tend à confirmer cette affirmation de (58). Le délai sélectionné par information mutuelle est en effet ici trois fois plus court que celui fourni par la fonction d'autocorrélation, un résultat cohérent avec la littérature (3).

Dans (58), il est recommandé d'observer l'information mutuelle en haute dimension, à savoir lorsque $p > 2$, et d'observer le gain en information lors de l'ajout dans le régresseur d'une variable supplémentaire. Néanmoins, (88) affirme que le calcul de l'information mutuelle dans le cas $d = 2$ est suffisant pour sélectionner le délai en haute dimension. Par ailleurs, (3) propose que la sélection d'un délai dans le cas $p > 2$ n'est utile que lorsque l'application nécessite réellement une optimisation poussée de la sélection des éléments du régresseur, vu les temps calculs demandés. Bien qu'il existe quelques estimateurs d'information mutuelle adaptés pour le cas $p > 2$, les quelques recommandations ci-dessus n'ont jamais incité à appliquer l'un ou l'autre de ces estimateurs à la sélection de délai(s) entre plus de deux variables.

Or, l'hypothèse que l'autocorrélation (ou l'information mutuelle) entre deux variables séparées par un intervalle de temps équivalent au délai τ permet de sélectionner automatiquement le délai entre p variables simplement en prenant des multiples de ce délai τ est relativement forte. En toute généralité, si on se situe dans un espace de dimension p, il semble normal de choisir le délai directement en dimension p. Par ailleurs, utiliser un délai unique constitue sans doute une contrainte artificielle à la construction des régresseurs. Un régresseur devrait donc pouvoir être construit selon :

$$\mathbf{x}_t = (x(t), x(t-\tau_1), x(t-\tau_2), \ldots, x(t-\tau_{p-1}))\,, \tag{5.3}$$

avec $0 < \tau_1 < \tau_2 < \ldots < \tau_{p-1}$.

Dans la suite de ce chapitre, il est étudié dans quelle mesure un régresseur construit en utilisant un (ou des) délais sélectionné(s) directement dans un espace de dimension p, selon (5.3) peut avoir un impact sur la qualité de la prédiction par rapport à un régresseur construit en utilisant un délai unique selon (4.3). Pour ce faire, ce chapitre montre les limites des critères dans le cas en deux dimensions et propose une généralisation des critères d'autocorrélation et d'information mutuelle à plus de deux variables. L'utilisation du critère de Distance à la Diagonale, introduit au chapitre précédent, est proposée comme alternative dans le cas de la sélection de délai(s) dans le cas $p > 2$.

5.3 Autocorrélation et information mutuelle à deux variables : les limites

La méthodologie de sélection d'un délai τ, rappelée aux sections précédentes, peut se résumer comme suit. Des critères, l'autocorrélation ou l'information mutuelle, sont calculés pour différentes valeurs du délai τ. Les valeurs obtenues sont reportées sur un graphe, à partir duquel un délai unique est sélectionné. Les régresseurs sont ensuite construits comme des suites de valeurs décalées dans le temps suivant des multiples du délai τ sélectionné.

Que ce soit dans le cas de l'autocorrélation (5.1) ou de l'information mutuelle (5.2), les critères ont été présentés suivant leur définition classique, où il n'est utilisé que deux variables seulement. Or, en pratique, les régresseurs comptent régulièrement plus de deux composantes. Dans la suite de ce chapitre, l'approche habituelle consistant à utiliser des multiples du délai sélectionné entre deux variables seulement est remise en cause. En effet, cette approche contraint artificiellement la construction des régresseurs.

Considérons le cas d'un régresseur en trois dimensions, construit en utilisant un délai τ unique sélectionné en deux dimensions :

$$\mathbf{x}_t = (x(t), x(t-\tau), x(t-2\tau))\,. \tag{5.4}$$

Que ce soit par autocorrélation ou par information mutuelle, utiliser un délai permet de maximiser le contenu informatif repris dans les deux variables $x(t)$ et $x(t-\tau)$. De même, l'écart étant identique, le contenu d'information contenu dans les variables $x(t-\tau)$ et $x(t-2\tau)$ est lui aussi maximal. Par contre, il n'y a aucune garantie que le contenu d'information contenu dans $x(t)$ et $x(t-2\tau)$ soit maximal. Au contraire, il peut même être très nettement sous optimal, par exemple lorsque le critère utilisé évolue de façon cyclique en fonction du délai. Il semble dès lors plus intéressant, d'un point de vue théorique, soit de sélectionner un délai unique mais en tenant compte des p variables, comme par exemple dans (5.4) pour $p = 3$, soit de reconstruire les régresseurs selon la relation (5.3). Dans le premier cas, certes le délai est toujours constant, mais la sélection est réalisée directement dans l'espace de dimension p, et donc en fonction de toutes les variables servant à la reconstruction. Dans le deuxième cas, la reconstruction est, potentiellement, encore plus intéressante car celle-ci n'est plus contrainte par l'obligation de conserver un délai constant

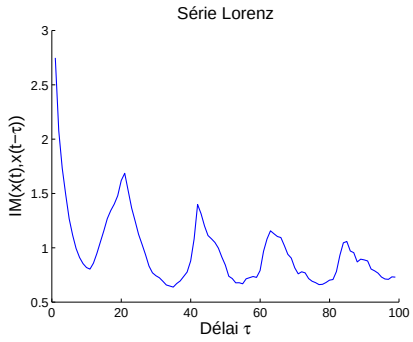

FIG. 5.5 – Graphe de l'information mutuelle en fonction du délai pour les 100 premiers délais, série Lorenz.

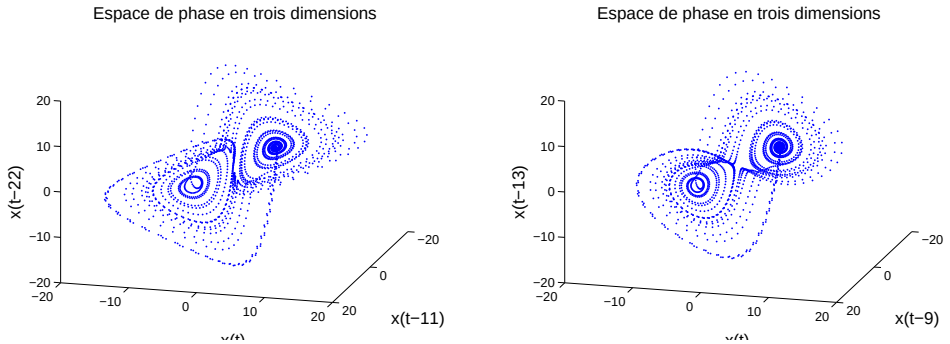

FIG. 5.6 – Reconstruction de la série Lorenz dans l'espace de phase en trois dimensions. A gauche : $\tau_1 = \tau_2 = \tau = 11$; à droite $\tau_1 = 9, \tau_2 = 13$.

entre les différentes composantes du régresseurs. Par la suite, la sélection du délai dans un espace de dimension $p > 2$ est appelée par abus de langage la sélection en haute dimension.

Pour illustrer l'intérêt d'une reconstruction avec délai sélectionné en dimension $p = 3$, considérons le cas de la sélection du délai par information mutuelle, appliqué à la série Lorenz. La Figure 5.3 est reproduite à la Figure 5.5. La reconstruction est proposée selon l'approche avec délai unique (4.3) et selon l'approche avec plusieurs délais (5.3).

En reconstruisant les régresseurs en trois dimensions de cette série, l'approche classique consiste à prendre $\mathbf{x}_t = (x(t), x(t-11), x(t-22))$, après avoir sélectionné $\tau = 11$ sur base de la Figure 5.5. Ce choix conduit à une reconstruction dans l'espace de phase en trois dimensions telle qu'illustrée dans la partie gauche de la Figure 5.6. En sélectionnant deux délais différents, tous deux proches du minimum mais non multiples, comme par exemple $\tau_1 = 9$ et $\tau_2 = 13$, on obtient une autre reconstruction, illustrée dans la partie droite de la Figure 5.6.

A première vue, les reconstructions sont fort semblables. Cependant, lorsqu'on se déplace simplement dans l'espace de phase en trois dimensions pour observer la reconstruction sous un autre angle, ce qui revient à effectuer une rotation de l'espace de phase sur

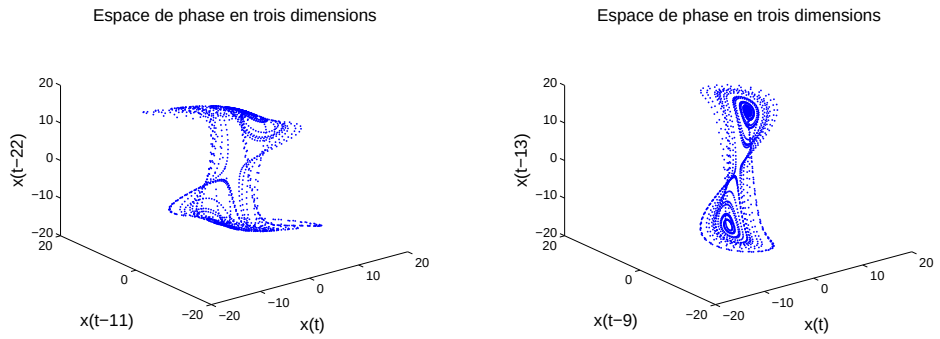

FIG. 5.7 – Reconstruction de la série Lorenz dans l'espace phase en trois dimensions. Par comparaison avec la Figure 5.6, l'espace de phase a subi une rotation.

lui-même, on constate une différence très nette dans la représentation, comme le suggère la Figure 5.7. En effet, autant la reconstruction est conservée dans les deux cas, autant on peut constater une différence assez nette dans la qualité du dépliage dans le cas de délais multiples du délai unique $\tau = 11$ choisi entre deux variables seulement : la reconstruction est dépliée de manière excessive, elle en arrive même à se refermer sur elle-même. Ce phénomène est surtout visible aux extrémités de la reconstruction, mais il va en s'accentuant pour l'ensemble de la reconstruction en utilisant un délai supérieur à $\tau = 11$. La Figure 5.7 illustre de manière assez nette le fait que le contenu informatif entre $x(t)$ et $x(t-2\tau)$ n'a pas été pris en compte lors de la sélection du délai unique τ sélectionné avec deux variables seulement.

Si on revient quelques instants à la Figure 5.5, on peut par ailleurs constater que, dans le cas de la série Lorenz, l'information mutuelle entre $x(t)$ et $x(t-22)$ est plus proche d'un maximum local que d'un minimum local (on se situe en fait juste après le premier pic, sur la courbe). L'approche habituelle avec délai unique sélectionné en deux dimensions est donc prise en défaut lorsqu'on construit des régresseurs dans un espace de dimension supérieure à deux.

Par la suite, on propose donc de généraliser les méthodes de sélection de délai pour pouvoir sélectionner soit un délai unique mais directement en haute dimension, donc dans le cas $p > 2$, soit plusieurs délais différents, mais dans ce cas aussi directement en haute dimension. Les méthodes proposées sont : une généralisation du critère linéaire d'autocorrélation pour un ordre supérieur quelconque, une généralisation du critère d'information mutuelle, en utilisant un estimateur adapté au cas des vecteurs de dimension $p > 2$, et le critère géométrique de Distance à la Diagonale introduit au chapitre 4.

5.4 Délai par autocorrélation d'ordre supérieur

La fonction d'autocorrélation (5.1) est généralisée ici à plus de deux variables, de façon immédiate, suivant :

$$\begin{aligned} C_{(p-1)}(\tau) &= \langle x(t)x(t-\tau)\dots x(t-(p-1)*\tau)\rangle \\ &= \sum_{t=(p-1)*\tau}^{N-1} x(t)x(t-\tau)\dots x(t-(p-1)*\tau), \end{aligned} \tag{5.5}$$

avec $0 \leq \tau \leq M$, où $M \leq N-p-1$ est le délai maximum. Cette relation est décrite dans (59; 143) pour l'ordre 3 (soit $p=3$), ou dans (127) pour l'ordre 4.

Notons que ce critère peut être utilisé tel quel pour sélectionner un délai τ unique. Mais il peut également servir, de manière encore plus générale, à sélectionner $p-1$ délais distincts $0 < \tau_1 < \tau_2 < \ldots < \tau_{p-1}$:

$$C_{(p-1)}(\tau_1, \tau_2, \ldots, \tau_{p-1}) = \langle x(t)x(t-\tau_1)\dots x(t-\tau_{p-1})\rangle. \tag{5.6}$$

L'application du critère avec délai unique, selon (5.5), est illustrée à la Figure 5.8 dans le cas de la série Rossler, série obtenue en intégrant les équations différentielles du système d'équations de Rossler et présentée en annexe A. La Figure 5.8 présente les courbes obtenues pour des ordres allant de 2 à 10. Bien évidemment, le graphe en haut à gauche correspond à l'autocorrélation d'ordre 2 et est identique au graphe obtenu par l'autocorrélation classique (5.1).

Par ailleurs, le critère d'autocorrélation généralisée confirme les observations du chapitre 4 lors de l'utilisation de la Distance à la Diagonale pour des régresseurs de taille p croissante. En effet, on constate ici aussi que la valeur du délai τ sélectionné directement en haute dimension diminue à nouveau lorsque la dimension p augmente. De $\tau = 27$ en dimension 2, on passe à $\tau = 10$ en dimension 3, pour finir à $\tau = 4$ en dimension 10. Cette diminution de la valeur du délai sélectionné en haute dimension est d'autant plus intéressante qu'elle se distingue de l'usage classique qui considère qu'il faut sélectionner des multiples d'un délai unique et constant.

Malheureusement, l'exploitation des critères (5.5) et (5.6) reste limitée. En effet, lorsque la dimension p des régresseurs augmente (lorsque l'ordre de l'autocorrélation augmente), les courbes des graphes ont tendance à devenir de plus en plus plates, ce qui est tout à fait normal. En effet, pour $p=4$, par exemple, la corrélation entre les variables $x(t)$, $x(t-15)$, $x(t-30)$ et $x(t-45)$ est beaucoup plus faible que celle calculée entre les variables $x(t)$, $x(t-5)$, $x(t-10)$ et $x(t-15)$: la dépendance entre variables étant de moins en moins forte avec le temps, il est tout à fait normal que l'information contenue dans le premier groupe de variables ci-dessus soit moins significative que pour le second groupe. Il est donc tout à fait normal d'observer à la Figure 5.8, à partir d'une autocorrélation d'ordre 6, des variables qui ont l'air indépendantes pour un délai $\tau \geq 7$. Toutefois, les variables représentent des valeurs issues d'une série temporelle. Il existe donc un lien entre elles : elles sont issues du même processus. Il devrait donc encore y avoir l'une ou l'autre

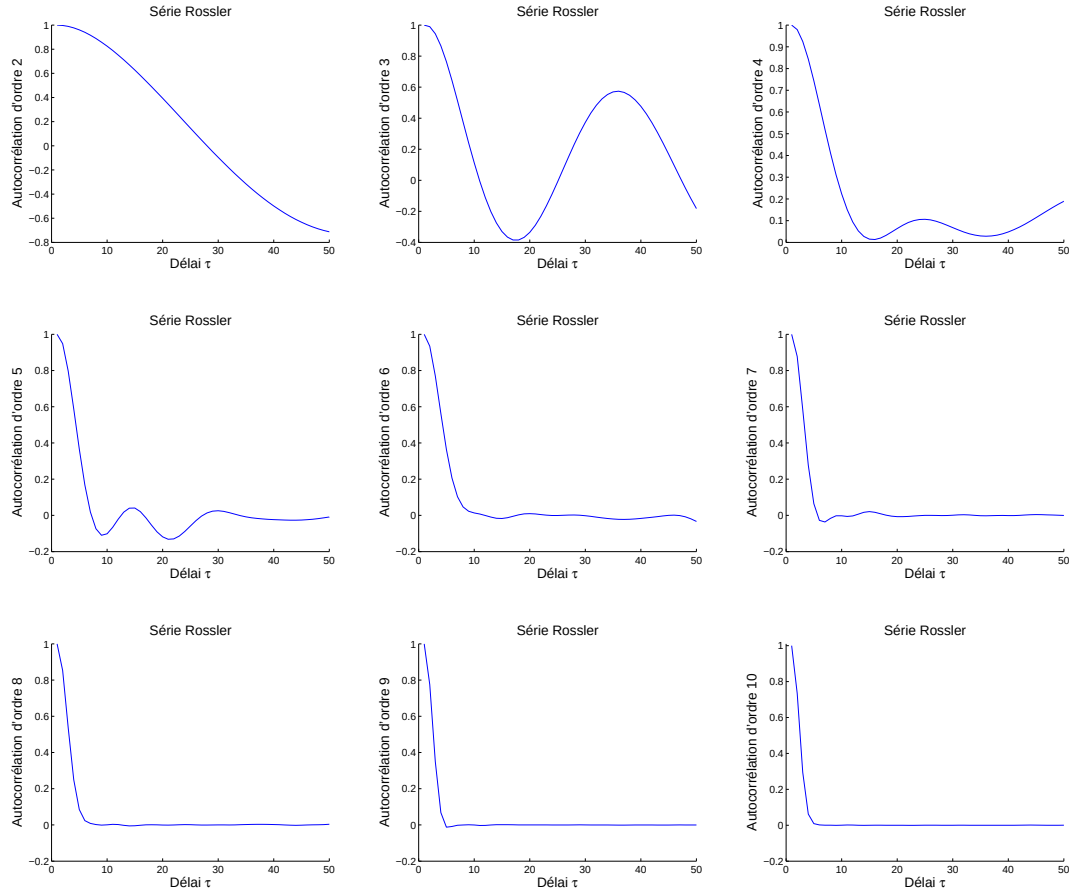

FIG. 5.8 – Autocorrélation d'ordre supérieur pour la série Rossler. L'ordre augmente ici de 2 à 10, en parcourant les graphes de gauche à droite, et de haut en bas.

dépendance qui devrait être visible, même dans les graphes des ordres supérieurs. Or, ces graphes sont presque uniformément plats, excepté pour les quelques premiers délais. En d'autres termes, pour un régresseur en dimension $p = 10$ construit avec un délai $\tau = 5$, par exemple, toutes les variables seraient très faiblement corrélées entre elles. Cependant, dans un tel régresseur, les deux premières composantes sont $x(t)$ et $x(t-5)$, qui sont très fortement corrélées si on s'en réfère au graphe de la corrélation entre deux variables, dans le coin supérieur gauche (ordre 2).

En répétant les expériences avec d'autres séries temporelles, on arrive aux mêmes conclusions. La Figure 5.9 présente les résultats obtenus pour la sélection du délai par ce même critère d'autocorrélation généralisée (5.5), pour la série Santa Fe A. Dans ce cas également, on fait les deux mêmes constatations que pour l'exemple précédent. La valeur du délai diminue lorsque la dimension p du régresseur augmente : de $\tau = 4$ en dimension 2 on passe à $\tau = 3$ en dimension 3 et 4, et $\tau = 2$ pour toutes les dimensions supérieures. Par ailleurs, on constate à nouveau que la courbe en fonction du délai τ est pratiquement

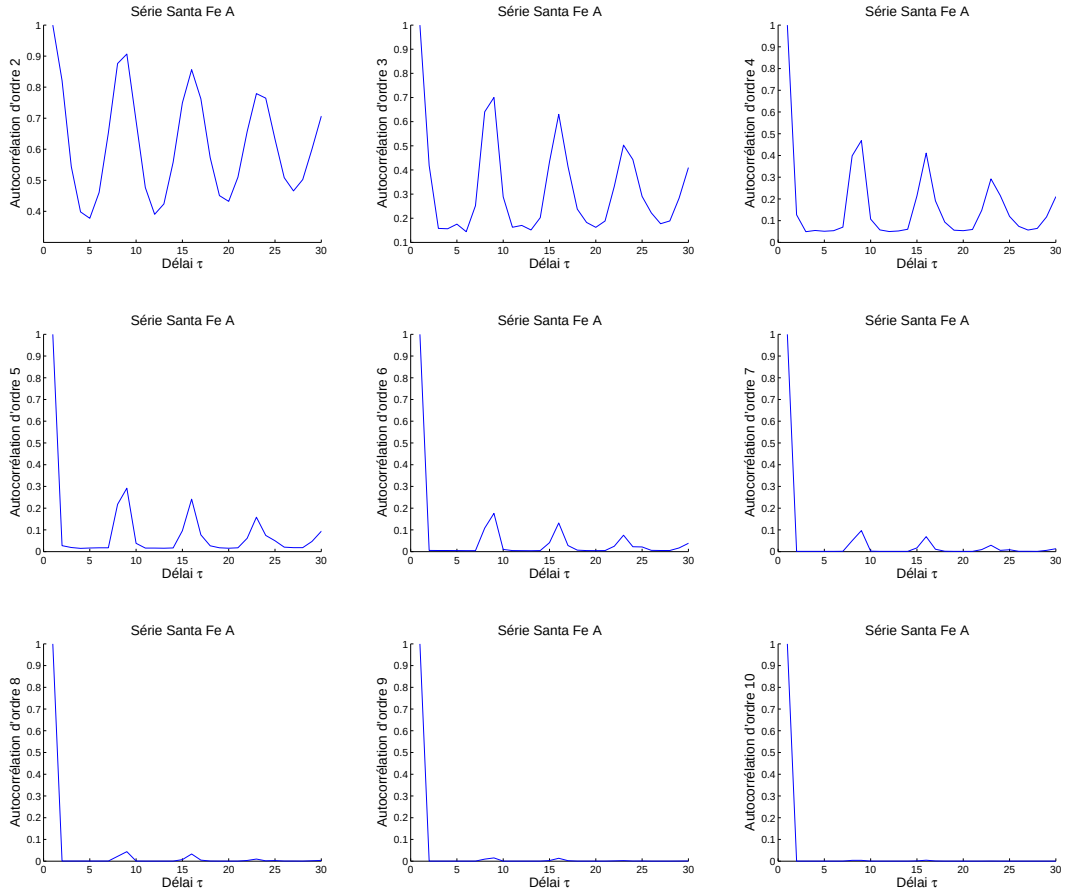

FIG. 5.9 – Autocorrélation d'ordre supérieur pour la série Santa Fe A, pour des ordres de 2 à 10.

plate pour les ordres élevés, à partir de l'ordre 7 dans ce cas-ci.

Pour résumer, sur base des Figures 5.8 et 5.9, on obtient donc deux enseignements. Premièrement, le délai, même unique, diminue lorsqu'il est sélectionné directement en haute dimension, au fur et à mesure que la taille du régresseur augmente. Par ailleurs, on constate que l'autocorrélation d'ordre supérieur ne doit être utilisée que pour des ordres relativement limités, la courbe devenant inexploitable lorsque l'ordre devient trop élevé.

5.5 Délai par information mutuelle en haute dimension

La généralisation de l'autocorrélation proposée dans la section précédente reste une mesure linéaire. Dans cette section, une généralisation de la sélection du délai par information mutuelle, critère non linéaire, est proposée.

La relation (5.2) définissant l'information mutuelle pour deux variables peut être facilement généralisée, au moins sur le plan théorique, pour un nombre quelconque de variables. En effet, on peut définir l'information mutuelle entre les v variables $x(1)$, $x(2)$, ..., $x(v)$

selon (32; 101) :

$$\mathrm{IM}(x(1), x(2), \ldots, x(v)) = \int \mathrm{P}\left[x(1), x(2), \ldots, x(v)\right] \log \left(\frac{\mathrm{P}\left[x(1), x(2), \ldots, x(v)\right]}{\mathrm{P}\left[x(1)\right] \mathrm{P}\left[x(2)\right] \ldots \mathrm{P}\left[x(v)\right]} \right) \mathrm{d}x(1)\ \mathrm{d}x(2)\ \ldots\ \mathrm{d}x(v). \tag{5.7}$$

Notons que, tout comme il est possible d'utiliser de deux manières différentes l'autocorrélation d'ordre supérieur, la définition (5.7) est valable pour deux généralisations distinctes de la méthodologie de sélection du délai par information mutuelle : la sélection d'un délai τ fixe, et la sélection de $p-1$ délais distincts $0 < \tau_1 < \tau_2 < \ldots < \tau_{p-1}$. Il suffit pour cela de remplacer les variables $x(1), \ldots, x(v)$ dans (5.7) par $x(t), x(t-\tau), \ldots, x(t-(p-1)*\tau)$ ou par $x(t), x(t-\tau_1), \ldots, x(t-\tau_p)$ respectivement.

Malgré la facilité d'écriture de la généralisation théorique décrite dans (5.7), le problème de cette définition est son calcul en pratique, qui est particulièrement difficile. En effet, les densités de probabilités, inconnues, doivent être estimées, ce qui ne peut être fait de façon très précise dans des espaces en haute dimension. Une approche alternative, basée sur les k-plus-proches-voisins, a été proposée récemment dans la littérature (101). Outre le fait qu'il ne faut pas estimer de densité, l'intérêt de cet estimateur particulier d'information mutuelle est qu'il est défini sur base d'un algorithme utilisant des vecteurs, qui peuvent être de dimension quelconque. Cet estimateur est décrit en détails dans (101), une implémentation étant par ailleurs disponible en ligne (7).

La Figure 5.10 illustre l'application de l'information mutuelle en haute dimension (5.7) dans le cas de la série Rossler. On y constate également deux phénomènes. Premièrement, la sélection d'un délai unique est moins aisée, les minima locaux étant de moins en moins nettement marqués lorsque p augmente. Ce phénomène est surtout visible à partir du calcul de l'information mutuelle entre 4 variables (le dernier graphe à droite de la première ligne, Figure 5.10). Deuxièmement, on constate sur cette même Figure 5.10 que l'utilisation de ce critère généralisé tend ici aussi à faire diminuer le délai sélectionné. On peut en effet voir que les courbes décroissent de plus en plus vite, en passant d'une figure à l'autre, ce qui se remarque par un redressement des courbes lorsqu'on les observe de gauche à droite, de bas en haut.

La Figure 5.11 est proposée pour illustrer l'utilisation de l'information mutuelle en haute dimension dans le cas de série Santa Fe A. Pour cette série, les minima locaux restent plus nettement marqués que dans le cas de la série Rossler. Cependant, on constate ici aussi que les différentes courbes ont encore cette même tendance à devenir de plus en plus lisses, surtout pour les premières valeurs du délai τ. L'observation des minima locaux reste pourtant possible dans ce cas précis. Par ailleurs, la pente des courbes des différents graphes est dans ce cas également de plus en plus négative au fur et à mesure que la dimension de l'espace augmente. Cette variation de la pente se traduit à nouveau par un redressement des courbes, un déplacement vers la gauche des minima locaux et donc une diminution du délai unique choisi directement en haute dimension.

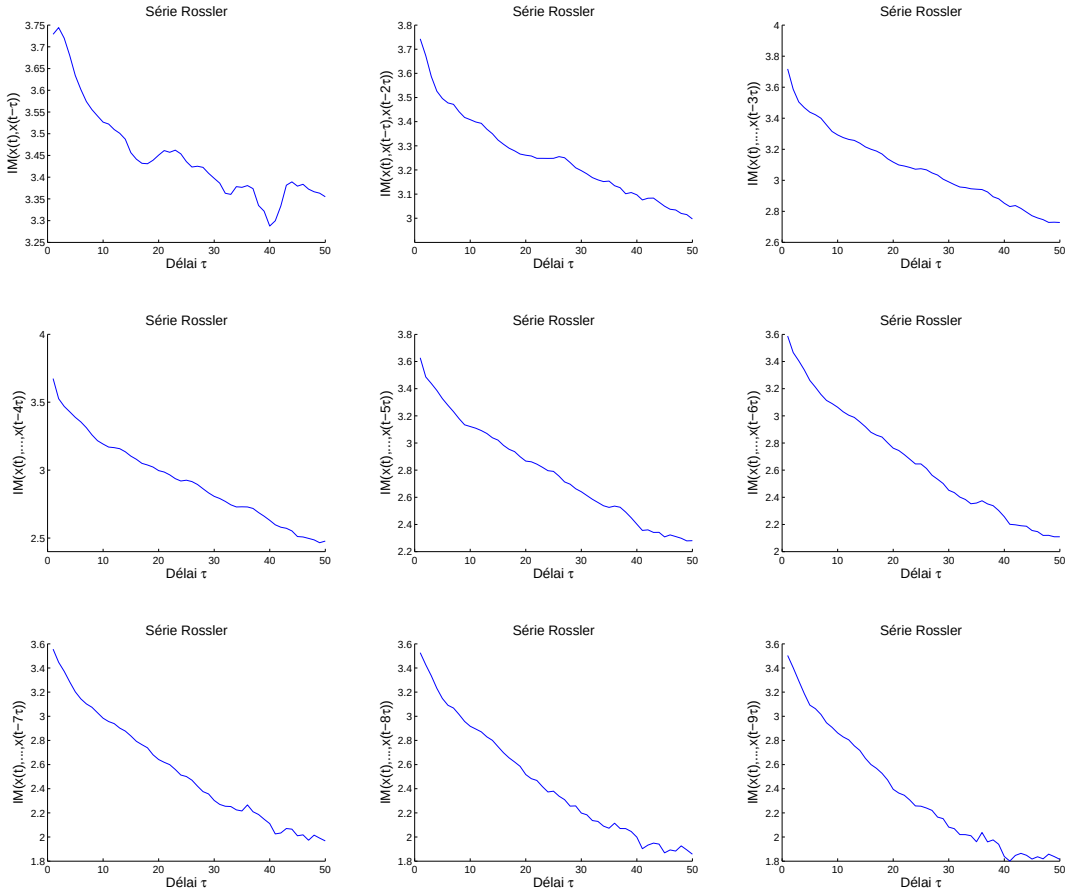

FIG. 5.10 – Courbes obtenues par information mutuelle en haute dimension, afin de sélectionner un délai unique, en utilisant de deux à dix variables respectivement de gauche à droite et de haut en bas.

Au delà de la diminution du délai préconisé par le critère d'information mutuelle en haute dimension, une conclusion cohérente avec les observations de l'autocorrélation d'ordre supérieur, on remarque en pratique deux limitations lors de l'utilisation de l'estimateur d'information mutuelle par k-plus-proches-voisins.

La première limitation est que, comme tout autre méthode de calcul de l'information mutuelle (qu'il s'agisse de méthodes par histogramme, noyaux ou splines), il ne s'agit toujours que d'un estimateur de cette information mutuelle. De manière générale, choisir un bon estimateur n'est pas chose aisée ; on l'a vu notamment lors des discussions biais/variance au Chapitre 2. C'est bien évidemment encore le cas ici. Néanmoins, le choix de cet estimateur particulier a été motivé par la possibilité de l'utiliser en haute dimension.

La seconde limitation est le temps calcul. L'algorithme tel que décrit dans (101) est quadratique par rapport au nombre de données N, et ce à cause de la recherche des plus proches voisins. Un grand soin a toutefois été apporté dans l'implémentation disponible en ligne afin de réduire le temps calcul, notamment en utilisant quelques heuristiques pour

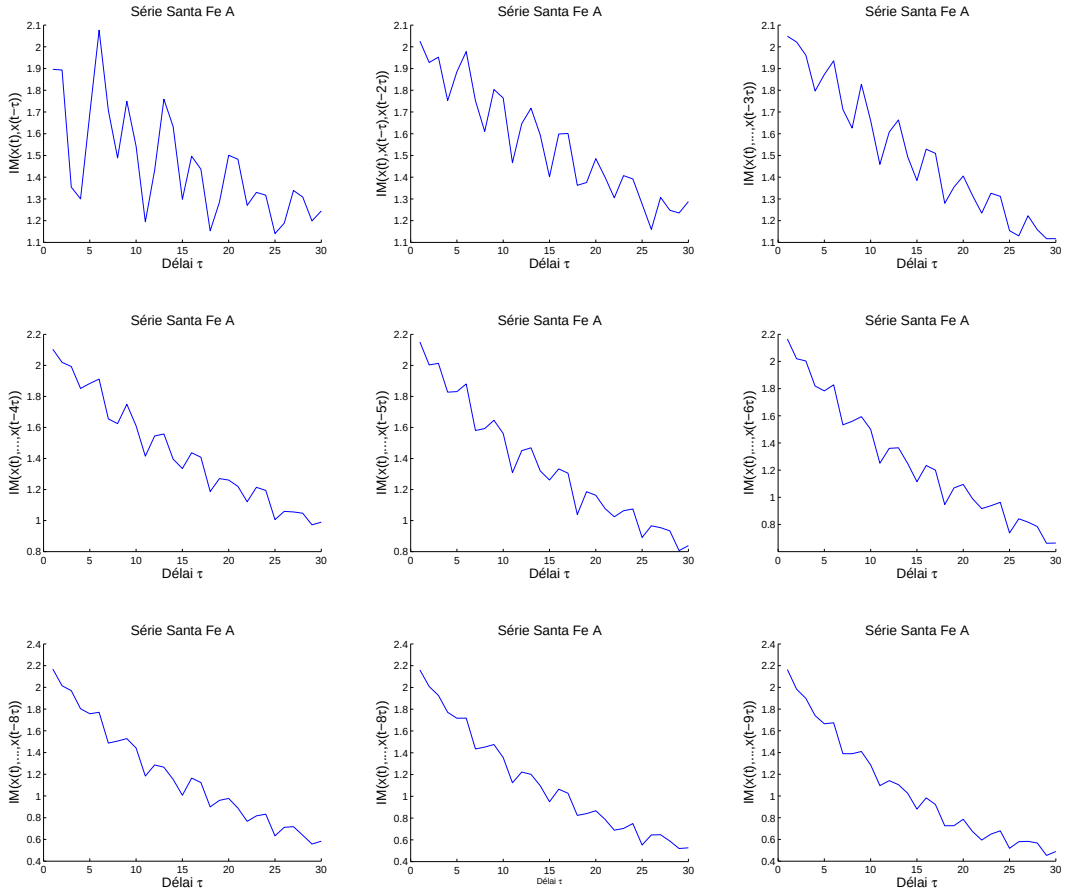

FIG. 5.11 – Courbes obtenues par information mutuelle en haute dimension pour sélectionner un délai unique, en utilisant de deux à dix variables, comme dans la Figure 5.10.

trouver les k-plus-proches-voisins en remplacement d'une recherche exhaustive (101). Dans le meilleur des cas, lorsque cette heuristique donne effectivement les k-plus-proches-voisins, le temps d'exécution devient linéaire en fonction de N, mais en moyenne, l'heuristique ne donne pas toujours exactement les k-plus-proches-voisins. Le temps nécessaire en pratique, lorsque l'estimateur est utilisé de façon ponctuelle est donc tout à fait raisonnable, mais il l'est beaucoup moins lorsqu'on l'utilise un grand nombre de fois, par exemple lorsqu'on essaie de sélectionner des délais différents, comme à la Section 5.7, et qu'il faut alors tester toutes les combinaisons de délais possibles en fonction de la dimension. On constate alors que le temps d'exécution n'est pas linéaire, par comparaison avec un critère réellement linéaire tel que la Distance à la Diagonale.

Par ailleurs, notons que l'approche proposée ici n'est pas la seule généralisation possible de l'information mutuelle en haute dimension. Mentionnons par exemple les travaux sur la corrélation totale (176), où la dépendance entre toutes les variables est prise en compte, ou encore les travaux sur l'information d'interaction (75; 118), où les dépendances entre

toutes les combinaisons de variables considérées sont également prises en compte. Par ailleurs, dans (101), deux estimateurs sont en fait proposés, l'un permettant d'obtenir l'information mutuelle entre plusieurs variables et l'autre l'information mutuelle entre deux variables dont l'une est un vecteur et l'autre un scalaire. Cette seconde version a été utilisée afin de sélectionner de manière supervisée un modèle, comme par exemple dans (86; 129; 152; 153). Ces travaux constituent une généralisation de la sélection de variables supervisée par information mutuelle, proposée par (9). Une autre approche a été utilisée dans (151) pour comparer la sélection supervisé et non supervisée d'un délai unique, directement en haute dimension. Tout comme illustré dans cette section pour le non supervisé, il a été observé dans (151) que le délai sélectionné est également plus court dans le cas supervisé lorsqu'il est sélectionné directement en haute dimension.

5.6 Délai par Distance à la Diagonale

Le critère de Distance à la Diagonale a été introduit au Chapitre 4 en tant que technique purement géométrique pour déplier des régresseurs. Ce critère peut toutefois être utilisé pour sélectionner le délai directement en haute dimension.

La Distante à la Diagonale permet de répondre aux deux limitations de l'estimateur d'information mutuelle par k-plus-proches-voisins. En effet, ce critère est une mesure de distance entre une distribution de points et une droite. Bien qu'il s'agisse d'une heuristique géométrique, la valeur du critère est obtenue sur base d'un calcul exact, et non d'une estimation. Ce calcul exact a l'avantage de ne pas être sensible aux questions de biais/variance auxquelles sont sujettes tout estimateur. Par ailleurs, le temps d'exécution de ce critère est toujours linéaire en fonction du nombre de données N. Notons qu'il s'agit là d'un ordre de temps d'exécution maximum, et non dans le meilleur des cas. Le critère de Distance à la Diagonale est donc en moyenne plus rapide que l'information mutuelle en haute dimension.

Pour rappel, la Distance à la Diagonale est définie dans la relation (4.12), en tant qu'application de la formule générale de la distance d'un point à une droite définie par deux points donnés (4.13). Il faut noter que ce critère est défini sur base de vecteurs représentant la position de points dans un espace de dimension p. Il est donc possible de l'appliquer directement à des régresseurs de dimension $p > 2$, tout comme dans l'étude du caractère significatif du clustering de séries temporelles, où la Distance à la Diagonale a été appliquée à des régresseurs de dimension 3, 5 et 10.

Le critère de Distance à la Diagonale est appliqué ici aux deux séries utilisées comme illustrations dans les deux sections précédentes, à savoir la série Rossler à la Figure 5.12 et la série Santa Fe A à la Figure 5.13. Notons que, comme indiqué à la Section 4.6, le but est ici de maximiser la Distance à la Diagonale afin d'obtenir une distribution aussi différente que possible de la diagonale principale de l'espace de dimension p. Pour la série Rossler, on observe à la Figure 5.12 que la pente de la courbe est de plus en plus élevée dans la partie croissante des différents graphes, lorsque p augmente. Cette augmentation

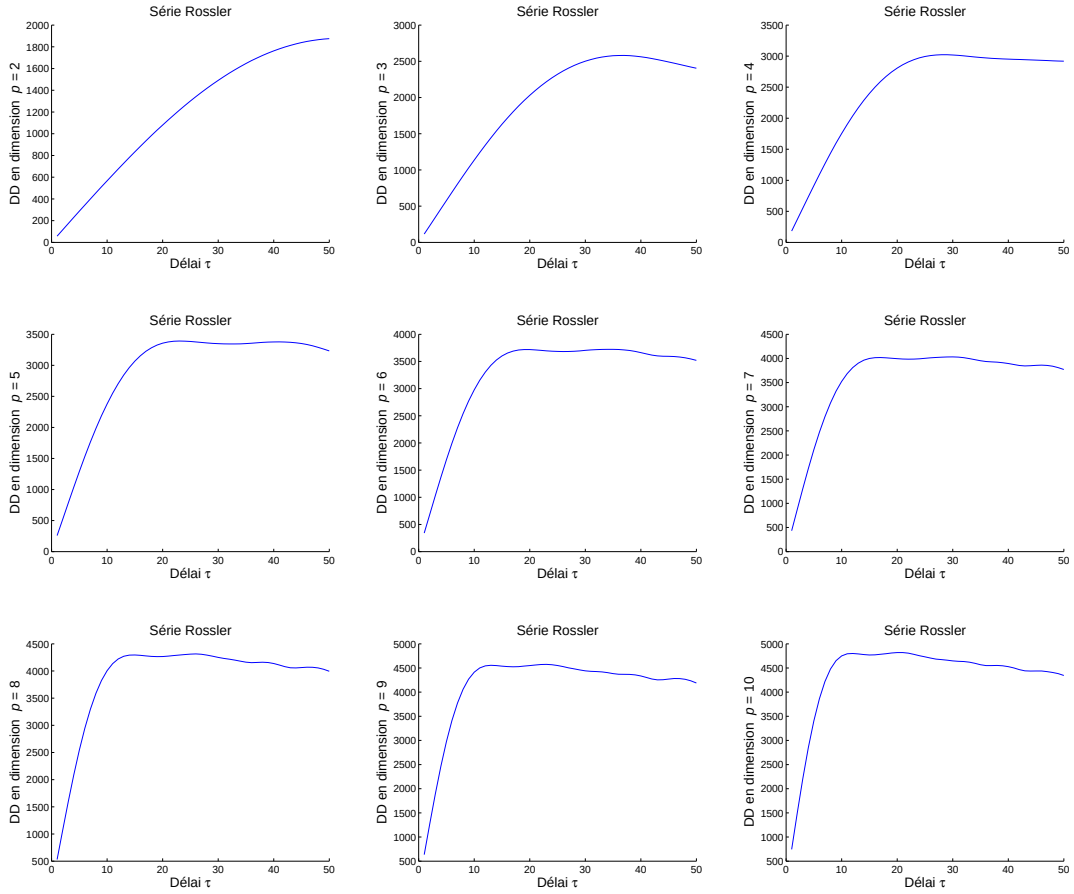

FIG. 5.12 – Courbes obtenues par Distance à la Diagonale pour des vecteurs de dimension deux à dix, de gauche à droite et de haut en bas.

de la pente se traduit par le déplacement vers la gauche du maximum. Concrètement, on assiste donc à une diminution de la valeur du délai τ, lorsqu'il est sélectionné directement en haute dimension. D'un délai de $\tau = 48$ en dimension 2, on passe à $\tau = 36$ en dimension 3, $\tau = 18$ en dimension 6 ou encore $\tau = 10$ en dimension 10. La même constat peut être fait pour la série Santa Fe A, même si les courbes présentées à la Figure 5.13 sont moins lisses que celles obtenues pour la série Rossler. D'un délai $\tau = 3$ en dimension 2, on passe à un délai de $\tau = 2$ en dimension 3, et un délai $\tau = 1$ en dimension 6 et plus.

Le critère de Distance à la Diagonale est donc une approche alternative à la sélection du délai, qui a l'avantage d'être plus rapide que l'information mutuelle en haute dimension, et d'être exploitable en toute dimension, contrairement à l'autocorrélation d'ordre supérieur. Néanmoins, ce critère est purement géométrique, et il a une tendance à choisir un délai plus long que nécessaire, afin de remplir encore un peu mieux l'espace de phase. Cette tendance se traduit par exemple par un dépliage parfois trop important, comme on peut le constater sur la Figure 4.5 où une représentation de l'espace de phase de la série Lorenz

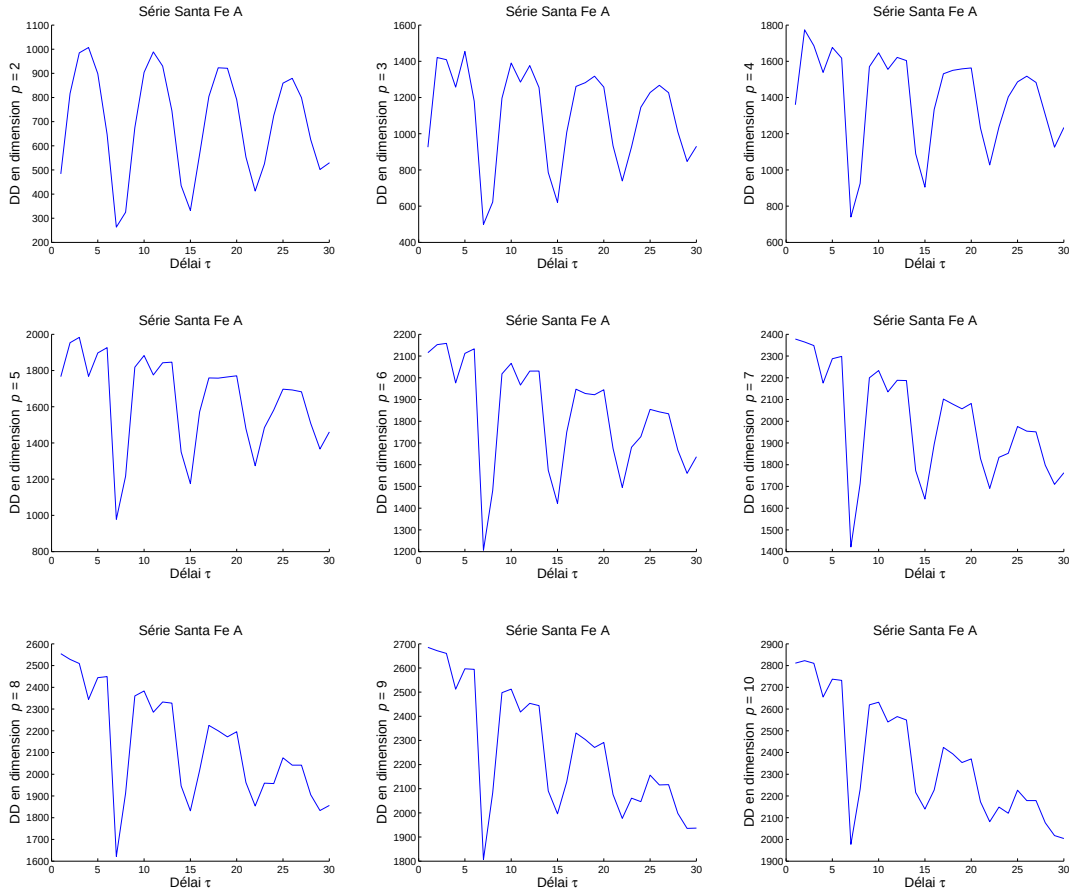

FIG. 5.13 – Courbes obtenues par Distance à la Diagonale pour des vecteurs de dimension deux à dix, de gauche à droite et de haut en bas.

est proposée, au centre. Ce graphe est à comparer avec la reconstruction obtenue par information mutuelle, proposée en début de ce chapitre, aux Figures 5.6 et 5.7.

Enfin, notons que le critère de Distante à la Diagonale possède quelques propriétés comparables aux propriétés des critères d'autocorrélation d'ordre supérieur et d'information mutuelle en haute dimension. Tout comme ces deux critères, la Distance à la Diagonale est insensible à une normalisation des données. L'échelle des valeurs obtenues est modifiée, mais la forme de la courbe ne change pas : les maxima locaux sont obtenus pour les mêmes délais. De plus, à nouveau comme pour les deux autres critères, le nombre de données influence peu le choix du délai. Par contre, la présence de bruit pénalise ce choix, ce qui est également le cas aussi bien pour l'autocorrélation que pour l'information mutuelle. Concernant le nombre de données, une augmentation de ce nombre se traduit par une augmentation du nombre de termes dans la somme (4.12), conduisant à un changement d'échelle. Mais les maxima locaux sont toujours situés aux mêmes endroits. Concernant le bruit, sa présence modifie la position des points de l'espace de phase, représentés par

les régresseurs, et fausse donc le calcul du critère. En effet, les positions respectives des régresseurs, et donc leurs distances à la diagonale principale, sont perturbées. La reconstruction est donc modifiée par la présence du bruit. Ces propriétés d'invariance par rapport à la normalisation, de faible dépendance au nombre de données et de dépendance au bruit ont été vérifiées expérimentalement.

5.7 Sélection multiple de délais

Dans les trois sections précédentes, une généralisation des critères d'autocorrélation et d'information mutuelle, ainsi que le critère de Distance à la Diagonale, ont permis de sélectionner un délai unique directement en haute dimension. Le principal enseignement de ces expériences est que la valeur du délai entre composantes d'un régresseur choisi directement en haute dimension diminue lorsque la taille p du régresseur, représentant la dimension de l'espace, augmente.

Dans cette section, les critères en haute dimension vont être utilisés pour sélectionner non plus un seul délai, mais une série de $p-1$ délais différents. L'étude va porter sur deux aspects. Premièrement, les critères vont être comparés en termes de valeurs de délai sélectionnées. Les critères étant différents, il est fort probable que les délais sélectionnés seront différents. Mais il est tout aussi intéressant de savoir si ces différences sont nettes, ou s'il existe une certaine cohérence entre les délais sélectionnés par les différentes approches. De plus, il est intéressant d'observer si l'emploi de délais différents influence significativement, ou non, la qualité de la prédiction.

Sur base des résultats expérimentaux des sections précédentes, seuls deux critères en haute dimension sont retenus, à savoir les méthodes d'information mutuelle et de Distance à la Diagonale. L'autocorrélation d'ordre supérieure n'est pas utilisée dans les différentes expériences qui vont être décrites ici, à cause de la difficulté d'interprétation des résultats obtenus décrite à la section 5.4.

5.7.1 Comparaison en termes de délais sélectionnés

Dans un but d'illustration, la méthodologie de sélection de plusieurs délais en haute dimension est d'abord décrite en détails dans le cas de la série Santa Fe A. Les résultats obtenus avec deux autres séries sont ensuite présentés, parmi tous les résultats expérimentaux obtenus sur de nombreuses séries temporelles.

Considérons des régresseurs en deux dimensions. La Figure 5.14 présente les valeurs obtenues pour différents délais via information mutuelle, à gauche, et via Distance à la Diagonale, à droite. Sur base de ces graphes, on sélectionne $\tau_1^{IM} = 2$ par information mutuelle, et $\tau_1^{DD} = 3$ par Distance à la Diagonale. Les délais obtenus sont effectivement différents, ce à quoi on pouvait s'attendre puisque ces deux critères mesurent des grandeurs différentes. Les valeurs retenues pour les délais τ_1^{IM} et τ_1^{DD} restent néanmoins cohérentes.

Les reconstructions obtenues dans l'espace de phase en deux dimensions sont présentées à la Figure 5.15. Le graphe de droite présente la reconstruction obtenue en sélectionnant

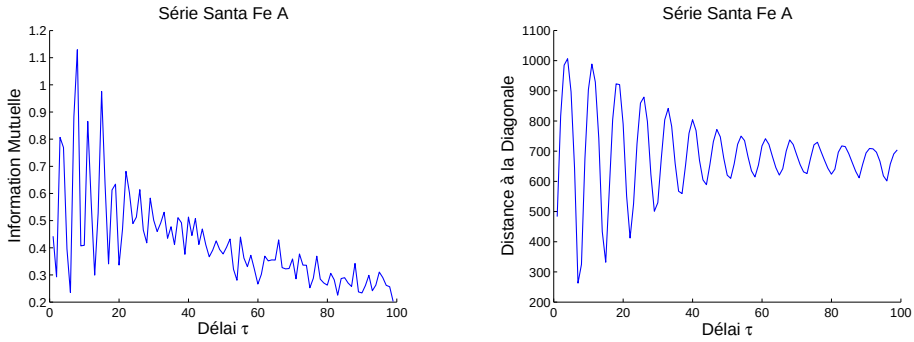

FIG. 5.14 – Graphes de l'information mutuelle (gauche) et de la Distance à la Diagonale (droite) pour la série Santa Fe A. Les régresseurs sont ici de dimension $p = 2$.

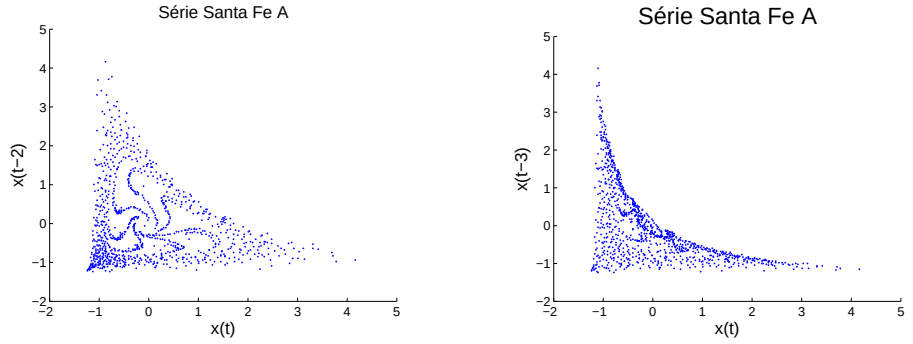

FIG. 5.15 – Reconstruction de la série Santa Fe A dans l'espace de phase en deux dimensions. A gauche, la reconstruction est obtenue en sélectionnant le délai par information mutuelle, à droite par Distance à la Diagonale.

le délai par Distance à la Diagonale. On observe sur ce graphe une différence assez nette, malgré la cohérence entre les valeurs τ_1^{IM} et τ_1^{DD}. Il existe dans le cas du critère de Distance à la Droite un dépliage excessif dans la partie proche de la diagonale. Ce dépliage excessif est une conséquence directe de la définition du critère, observable dans le cas où une partie de la reconstruction de la série se trouve le long de la diagonale : le but du critère de Distance à la Diagonale est justement de maximiser le dépliage pour être le plus possible distinct de la diagonale.

Observons ce qui se passe maintenant pour $p \geq 3$. D'un point de vue pratique, il est plus difficile de représenter sur un graphe en deux dimensions les valeurs obtenues par un critère fonction de deux délais (ou plus), quel que soit le critère utilisé. En effet, pour $p = 2$, les valeurs obtenues pour les différents délais peuvent être représentés sur un graphe du critère en fonction de la valeur de τ. Par contre, cette représentation n'est plus possible lorsqu'on utilise deux délais τ_1 et τ_2, puisqu'il faut maintenant représenter les valeurs d'un critère en fonction de deux paramètres.

Dans les représentations graphiques qui suivent, l'axe des abscisses n'est plus fonction

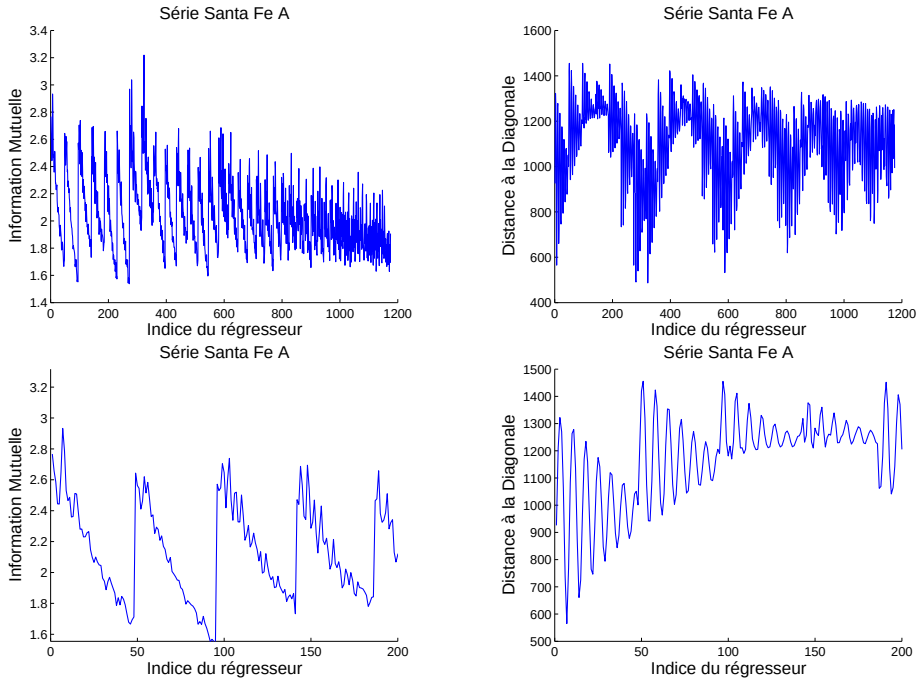

FIG. 5.16 – Graphes de l'information mutuelle (gauche) et de la Distance à la Diagonale (droite) pour la série Santa Fe A, en haut. Un agrandissement des deux graphes est proposé dans la partir inférieure de la figure. Les régresseurs sont ici de dimension $p = 3$.

du délai, mais de l'indice du régresseur, construit sur base d'une combinaison particulière de délais, dans la suite de tous les régresseurs possibles. Concrètement, on construit les régresseurs dans un certain ordre : on commence par fixer un intervalle de valeurs que peuvent prendre τ_1 et τ_2. Ensuite, on fixe temporairement la valeur de τ_1 et on considère toutes les valeurs possibles de τ_2, puis on recommence en incrémentant la valeur de τ_1, etc. jusqu'au terme de l'intervalle de valeurs considéré. Par construction, les régresseurs sont donc ordonnés. Deux commentaires doivent être faits par rapport à cette démarche de construction systématique des régresseurs. Premièrement, la dépendance temporelle a, bien évidemment, une grande importance. Ne sont donc considérés comme régresseurs que les vecteurs dont les délais sont ordonnés de manière croissante dans le temps, ce qui réduit sensiblement le nombre de combinaisons qui sinon exploserait de manière factorielle. Deuxièmement, on ne considère que les régresseurs qui commence par la variable $x(t)$. En effet, il ne sert à rien de comparer les valeurs obtenues respectivement pour, par exemple, $(x(t), x(t-7), x(t-11))$ et $(x(t-1), x(t-8), x(t-12))$ ou encore $(x(t-5), x(t-12), x(t-16))$: il s'agit là du même régresseur, il est simplement décalé dans le temps.

Dans la partie supérieure de la Figure 5.16, on peut voir les valeurs obtenues selon les critères d'information mutuelle et de Distance à la Diagonale en dimension $p = 3$. Evidemment, ces figures sont moins faciles à lire que lorsqu'il n'y a qu'une variable, il y a donc lieu de les agrandir, ce qui est proposé dans la partie inférieure de cette même figure.

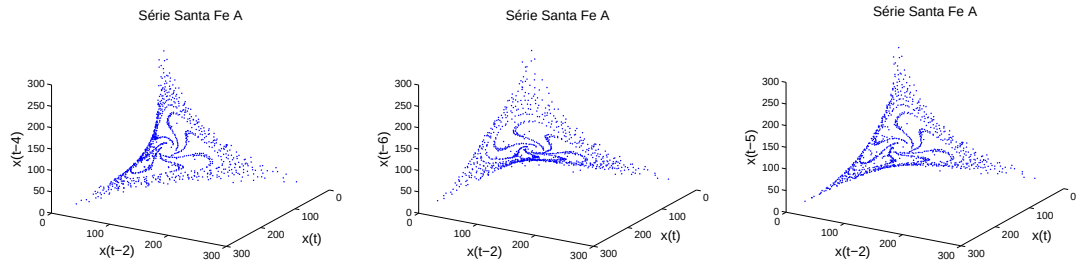

FIG. 5.17 – Reconstruction de la série Santa Fe A dans l'espace de phase en trois dimensions. A gauche, la reconstruction est obtenue en sélectionnant le délai par information mutuelle en deux dimensions et en utilisant un multiple de ce délai ; au centre en utilisant l'information mutuelle en haute dimension, et à droite par Distance à la Diagonale.

Pour le graphe de l'information mutuelle, en bas à gauche, on peut y voir la concaténation d'une série de courbes correspondant chacune à un ensemble de régresseurs ayant les deux premières composantes en commun. Pour chacune de ces petites courbes, on sélectionne le premier minimum local. Les différents minima locaux sont ensuite comparés pour obtenir la plus petite valeur d'information mutuelle. Dans le cas de la série Santa Fe A, le régresseur sélectionné est $\mathbf{x}_t^{IM} = (x(t), x(t-2), x(t-6))$.

On applique une stratégie comparable pour la Distance à la Diagonale, et on sélectionne cette fois le premier maximum local de chaque groupe d'oscillations visibles dans les deux courbes de droite de la Figure 5.16, chacun de ces groupes d'oscillations correspondant à un ensemble de régresseurs dont les deux premières composantes sont les mêmes. En comparant les maxima locaux entre eux et en sélectionnant le plus élevé, on est amené à choisir le régresseur $\mathbf{x}_t^{DD} = (x(t), x(t-2), x(t-5))$.

On peut maintenant comparer les reconstructions obtenues en trois dimensions en utilisant $\mathbf{x}_t^{IM}$ et $\mathbf{x}_t^{DD}$. Pour être complet, la reconstruction obtenue en utilisant un multiple du délai obtenu par information mutuelle en deux dimensions est également proposée. Ces trois graphes sont proposés à la Figure 5.17. Il est intéressant de noter que, bien que les régresseurs sélectionnés soient différents, les reconstructions proposées sont ici encore cohérentes entre elles, à quelques détails près au niveau des extrémités de la reconstruction. En appliquant des rotations de l'espace de phase, on peut observer que les trois reconstructions présentées ici sont en fait assez semblables quel que soit l'angle d'observation, ce qui n'est a priori pas aussi évident, et n'est surtout pas le cas avec d'autres régresseurs pour lesquels la reconstruction est moins clairement distinguable, quel que soit l'angle d'observation.

En dimension(s) supérieure(s), on obtient le même type de courbes que celles de la Figure 5.16. La Figure 5.18 présente les résultats pour toutes les combinaisons possibles lorsqu'on considère $0 < \tau_1 < \tau_2 < \tau_3 \leq 50$, en ne prenant que les régresseurs dont la première composante est $x(t)$, comme expliqué ci-dessus. Les courbes sont moins lisibles.

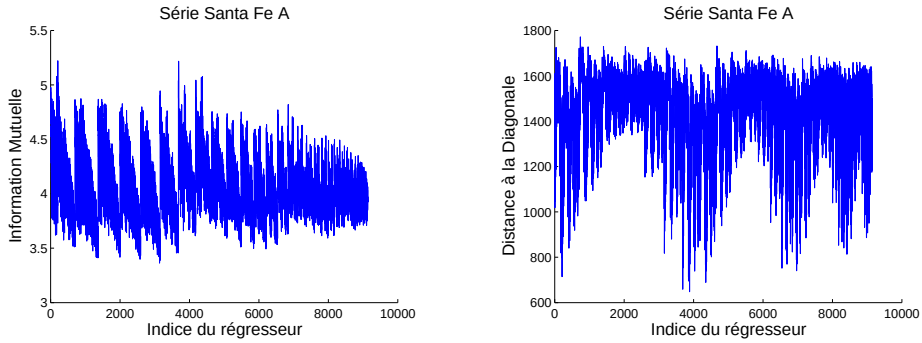

FIG. 5.18 – Graphes de l'information mutuelle (gauche) et de la Distance à la Diagonale (droite) pour la série Santa Fe A. Les régresseurs sont ici de dimension $p = 4$.

p	Information mutuelle classique	Information mutuelle haute dimension
2	$x(t), x(t-2)$	/
3	$x(t), x(t-2), x(t-4)$	$x(t), x(t-2), x(t-6)$
4	$x(t), x(t-2), x(t-4), x(t-6)$	$x(t), x(t-3), x(t-5), x(t-9)$
5	$x(t), x(t-2), x(t-4), x(t-6), x(t-8)$	$x(t), x(t-1), x(t-5), x(t-6), x(t-11))$
6	$x(t), x(t-2), x(t-4), x(t-6), x(t-8), x(t-10)$	$x(t), x(t-1), x(t-2), x(t-5), x(t-6), x(t-11)$

TAB. 5.1 – Composantes retenues pour la construction des régresseurs par information mutuelle classique et information mutuelle en haute dimension, série Santa Fe A.

p	Information mutuelle classique	Distance à la Diagonale
2	$x(t), x(t-2)$	$x(t), x(t-3)$
3	$x(t), x(t-2), x(t-4)$	$x(t), x(t-2), x(t-5)$
4	$x(t), x(t-2), x(t-4), x(t-6)$	$x(t), x(t-2), x(t-4), x(t-6)$
5	$x(t), x(t-2), x(t-4), x(t-6), x(t-8)$	$x(t), x(t-3), x(t-4), x(t-6), x(t-9)$
6	$x(t), x(t-2), x(t-4), x(t-6), x(t-8), x(t-10)$	$x(t), x(t-2), x(t-4), x(t-5), x(t-6), x(t-9)$

TAB. 5.2 – Composantes retenues pour la construction des régresseurs par information mutuelle classique et Distance à la Diagonale, série Santa Fe A.

Néanmoins, en appliquant la même procédure de sélection du premier minimum ou maximum local, selon le critère utilisé, on obtient le régresseur optimal pour la dimension considérée.

La même procédure est répétée en dimension $p = 5$ et $p = 6$. Le résumé des régresseurs obtenus pour $p = 2$ jusque $p = 6$ est proposé aux Tableaux 5.1 et 5.2, à chaque fois en comparaison avec les régresseurs obtenues par information mutuelle classique en deux dimensions. Dans ces tableaux, on constate que les régresseurs sélectionnés sont généralement comparables avec ceux obtenus par information mutuelle en dimension $p = 2$, en utilisant un délai constant et des multiples de ce délai. Les différences sont souvent limitées au décalage d'une composante $x(t)$, à un, deux ou trois pas de temps, jamais plus. Bien que les critères mesurent des grandeurs différentes, il est intéressant de noter que les méthodes sont relativement cohérentes entre elles et que les régresseurs finalement obtenus sont comparables entre eux.

p	Info. mut. classique	Info. mut. haute dimension	Distance à la Diagonale
2	$x(t), x(t-11)$	/	$x(t), x(t-31)$
3	$x(t), x(t-11), x(t-22)$	$x(t), x(t-4), x(t-15)$	$x(t), x(t-3), x(t-33)$
4	$x(t), x(t-11), x(t-22), x(t-33)$	$x(t), x(t-6), x(t-11), x(t-20)$	$x(t), x(t-13), x(t-15), x(t-30)$

TAB. 5.3 – Composantes retenues pour la construction des régresseurs par information mutuelle en haute dimension et Distance à la Diagonale, comparés aux composantes retenues par information mutuelle classique. La série considérée est la série Lorenz.

p	Info. mut. classique	Info. mut. haute dimension	Distance à la Diagonale
2	$x(t), x(t-3)$	/	$x(t), x(t-1)$
3	$x(t), x(t-3), x(t-6)$	$x(t), x(t-3), x(t-8)$	$x(t), x(t-5), x(t-8)$
4	$x(t), x(t-3), x(t-6), x(t-9)$	$x(t), x(t-1), x(t-4), x(t-9)$	$x(t), x(t-1), x(t-4), x(t-9)$

TAB. 5.4 – Tableau des composantes retenues pour la construction des régresseurs par les trois critères, série $AR(3)$.

La même procédure a été appliquée à la série Lorenz et à une série linéaire générée artificiellement suivant un modèle $AR(3)$, décrite dans l'Annexe A. Les Tableaux 5.3 et 5.4 présentent une comparaison des régresseurs obtenus selon les trois approches, pour ces deux autres séries temporelles. Toutes les combinaisons de délais $0 < \tau_1 < \ldots < \tau_{p-1} \leq 50$, telles que la première composante est $x(t)$, ont été testées pour la série Lorenz, de même que pour la série $AR(3)$, à la différence que l'intervalle de valeur pour les délais $\tau_1, \ldots \tau_{p-1}$ a été limité à 30 au lieu de 50.

En observant les Tableaux 5.1 à 5.4, on peut retirer deux informations. La première est que les critères utilisés permettent, à nouveau, de sélectionner des composantes sensiblement différentes. Ce fait est relativement évident puisque les trois critères mesurent des quantités différentes, comme mentionné plus tôt dans cette section. La seconde information est qu'il y a, au-delà de petites différences, une certaine cohérence dans les valeurs retenues par les différentes méthodes. De plus, les trois méthodes fournissent des régresseurs compacts dans le temps, malgré les intervalles relativement larges de valeurs possibles pour les délais. Les reconstructions dans l'espace de phase sont donc comparables, les différences dans les régresseurs sont généralement limitées au décalage d'un ou deux composantes, à quelques pas de temps, principalement dans le cas des séries Santa Fe A et $AR(3)$.

Cependant, il faut nuancer le cas du critère de Distance à la Diagonale, qui a tendance à déplier excessivement les distributions de régresseurs dans l'espace de phase. Les graphes des Figures 5.6 et 5.7 illustrent le fait que le choix d'un délai en deux dimensions est sous optimal en trois dimensions. Or, dans le tableau 5.3, on peut voir apparaître un délai de 30 ou de 33 pour la dernière composante des régresseurs construits par Distance à la Diagonale. Ces valeurs de 30 et de 33 sont proches (voir identique) d'un multiple de $\tau = 11$ sélectionné en deux dimensions. Or, on a pu observer à la Section 5.3 que les multiples de ce délai fournissent généralement un dépliage de moins bonne qualité, les régresseurs n'étant plus suffisamment compacts dans le temps. On peut donc se douter que les délais sélectionnés par Distance à la Diagonale sont, dans le cas de la série Lorenz, trop longs.

5.7.2 Comparaison en termes de prédiction

Dans cette section, il est étudié dans quelle mesure les régresseurs obtenus, selon une des trois approches utilisées dans la section précédente ont un impact sur les performances en prédiction à un pas de modèles de prédiction. A priori, rien ne permettait de dire que des régresseurs sélectionnés suivant des approches différentes étaient cohérents entre eux malgré leurs différences, mais ils le sont. Il est dès lors intéressant d'observer si ces régresseurs conduisent à des modèles de prédiction dont les performances sont elles aussi comparables ou si, au contraire, des régresseurs obtenus suivant une méthode particulière permettent d'améliorer significativement les prédictions fournies par un modèle construit sur base de ces régresseurs par rapport aux modèles construits en utilisant les régresseurs obtenus suivant les autres approches.

Des RBFN sont utilisés en tant que modèles de prédiction pour un horizon limité à $h = 1$. La procédure de sélection de structure RBFN est le bootstrap, avec un nombre $R = 50$ de réplications, pour toutes les expériences décrites ci-dessous. Les délais utilisés dans les régresseurs utilisés pour construire les RBFN sont ceux obtenus à la section précédente, et repris aux Tableaux 5.1 et 5.2. Des modèles RBFN dont la complexité varie de 10 à 200 noyaux Gaussiens, par incrément de 10, ont été utilisés. Les estimations de l'erreur de généralisation obtenue par bootstrap, pour les différentes structures de modèle, et pour les différents régresseurs, sont présentées à la Figure 5.19. On note, sur le graphe en haut à droite correspondant aux régresseurs de dimension $p = 4$ une superposition des courbes obtenues en utilisant le régresseur obtenu par Distance à la Diagonale et celui obtenu par information mutuelle à deux variables. Cette superposition est normale, les deux régresseurs étant dans ce cas précis identiques.

On peut faire trois observations sur base des courbes de la Figure 5.19. Tout d'abord, le minimum des trois courbes d'erreur n'est pas obtenu pour une même structure de modèle. Ensuite, les valeurs respectives des minima obtenus pour les différentes reconstructions ne sont pas strictement les mêmes. Enfin, les courbes se croisent ; l'ordre dans lequel elles apparaissent est en effet modifié non seulement au sein d'un même graphe, mais aussi et surtout d'un graphe à l'autre. En d'autres termes, on ne peut pas mettre en évidence une approche de sélection de délai(s) qui permettrait d'obtenir un modèle construit sur base des régresseurs utilisant ce(s) délai(s) et dont l'erreur de généralisation serait toujours meilleure (ou toujours moins bonne) que les deux autres approches. Aucun critère ne fournit en effet des courbes qui sont systématiquement inférieures à celles obtenues par les autres critères.

Les mêmes expériences sont répétées pour la série Lorenz et pour la série $AR(3)$. La complexité des modèles RBFN utilisés varie de 10 à 200 noyaux par incrément de 10 dans le cas de la série Lorenz, et de 1 à 20 noyaux par incrément de 1 dans le cas de la série $AR(3)$. Les courbes d'erreur obtenues pour la série Lorenz, en utilisant les régresseurs repris au Tableau 5.3, sont présentées à la Figure 5.20, celles obtenues pour la série $AR(3)$ sur base des régresseurs du Tableau 5.4 à la Figure 5.21. Sur base de ces graphes, on fait les mêmes observations que pour la série Santa Fe A, si ce n'est que pour la série

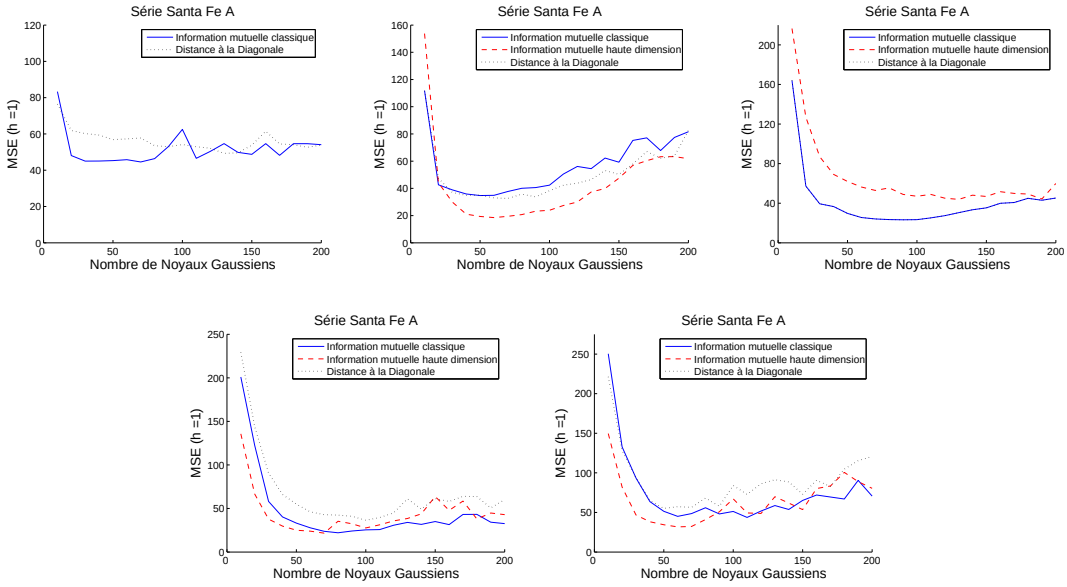

FIG. 5.19 – Estimations de l'erreur de généralisation par bootstrap, pour des RBFN comptant de 10 à 200 noyaux. De gauche à droite, et de base en haut : des régresseurs de taille $p = 2$, $p = 3$, ..., $p = 6$.

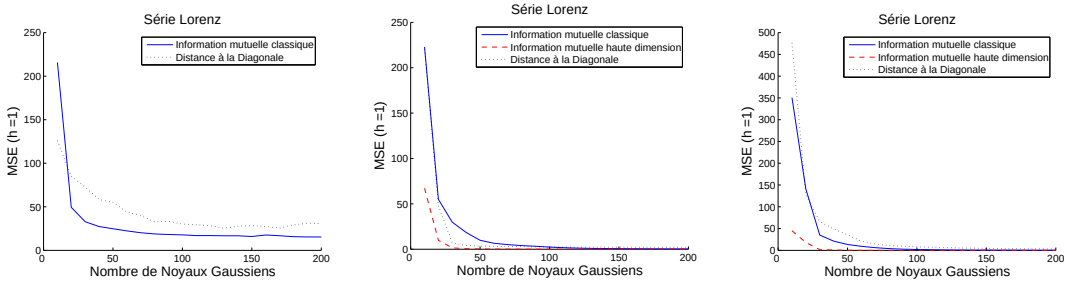

FIG. 5.20 – Estimations de l'erreur de généralisation par bootstrap, série Lorenz. De gauche à droite : des régresseurs de taille $p = 2$, $p = 3$ et $p = 4$.

Lorenz, les estimations de l'erreur de généralisation obtenues aux minima respectifs sont dans ce cas très proches les unes des autres. On observe en effet des courbes très plates, ce qui s'explique par le fait que la correction de biais du terme d'optimisme augmente très lentement, même en considérant des complexités jusque 300 noyaux Gaussiens.

5.7.3 Analyse des observations

Dans cette sous-section, les observations déduites des résultats expérimentaux obtenus aux sous-sections précédentes sont discutées. Les enseignements qu'on peut en retenir, étant d'origine expérimentale, sont donc d'ordre qualitatif.

Le fait que le minimum des courbes d'erreur ne soit pas obtenu pour une seule et

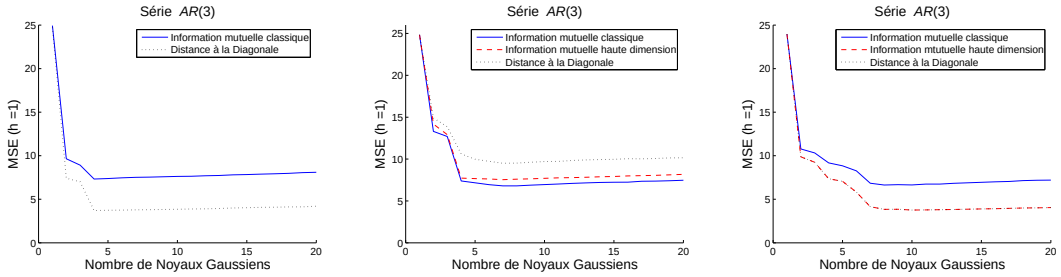

FIG. 5.21 – Estimations de l'erreur de généralisation par bootstrap obtenues pour des reconstruction de la série $AR(3)$ dans des espaces de dimension $p = 2$, $p = 3$ et $p = 4$, respectivement de gauche à droite.

même structure dans les différents cas n'est en fait pas très surprenant. Excepté lorsque les régresseurs utilisent les mêmes délais, on a en fait des reconstructions sensiblement différentes, dans l'espace de phase. Bien que la dynamique de la série temporelle soit unique, sa reconstruction peut être plus ou moins fidèle. Il faut donc selon le cas un modèle plus ou moins complexe, en fonction de la qualité du dépliage. De même, la présence de bruit dans les données explique également en partie cette observation, le bruit influençant la reconstruction.

Un autre corollaire de la différence de reconstruction, en utilisant différents régresseurs, c'est l'estimation de l'erreur de généralisation obtenue pour les différents minima. Tout comme la reconstruction influence le choix de la structure, il est évident que la reconstruction influence également la qualité de la modélisation, et donc au final l'erreur de généralisation des modèles. La reconstruction peut donc favoriser la modélisation tout comme elle peut la pénaliser, ce qui tend alors à faire augmenter l'erreur de généralisation, même pour le modèle optimal de la reconstruction considérée, par rapport aux modèles issus d'autres reconstructions.

L'observation selon laquelle les différentes courbes ont tendance à se présenter dans un ordre différent lorsque p varie s'explique une fois encore par la reconstruction dans l'espace de phase. D'un graphe à l'autre, on a une reconstruction avec des délais obtenus selon chaque critère, cette reconstruction pouvant être plus ou moins favorable à la modélisation. Si la reconstruction est favorable, la courbe aura tendance à rester constamment inférieure aux autres courbes. Elle sera bien évidemment supérieure dans le cas contraire. Le fait que les courbes se croisent parfois, à l'intérieur d'un même graphe, est probablement un effet des rééchantillonnages de la procédure boostrap : il est possible que les ensembles d'apprentissage bootstrapés ne reprennent pas l'ensemble de la dynamique de la série. Bien évidemment, des tels ensembles pénalisent le modèle lorsqu'il est confronté à de nouvelles données qui ne correspondent pas tout à fait à la dynamique apprise sur un échantillon peu représentatif de la véritable dynamique de la série considérée.

Notons qu'il est tout à fait normal que les courbes d'erreur obtenues par des modèles construits sur base de régresseurs identiques, bien que sélectionnés par des approches

différentes, se superposent. Les régresseurs utilisant les mêmes délais, les reconstructions dans l'espace de phase sont les mêmes. Les échantillons bootstrappés étant construits de la même façon pour les trois approches (ce qui est un minimum si on veut pouvoir comparer les expériences), il n'y a donc aucun effet dû à l'échantillonnage. Les seules différences qui pourraient survenir concernent la convergence des différentes applications de l'algorithme RBFN, dont l'initialisation change. En choisissant adéquatement les paramètres des RBFN, la convergence de l'algorithme conduit à un modèle comparable, quelle que soit la méthode utilisée pour sélectionner le(s) délai(s). On a donc des modèles comparables, appris sur des échantillons identiques, et dont les régresseurs ont été construits de la même façon, en utilisant des délais strictement les mêmes. Il est donc tout à fait normal que les erreurs de généralisation soient identiques et que les courbes finalement obtenues se superposent.

Enfin, on peut conclure cette discussion en rappelant qu'une "bonne" reconstruction de l'espace de phase permet d'aider l'étape de modélisation d'une série temporelle, et permet donc au final d'obtenir de bonnes prédictions. Il faut néanmoins pouvoir mesurer quantitativement ce que serait une "bonne" reconstruction, éventuellement par comparaison avec la dynamique réelle de la série temporelle. Or, cette dynamique est généralement inconnue. Même lorsque la série est générée artificiellement sur base de systèmes déterministes, comme par exemple pour les séries Lorenz ou Rossler, la représentation de référence de la dynamique, que ce soit dans ce travail ou dans la littérature, n'est jamais ... qu'une reconstruction par immersion. Grâce aux expériences de cette section, on sait maintenant que plusieurs critères peuvent être utilisés indifféremment pour obtenir des régresseurs comparables en termes de reconstruction dans l'espace de phase, bien que sensiblement différents en termes de délais sélectionnés. On sait aussi que ces régresseurs permettent de construire des modèles de prédiction de performance comparable, quelle que soit l'approche utilisée pour obtenir le(s) délai(s) utilisé(s) dans les régresseurs.

5.7.4 En pratique

Un point non négligeable, dans des expériences comme celles présentées dans les sous-sections précédentes, est le temps calcul. Comme on peut le voir à partir des Figures 5.14, 5.16 et 5.18, le nombre de combinaisons possibles de valeurs de délais à considérer augmente factoriellement lorsque la dimension des régresseurs augmente. Ce nombre de possibilités augmente par ailleurs en fonction de la limite supérieure des valeurs de délai testées. Si le critère de Distance à la Diagonale reste de complexité linéaire, l'information mutuelle en haute dimension est en moyenne moins performante. Si on veut donc tester plusieurs critères de sélection de délais, il faut donc additionner les temps d'exécution pour chacun des critères, ce qui représente beaucoup de temps calcul. Concrètement, pour tester plusieurs dimensions, même en se limitant à $p = 4$, avec un délai maximum relativement limité de $\tau = 50$, plusieurs heures sont nécessaires pour tester toutes les combinaisons de délais possibles, voire plusieurs jours en fonction du nombre de données dans la série ...

Notons que la dimension p des régresseurs a été testée ici de façon exhaustive, pour des valeurs relativement raisonnable, dans le but d'observer le comportement des critères de sélection du délai en fonction de p. Il est évident qu'en pratique, il est conseillé de déterminer la dimension intrinsèque d_i et d'en déduire p. Pour ce faire, des méthodes telles que la dimension de corrélation, le Box-Couting ou la méthode des faux voisins, peuvent être utilisées. En ne testant que les combinaisons de délai(s) pour les régresseurs de cette dimension p, on gagne immédiatement le temps calcul passé à tester des combinaisons dans d'autres espaces de dimension différente. Néanmoins, il ne faut pas oublier que l'estimation de la dimension intrinsèque d_i est généralement réalisée sur base de régresseurs (dimension de corrélation, faux-voisins), et est donc influencée par le délai utilisé dans les régresseurs. Il faut donc malgré tout continuer à tester en même temps les valeurs de délai(s) choisi(s) et la dimension intrinsèque d_i (et donc également la taille des régresseurs p).

5.8 Résumé du chapitre

Les principaux théorèmes liés à la reconstruction d'une série dans son espace de phase sont d'abord rappelés. Les deux paramètres principaux de la reconstruction, en pratique, sont ensuite mentionnés, à savoir la dimension de l'espace de phase et le délai entre composantes des régresseurs. Quelques méthodes permettant d'estimer la dimension intrinsèque d'une série temporelle sont alors présentées.

Les méthodes habituelles de sélection d'un délai unique sont rappelées, à savoir les approches basées sur l'autocorrélation et l'information mutuelle. Un exemple permet d'illustrer les limites de la sélection du délai lorsque cette sélection est réalisée en tenant compte de deux variables seulement. Dans cet exemple, on peut observer la déformation de la reconstruction, qui est par endroit trop dépliée lorsqu'on construit des régresseurs de dimension supérieure à deux en utilisant des multiples d'un délai constant sélectionné en deux dimensions.

Une généralisation du critère d'autocorrélation est alors proposée, afin de prendre en compte plus de deux variables. Cette généralisation permet de sélectionner soit un délai unique, mais directement dans un espace de dimension plus grande que deux, soit plusieurs délais différents. Lors de l'application de ce critère pour la sélection d'un délai unique directement en haute dimension, on a pu souligner la diminution de la valeur du délai lorsque la taille du régresseur augmente. On a également pu observer une limite à ce critère d'ordre supérieur, qui devient difficilement exploitable lorsque l'ordre devient trop important.

Une généralisation de l'approche par information mutuelle est également proposée, afin de prendre en compte plus de deux variables. Cette généralisation est basée sur un estimateur particulier, estimant l'information mutuelle dans un espace de dimension quelconque au moyen de l'algorithme des k-plus-proches-voisins. Cet estimateur permet, lui aussi, de sélectionner un délai unique ou plusieurs délais, directement en haute dimension. Lors de l'application de ce deuxième critère généralisé, on a pu à nouveau souligner la diminution

de la valeur du délai unique sélectionné en haute dimension, lorsque la taille du régresseur p augmente.

Le critère de Distance à la Diagonale, introduit au chapitre précédent, est à nouveau utilisé, sa définition sur base de vecteurs lui permettant d'être employé aisément en haute dimension pour un coût en temps calcul réduit. L'utilisation de ce critère dans le cadre de la sélection d'un délai unique, mais directement en haute dimension, a conduit à la même conclusion que pour les deux critères précédents : la valeur du délai diminue lorsqu'il est sélectionné directement dans un espace de dimension supérieure à deux, la diminution étant de plus en plus nette lorsque la dimension de l'espace augmente. Bien entendu, ce critère est également adapté à la sélection de plusieurs délais.

Les critères d'information mutuelle en haute dimension et de Distance à la Diagonale sont utilisés pour construire des régresseurs comportant plusieurs délais différents au lieu d'un délai unique. Les critères sont également comparés à l'approche classique avec information mutuelle en deux dimensions. En termes de reconstruction dans l'espace de phase, les trois critères fournissent des régresseurs qui, bien que sensiblement différents, sont cohérents entre eux. Les régresseurs sont relativement compacts dans le temps, et les différentes entre variables sélectionnées comme composantes des régresseurs sont limitées d'une méthode à l'autre, de quelques pas dans le temps seulement. En termes de prédiction, les performances obtenues sont variables d'un critère à l'autre, d'une dimension à l'autre : les complexités optimales ne sont pas toujours les mêmes, et les estimations de l'erreur de généralisation obtenues pour ces complexités restent différentes. Toutes ces observations s'expliquent en fonction des reconstructions dans l'espace de phase obtenues en utilisant les différents régresseurs. De plus, on constate expérimentalement qu'aucune méthode ne permet de fournir des régresseurs, et donc une reconstruction, toujours supérieurs à ceux obtenus par les autres méthodes. Notons qu'en termes de temps calcul, il y a sans doute beaucoup mieux à faire que de tester toutes les combinaisons de délais dans toutes les dimensions, approche néanmoins utilisée ici dans un but de comparaison approfondie des méthodes généralisées.

Concluons en rappelant les deux contributions les significatives de ce chapitre. D'une part le constat de la diminution de la valeur du délai, lorsqu'il est unique et choisi directement dans un espace de dimension supérieure à deux, en fonction de l'augmentation de la dimension de l'espace de phase. Cet enseignement, certes qualitatif, est très intéressant puisqu'il se distingue de l'utilisation classique de ces méthodes consistant à ne considérer que deux variables pour sélectionner un délai unique et ensuite prendre des multiples du délai sélectionné. D'autre part, le fait que les différentes méthodes conduisent à des reconstructions dans l'espace de phase qualitativement semblables permet d'affirmer que ces méthodes peuvent être utilisées indifféremment.

Chapitre 6

Conclusion et perspectives

Dans l'introduction, il est expliqué que la prédiction de séries temporelles est bien plus que la "simple" estimation de la valeur future d'une série. En effet, pour pouvoir prédire la valeur future d'une série, il faut construire un modèle mathématique de l'évolution passée de la série temporelle. Et pour obtenir un bon modèle, il est important d'analyser la série temporelle. La problématique de la prédiction couvre donc trois aspects, l'analyse, la modélisation et la prédiction en elle-même. Au cours de cette thèse, différentes contributions ont été apportées pour ces trois aspects, parfois sur le plan théorique, parfois sur un plan plus expérimental.

Au niveau de l'analyse des données, le problème de la sélection du délai a été abordé. Les approches classiques de sélection du délai ont leurs limites, principalement dues aux définitions des critères qui sont proposées en deux dimensions. Pour pouvoir dépasser ces limites, trois critères généralisés, capable de prendre en compte un nombre quelconque de variables, sont proposés : l'autocorrélation d'ordre supérieur, l'information mutuelle en haute dimension et la Distance à la Diagonale. Ces critères ont une double utilité. Ils peuvent être utilisés pour sélectionner un délai unique directement dans un espace de dimension supérieure à deux. Ils peuvent également être utilisés pour sélectionner plusieurs délais entre les différentes variables considérées.

Au niveau de la modélisation, la Double Quantification Vectorielle est décrite, en tant que modèle de prédiction. La Double Quantification Vectorielle est un modèle très souple, capable de prédire à court terme et à long terme, aussi bien des séries scalaires que des séries vectorielles, sans modification particulière de la méthode. Par ailleurs, la sélection de structure par fast-bootstrap est complétée par un test statistique. Ce test permet d'entériner l'hypothèse de linéarité du terme d'optimisme. Enfin, le caractère significatif des méthodes de clustering est confirmé. Cette confirmation du caractère significatif valide l'utilisation de l'algorithme des cartes auto-organisées, considérée ici comme un outil de clustering pour créer des clusters de régresseurs, tant dans le cadre de la Double Quantification Vectorielle que pour les deux méthodes que sont la Double Quantification avec RBFN locaux et la Double Matrice des Transitions.

Au niveau de la prédiction, les capacités de la Double Quantification Vectorielle sont illustrées au travers de plusieurs exemples. La méthode a notamment été classée dans le cadre de la compétition CATS. La prédiction à long terme est abordée de manière théorique avec la preuve de stabilité de la Double Quantification Vectorielle. De plus, les capacités de tout modèle de prédiction à long terme ont été discutées au travers des performances attendues, dans le cadre des approches récursive et directe.

Une série d'enseignements très concrets peuvent être dégagés du développement de tous ces points, détaillés dans le contenu de cette thèse. Résumons ici ces enseignements :

- la Double Quantification Vectorielle est souple d'utilisation et stable à long terme, une preuve théorique de la stabilité est d'ailleurs proposée pour des régresseurs en deux dimensions,
- il n'est pas possible d'affirmer de façon définitive que l'approche directe à long terme est supérieure à l'approche itérative à long terme, ni l'inverse,
- le clustering de séries peut être appliqué à des régresseurs correctement dépliés dans leur espace de phase,
- le(s) délai(s) doi(ven)t idéalement être choisi(s) en tenant compte de toutes les composantes du régresseur,
- quand il est choisi directement dans la dimension de l'espace de phase, la valeur du délai diminue lorsque la dimension de cette espace augmente,
- les méthodes de sélection de délai en dimension supérieure à deux peuvent être utilisées indifféremment, même si les délais choisis sont différents : les régresseurs construits sur base de ces délais sont cohérents d'une méthode à l'autre, peu importe la méthode utilisée.

Au travers des sections et des chapitres de cette thèse, des pistes ont été proposées pour toute une série de questions abordées au cours des travaux de cette thèse et qui restent actuellement ouvertes :

- Est-il possible de généraliser la preuve de stabilité à long terme de la Double Quantification Vectorielle en dimension trois ? Comment ? Et en dimension supérieure ?
- Il semble normal, au moins intuitivement, que les graphes de l'autocorrélation d'ordre supérieur deviennent plats lorsque l'ordre augmente. Comment expliquer plus rigoureusement que ce critère devienne inexploitable lorsque l'ordre augmente ?
- Comment faire pour que la Distance à la Diagonale n'ait pas tendance à sélectionner un délai qui se borne à remplir l'espace de phase par les régresseurs ? Peut-on contraindre ce critère pour qu'il déplie avec plus de précision la structure ? Comment ?
- Comment déterminer une borne supérieure pour les valeurs que peuvent prendre les différents délais en haute dimension ? Une première approche, proposée ici dans le cadre du critère de Distance à la Diagonale, est de considérer la valeur obtenue en deux dimensions comme une borne supérieure pour les dimensions suivantes. Est-il possible de définir une borne plus précise ?

Par ailleurs, au delà de ces questions ouvertes, les contributions de ce travail ouvrent de nouvelles voies. Les sujets abordés posent en effet plusieurs questions intéressantes, tant sur le plan théorique qu'au niveau applicatif. Les aspects les plus intéressants à explorer dans le futur sont sans doute :

- une étude approfondie du critère de Distance à la Diagonale. Sur le plan théorique, il est certainement possible de faire le lien de façon plus formelle entre ce critère géométrique et d'autres approches géométriques proposées dans la littérature et mentionnées dans ce texte. Sur le plan pratique, une première question est bien entendu de limiter la tendance de ce critère à déplier de façon excessive la structure de la série dans son espace de phase. De plus, sachant que la taille du régresseur et la (les) valeur(s) de délai(s) sont liées, il serait intéressant d'observer le comportement de ce critère dans le cadre de méthodologies où il est sélectionné en même temps la taille du régresseur et la valeur du (ou des) délai(s), ces deux questions étant considérées séparément ici pour se concentrer sur le choix du(des) délai(s).
- le développement des méthodes de sélection de délai(s) en haute dimension. A partir du moment où les régresseurs sont généralement de dimension supérieure à deux, il semble tout à fait normal que la sélection du(des) délai(s) soit effectués directement dans un espace de cette dimension. C'est presque une évidence, et pourtant c'est loin d'être le cas en pratique. Les résultats expérimentaux repris dans cette thèse apportent un premier enseignement, même s'il n'est que qualitatif. Il y a sans aucun doute encore beaucoup à faire, non seulement pour développer ces critères, mais encore pour généraliser l'usage de ces approches de sélection de délai(s) directement en haute dimension auprès de ceux qui manipulent en pratique des séries temporelles.
- la poursuite des comparaisons entre les approches de prédiction à long terme, récursive et directe. Il serait en effet très intéressant de déterminer si une approche est supérieure ou non, indépendamment de toute série temporelle et de tout modèle. L'apport de contributions, telles que par exemple la méthode DirRec citée dans ce document, devrait sans doute apporter un élément de réponse pour une question particulièrement difficile mais très intéressante tant sur le plan théorique que pour bien des applications concrètes.

Annexe A

Séries temporelles utilisées

Les séries temporelles utilisées dans le cadre de cette thèse sont rassemblées ici, dans leur ordre d'introduction, au cours des différents chapitres. Pour chaque série, une description brève sur l'origine des données est proposée. Le graphe de la série est à chaque fois proposé, afin d'illustrer la dynamique d'évolution.

A.1 Série SunSpot

La série des taches solaires provient de l'observation du Soleil, dont l'activité varie au cours du temps. Plusieurs phénomènes liés à cette activité sont observables, dont des éruptions ou des taches noires, entre autres. Ces taches noires apparaissent quotidiennement selon un cycle approximatif de onze ans. Plusieurs séries temporelles sont obtenues sur base des observations quotidiennes (144) :

- la série des observations quotidiennes,
- la série des observations mensuelles, obtenue par sommation des observations quotidiennes,
- la série corrigée des observations par mois, obtenue par un lissage utilisant un filtre sur les 13 derniers mois,
- la série des observations annuelles, obtenue par sommation des observations mensuelles.

Dans les différentes expériences, la série des observations mensuelles est utilisée. Cette série est représentée à la Figure A.1. Cette série comporte 3074 valeurs, couvrant l'intervalle de temps entre janvier 1749 et février 2005. Les données sont disponibles en ligne (144).

A.2 Série Santa Fe A

La série Santa Fe A (177) a été proposée dans le cadre d'une compétition d'analyse et de prédiction de séries temporelles organisée en 1991 à l'Institut de Santa Fe. Les données correspondent aux mesures de l'intensité d'un laser NH_3-FIR dans un état chaotique. Une

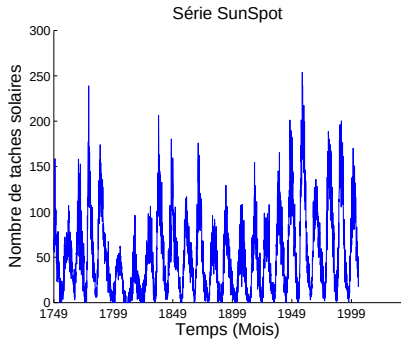

FIG. A.1 – Représentation graphique de la série temporelle des taches solaires SunSpot.

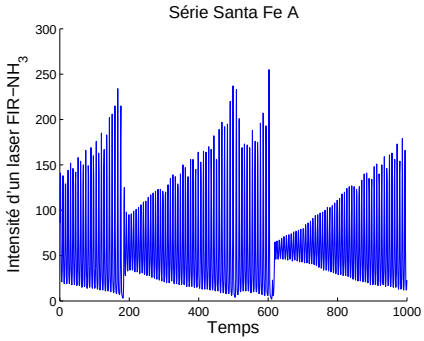

FIG. A.2 – Représentation graphique de la série temporelle de la compétition Santa Fe A.

description complète des données est fournie dans (80). Le but de la compétition était, sur base de 1000 valeurs mises à disposition, de prédire les 100 valeurs suivantes (177). La Figure A.2 propose une représentation graphique de la série.

Une particularité de cette série est que la suite des données ayant servi à la compétition est également disponible, cette deuxième série comptant 9000 valeurs. Les séries de la compétition et des valeurs suivantes sont disponibles en ligne (1).

A.3 Série Polonaise

La série Polonaise contient les valeurs de la consommation électrique en Pologne sur une période d'un peu plus de 8 ans, allant de janvier 1990 à décembre 1998. La série complète compte les 72 000 valeurs de la consommation mesurée heure par heure. Cette série est représentée graphiquement à la Figure A.3.

A.4 Série CATS

La série CATS est une série temporelle proposée dans le cadre de la compétition de prédiction CATS, organisée en 2004 lors de la conférence IJCNN (104). 4900 données

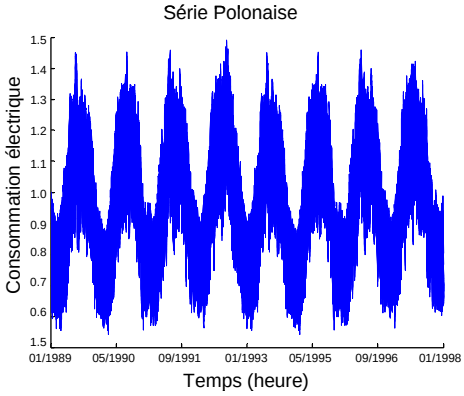

FIG. A.3 – Représentation graphique de la série Polonaise des consommations électriques.

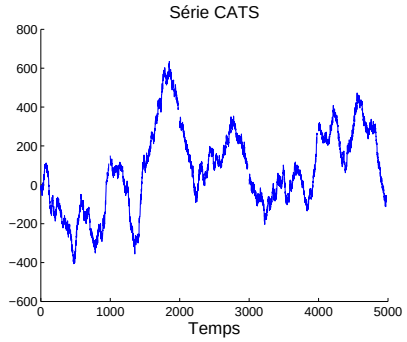

FIG. A.4 – Représentation graphique de la série entière de la Compétition CATS.

étaient mises à disposition des participants, parmi lesquelles figuraient 100 valeurs manquantes. Le but de la compétition était de prédire ces 100 valeurs manquantes. La Figure A.4 présente une représentation graphique des 4900 valeurs de la série.

En pratique, cinq groupes de 20 valeurs manquantes devaient être complétés, les 4900 valeurs de la série formant cinq suites de 980 données suivies d'un bloc de 20 valeurs manquantes. Un agrandissement de la série autour des cinq blocs de valeurs manquantes est proposé à la Figure A.5.

Signalons que, bien que la compétition soit terminée, il n'y a toujours aucune information disponible à propos de l'origine des données. Bien qu'il semble que la série soit artificielle, la relation ayant permis de générer les valeurs de la série n'a toujours pas été rendue publique.

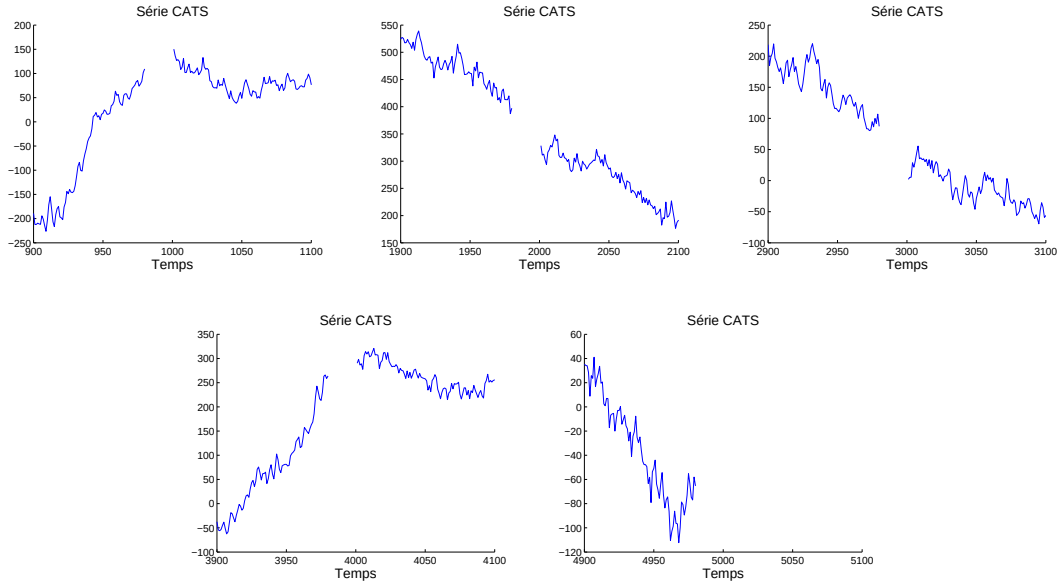

FIG. A.5 – Représentation graphique des cinq blocs de valeurs manquantes de la série de la Compétition CATS. De gauche à droite et de haut en bas : les blocs de valeurs manquantes 1 à 5.

A.5 Série Lorenz

La série Lorenz a été obtenue en intégrant les équations différentielles du système de Lorenz (3; 88; 89) :

$$\begin{aligned} \dot{x}_t &= 10(y_t - x_t), \\ \dot{y}_t &= 28x_t - y_t - x_t z_t, \\ \dot{z}_t &= x_t y_t - \frac{8}{3} z_t. \end{aligned}$$

Sur base de ces équations, 4000 données ont été générées en utilisant un pas d'intégration de 0.015 à partir des conditions initiales $x_0 = 0.1$, $y_0 = 0.5$ et $z_0 = -0.2$. Toutefois, les conditions initiales sont relativement peu importantes pour ce système chaotique, les 2000 premières données ayant été enlevées de la série temporelle. Le but de cette manipulation est de laisser la trajectoire de la série converger vers son attracteur, après passage de l'état transitoire fonction des conditions initiales de la série. Une représentation graphique de la série est proposée à la Figure A.6.

A.6 Série MarcheAléatoire

La série MarcheAléatoire est obtenue sur base de la relation habituelle définissant une marche aléatoire :

$$x_{t+1} = x_t + \varepsilon_t,$$

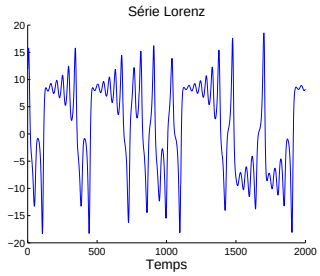

FIG. A.6 – Représentation graphique de la série Lorenz.

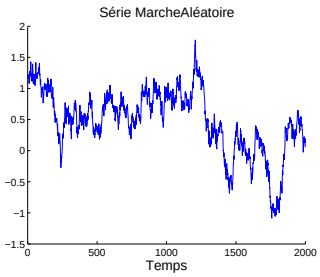

FIG. A.7 – Représentation graphique de la série MarcheAléatoire.

où ε_t est un bruit blanc Gaussien, de moyenne nulle. Pour cette série, 2000 données ont été générées. La valeur est $x_0 = 1.235$. Ces données sont représentées graphiquement à la Figure A.7.

A.7 Série Rossler

La série Rossler a été obtenue en intégrant les équations différentielles du système de Rossler (3; 89) :

$$\begin{aligned} \dot{x}_t &= -(y_t + z_t), \\ \dot{y}_t &= x_t + 0.2y_t, \\ \dot{z}_t &= 0.2 + z_t(x_t - 5.7). \end{aligned}$$

Sur base de ces équations, 4000 données ont été générées en utilisant un pas d'intégration de 0.075, à partir des conditions initiales $x_0 = 1$, $y_0 = 1$ et $z_0 = 2$. Les conditions initiales n'ont ici encore que peu d'influence, le système d'équations décrivant un processus chaotique. Tout comme pour la série Lorenz, les 2000 premières données ont également été enlevées ici, afin de laisser la trajectoire de la série converger vers son attracteur. Une représentation graphique de la série est proposée à la Figure A.8.

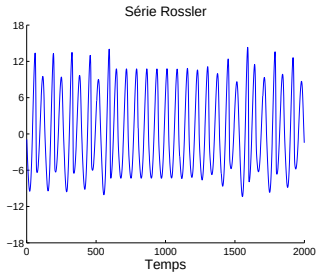

FIG. A.8 – Représentation graphique de la série Rossler.

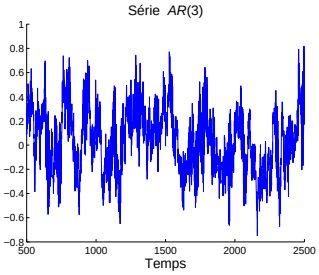

FIG. A.9 – Représentation graphique de la série $AR(3)$.

A.8 Série AR3

La série $AR(3)$ est générée artificiellement sur base de la relation suivante :

$$x_t = a_1 * x_{t-1} + a_2 * x_{t-2} + a_3 * x_{t-3} + \varepsilon_t.$$

Les coefficients a_1, a_2 et a_3 ont été choisi aléatoirement suivant une distribution normale de moyenne nulle et de variance unitaire. ε_t a également été généré suivant une distribution normale de moyenne nulle mais de variance 0.1 cette fois. 2500 valeurs ont été générées. Pour éviter tout biais dû à l'initialisation, les 500 premières valeurs ont été retirées de la série. Une représentation graphique des 2000 données restantes est proposée à la Figure A.9.

Liste des publications

[gs-1] Amaury Lendasse, Vincent Wertz, Geoffroy Simon and Michel Verleysen. Fast Bootstrap for Model Structure Selection. In *Proceedings of 22th Benelux Meeting on Systems, Control*, page 81, Lommel, Belgium, March 2003. 6

[gs-2] Geoffroy Simon, Amaury Lendasse, Vincent Wertz and Michel Verleysen. Fast Approximation of the Bootstrap for Model Selection. In *Proceedings of ESANN'2003, European Symposium on Artificial Neural Networks*, pages 475-480, Bruges, Belgium, April 2003. 6

[gs-3] Geoffroy Simon, Amaury Lendasse and Michel Verleysen. Bootstrap for Model Selection : Linear Approximation of the Optimism. In J. Mira, J.R. Alvarez, editors, *Computational Methods in Neural Modeling, Proceedings of International Work-Conference on Artificial, Natural Neural Networks, IWANN2003*, volume 2686 of *Lecture Notes in Computer Science*, pages **I**182-**I**189, Mao, Menorca (Spain), June 2003. 6

[gs-4] Michel Verleysen, Damien François, Geoffroy Simon and Vincent Wertz. On the effects of dimensionality on data analysis with neural networks. In J. Mira, J.R. Alvarez, editors, *Artificial Neural Nets Problem solving methods, Proceedings of International Work-Conference on Artificial, Natural Neural Networks, IWANN2003*, volume 2687 of *Lecture Notes in Computer Science*, pages **II**105-**II**112, Mao, Menorca (Spain), June 2003.

[gs-5] Geoffroy Simon, Amaury Lendasse, Marie Cottrell and Michel Verleysen. Long-Term Time Series Forecasting using Self-Organizing Maps : the Double Vector Quantization Method. In *Proceedings of ANNPR 2003, Artificial Neural Networks in Pattern Recognition*, pages 8-14, Florence, Italy, September 2003. 6

[gs-6] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort and Michel Verleysen. Double SOM for Long-term Time Series Prediction. In *Proceedings of WSOM'03, Workshop on Self-Organizing Maps*, pages 35-40, Kitakyushu, Japan, September 2003. 6

[gs-7] Simon Dablemont, Geoffroy Simon, Amaury Lendasse, Alain Ruttiens, Francois Blayo and Michel Verleysen. Time series forecasting with SOM, local non-linear models - Application to the DAX30 index prediction. In *Proceedings of WSOM'03, Workshop on Self-Organizing Maps*, pages 340-345, Kitakyushu, Japan, September 2003. 6

[gs-8] Simon Dablemont, Geoffroy Simon, Amaury Lendasse, Alain Ruttiens and Michel Verleysen. Prédiction de séries temporelles financières par double carte de Kohonen et modèles RBFNs locaux - Application à la prédiction de l'indice boursier DAX30. In *Proceedings of ACSEG'2003, 10ème Rencontre Internationale sur les Approches Connexionnistes en Sciences Economiques et de Gestion*, pages 153-164, Nantes, France, November 2003. 6

[gs-9] Amaury Lendasse, Geoffroy Simon, Vincent Wertz and Michel Verleysen. Fast bootstrap for Least-Square Support Vector Machines. In *Proceedings of ESANN'2004, European Symposium on Artificial Neural Networks*, pages 525-530, Bruges, Belgium, April 2004. 6

[gs-10] Geoffroy Simon, John A. Lee, Marie Cottrell and Michel Verleysen. Double Quantization Forecasting Method for Filling Missing Data in the CATS Time Series. In *Proceedings of IJCNN 2004, International Joint Conference on Neural Networks*, pages 1635-1640, Budapest, Hungary, July 2004. 6

[gs-11] Amaury Lendasse, Vincent Wertz, Geoffroy Simon and Michel Verleysen. Fast Bootstrap applied to LS-SVM for Long Term Prediction of Time Series. In *Proceedings of IJCNN 2004, International Joint Conference on Neural Networks*, pages 705-710, Budapest, Hungary, July 2004. 6

[gs-12] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort and Michel Verleysen. Double quantization of the regressor space for long-term time series prediction : Method, proof of stability. *Neural Networks*, 17(8-9) :1169-1181, 2004. 6

[gs-13] Amaury Lendasse, Geoffroy Simon, Vincent Wertz and Michel Verleysen. Fast bootstrap methodology for model selection. *Neurocomputing*, 64 :161-181, 2005. 6

[gs-14] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort and Michel Verleysen. Time series forecasting : Obtaining long term trends with self-organizing maps. *Pattern Recognition Letters*, 26(12) :1795-1808, 2005. 6

[gs-15] Geoffroy Simon, John A. Lee and Michel Verleysen. On the need of unfolding preprocessing for time series clustering. In *Proceedings of WSOM'05, Workshop on Self-Organizing Maps*, pages 251-258, Paris, France, September 2005. 6

[gs-16] Didier Catteau, Geoffroy Simon, Walid Ben Omrane and Michel Verleysen. Using the Self-Organizing Maps to prove empirically the market inefficiency : Evidence from Paris Stock Exchange. In *Proceedings of ACSEG'2005, 12ème Rencontre Internationale sur les Approches Connexionnistes en Sciences Economiques et de Gestion*, pages 153-164, Aix-en-Provence, France, November 2005. 6

[gs-17] Geoffroy Simon and Michel Verleysen. Lag Selection for Regression Models Using High-Dimensional Mutual Information. In *Proceedings of ESANN'2006, European Symposium on Artificial Neural Networks*, pages 395-400, Bruges, Belgium, April 2006. 6

[gs-18] Geoffroy Simon, John Aldo Lee and Michel Verleysen. Unfolding preprocessing for meaningful time series clustering. *Neural Networks*, 19(6-7) :277-888, 2006. 6

[gs-19] Geoffroy Simon, John A. Lee and Michel Verleysen. High-dimensional delay selection for regression models with mutual information, distance-to-diagonal criteria. Article in press, *Neurocomputing*, 2007. 6

[gs-20] Geoffroy Simon, John Aldo Lee, Marie Cottrell and Michel Verleysen. Forecasting the CATS benchmark with the Double Vector Quantization method. Article in press, *Neurocomputing*, 2007. 6

Références

[1] 1994. The time series of the Santa Fe competition are available at `http://www-psych.stanford.edu/~andreas/Time-Series/SantaFe.html`. 132

[2] March 2006. On-line bibliography of SOM publications, `http://www.cis.hut.fi/mpolla/sombibliography/`. 10, 32

[3] H. D. I. Abarbanel. *Analysis of Observed Chaotic Data*. Springer-Verlag, New-York, 1997. 96, 100, 101, 134, 135

[4] Huseyin Abut. *Vector Quantization*. IEEE Press, New York, 1990. 9

[5] K. T. Alligood, T. D. Sauer, and J. A. Yorke. *Chaos : An Introduction to Dynamical Systems*. Springer-Verlag, Berlin, 1996. 98

[6] Peter Angelovič. Time series prediction using rsom and local models. In *Proceedings of Informatics and Information Technology Student Research Conference*, pages 27–34, Bratislava, Slovakia, April 2005. 34

[7] S. Astakhov, P. Grassberger, A. Kraskov, and H. Stögbauer. Milca - mutual information least-dependent component analysis, 2004. software implementation package available from `http://www.klab.caltech.edu/~kraskov/MILCA/`. 108

[8] Fernando Bação, Victor Sousa Lobo, and Marco Painho. Self-organizing maps as substitutes for k-means clustering. In Vaidy S. Sunderam, G. Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Computational Science, Proceedings of 5th International Conference on Computational Science, ICCS 2005, Part III*, volume 3516 of *Lecture Notes in Computer Science*, pages 476–483, Atlanta, GA, May 2005. 91

[9] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5 :537–550, 1994. 111

[10] Nabil Benoudjit and Michel Verleysen. On the kernel widths in radial-basis function networks. *Neural Processing Letters*, 18(2) :139–154, 2003. 14, 15

[11] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995. 8, 13, 15

[12] George Box, Gwilym M. Jenkins, and Gregory Reinsel. *Time series analysis ; forecasting and control*. Prentice Hall, Upper Saddle River, N.J., third edition, 1994. 7

[13] Pierre Bremaud. *Markov Chains*. Springer-Verlag, New York, 2001. 44

[14] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, New York, 1996. 7

[15] D. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2 :321–355, 1988. 8

[16] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2 :121–167, 1998. 8

[17] T. Buzug and G. Pfister. Optimal delay time and embedding dimension for delay-time coordinates by analysis of the global static and local dynamical behavior of strange attractors. *Physical Review A*, 45 :7073–7084, 1992. 84

[18] F. Camastra and A. M. Colla. Neural short-term prediction based on dynamics reconstruction. *Neural Processing Letters*, 9(1) :45–52, 1999. 63, 96

[19] Liangyue Cao. Practical method for determining the minimum embedding dimension of a scalar time series. *Physica D*, 110 :43–50, 1997. 97

[20] Otàvio Augusto S. Carpinteiro. A hierarchical self-organizing map model for sequence recognition. *Neural Processing Letters*, 9(3) :209–220, 1999. 33

[21] M. Casdagli, S. Eubank, J.D. Farmer, and J. Gibson. State space reconstruction in the presence of noise. *Physica D*, 51 :52–98, 1991. 95, 97

[22] Didier Catteau, Geoffroy Simon, Walid Ben Omrane, and Michel Verleysen. Using the self-organizing maps to prove empirically the market inefficiency : Evidence from paris stock exchange. In *ACSEG 2005, 12ème Rencontre Internationale sur les Approches Connexionnistes en Sciences Economiques et de Gestion*, Aix-en-Provence, France, November 2005. 72

[23] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing kernel parameters for support vector machines. *Machine Learning*, 46(1-3) :131–159, 2002. 8

[24] G. Chappell and J. Taylor. The temporal kohonen map. *Neural Networks*, 6 :441–445, 1993. 33

[25] Jason R. Chen. Making subsequence time series clustering meaningful. In *Proceedings of the 5th IEEE International Conference on Data Mining - ICDM 2005*, pages 114–121, Houston, TX, November 2005. 91

[26] T. Chordia, R. Roll, and A. Subrahmanyam. Evidence on the speed of convergence to market efficiency. *Journal of Financial Economics*, 76 :271–292, 2005. 72

[27] Charles K. Chui. *An Introduction to Wavelets*, volume 1. Academic Press, London, 1992. 8

[28] Marie Cottrell, Eric de Bodt, and Philippe Grégoire. Simulating interest rate structure evolution on a long term horizon : A kohonen map application. In Pasadena World Scientific Ed., editor, *Proceedings of Neural Networks in The Capital Markets*, pages 162–174, Californian Institute of Technology, 1996. 34

[29] Marie Cottrell, Jean-Claude Fort, and Gilles Pagès. Theoretical aspects of the som algorithm. *Neurocomputing*, 21(1-3) :119–138, 1998. 10

[30] Marie Cottrell, B. Girard, and Patrick Rousset. Long term forecasting by combining kohonen algorithm and standard prevision. In W. Gerstner, A. Germond, M. Hasler, and J. D. Nicoud, editors, *Proceedings of ICANN'97 International Conference on Artificial Neural Networks*, volume 1327 of *Lecture Notes in Computer Science*, pages 993–998. Springer, Lausanne, Switzerland, 1997. 34, 59

[31] Marie Cottrell, B. Girard, and Patrick Rousset. Forecasting of curves using a kohonen classification. *Journal of Forecasting*, 17 :429–439, 1998. 34

[32] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, New York, 1991. 99, 108

[33] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge, 2000. 8

[34] Simon Dablemont, Geoffroy Simon, Amaury Lendasse, Alain Ruttiens, François Blayo, and Michel Verleysen. Time series forecasting with som and local non-linear models - application to the dax30 index prediction. In *Proceedings of Workshop on Self-Organizing Maps WSOM'03*, Kitakyushu, Japan, Septembre 2003. 34, 68, 69

[35] Simon Dablemont, Geoffroy Simon, Amaury Lendasse, Alain Ruttiens, and Michel Verleysen. Prédiction de séries temporelles financières par double carte de kohonen et modèles rbfns locaux - application à la prédiction de l'indice boursier dax30. In *Proceedings ACSEG 2003, 10ème Rencontre Internationale sur les Approches Connexionnistes en Sciences Economiques et de Gestion*, pages 153–164, Nantes, France, November 2003. 68, 69

[36] Guilherme de A. Barreto and Aluizio F. R. Araújo. Time in self-organizing maps : An overview of models. *International Journal of Computer Research*, 10(2) :139–179, 2001. 34

[37] Guilherme de A. Barreto and Aluizio F. R. Araújo. Identification and control of dynamical using the self-organizing map. *IEEE Transactions on Neural Networks*, 15(5) :1244–1259, 2004. 34

[38] Guilherme de A. Barreto, Aluizio F. R. Araújo, and Stefan C. Kremer. A taxonomy for spatiotemporal connectionist networks revisited : The unsupervised case. *Neural Computation*, 15(6) :1255–1320, 2003. 34

[39] Eric de Bodt, Marie Cottrell, Patrick Letremy, and Michel Verleysen. On the use of self-organizing maps to accelerate vector quantization. *Neurocomputing*, 56 :187–203, 2003. 32

[40] Carl de Boor. *A Practical Guide to Splines*. Springer Verlag, New York, 1978. 8

[41] Guido Deboeck and Teuvo Kohonen. *Visual Explorations in Finance with self-organizing maps*. Springer-Verlag, Berlin, 1998.

[42] Pierre Demartines. *Analyse de Données par Réseaux de Neurones Auto-Organisés*. PhD thesis, Institut Polytechnique de Grenoble, France, 1992. 96

[43] Anne Denton. Density-based clustering of time series subsequences. In *Proceedings of 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Seattle, WA, 2004. 90

[44] Pedro Domingos. A unified bias-variance decomposition and its applications. In *Proceedings 17th International Conference on Machine Learning*, pages 231–238. Morgan Kaufmann, San Francisco, CA, 2000. 19

[45] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2000. 8

[46] B. Efron and R. Tibshirani. Cross-validation and the bootstrap : Estimating the error rate of a prediction rule, 1995. Technical Report (TR-477), Dept. of Statistics, Stanford University. 21

[47] Bradley Efron and Robert J. Tibshirani. *An introduction to the Bootstrap*. Chapman and Hall, New York, 1993. 21, 23

[48] Walters Enders. *Applied Econometric Time Series*. Wiley, New York, 1995. 8

[49] N. R. Euliano and J. C. Principe. Spatio-temporal self-organizing feature maps. In *Proceedings of the 1996 IEEE International Conference on Neural Networks - ICNN'96*, volume 4, pages 1900–1905, New York, NY, 1996. 33

[50] N. R. Euliano and J. C. Principe. A spatio-temporal memory based on SOMs with activity diffusion. In S. Oja and E. Kaski, editors, *Kohonen Maps*, pages 253–266. Elsevier, Amsterdam, 1999. 33

[51] G. Fayolle, V. A. Malyshev, and M. V. Menshikov. *Topics in constructive theory of countable Markov chains*. Cambridge University Press, 1995. 45, 50

[52] E. Fiesler and R. Beale. *Handbook Of Neural Computation*. IOP Publishing Ltd and Oxford University Press, 1997. 8

[53] John A. Flanagan. Sufficient conditions for self-organisation in the one-dimensional som with reduced width neighbourhood. *Neurocomputing*, 21(1-3) :51–60, 1998. 10

[54] Jean-Claude Fort. Som's mathematics. *Neural Networks*, 19(6-7) :812–816, 2005. 10

[55] Jean-Claude Fort. Som's mathematics. In *Proceedings of 5th Workshop on Self-Organizing Maps*, pages 579–586, Paris, France, September 2005.

[56] Klaus Fraedrichb and Risheng Wangb. Estimating the correlation dimension of an attractor from noisy and small datasets based on re-embedding. *Physica D*, 65(4) :373–398, 1993. 96

[57] Philip Hans Franses. *Time series models for business and economic forecasting*. Cambridge University Press, Cambridge, 1998. 8

[58] A. M. Fraser and H. L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33 :1134–1140, 1986. 100, 101

[59] T. Gautama, D. P. Mandic, and M. M. Van Hulle. The delay vector variance method for detecting determinism and nonlinearity in time series. *Physica D*, 190 :167–176, 2004. 105

[60] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4 :1–58, 1992. 18, 19

[61] John Aldo Lee Geoffroy Simon and Michel Verleysen. Unfolding preprocessing for meaningful time series clustering. *Neural Networks*, 19(6-7) :277–888, 2006. 90

[62] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression*. Kluwer Academic Press, 1992. 9

[63] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7 :219–269, 1995. 8

[64] P. Grassberger and I. Procaccia. Measuring the strangeness of strange attractors. *Physica D*, 9 :189–208, 1983. 62, 63, 96

[65] Robert M. Gray. Vector quantization. *IEEE ASSP Mag.*, 1 :4–29, 1984. 9, 78

[66] Robert M. Gray and D. L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44 :2325–2384, 1998. 9

[67] William H. Greene. *Econometric analysis*. Prentice Hall, Upper Saddle River, N.J., third edition, 1997. 8

[68] S. Grossberg. Competitive learning : from interactive activation to adaptive resonance. *Cognitive Science*, 11 :23–63, 1987. 9

[69] Martin T. Hagan and Mohammad B. Menhaj. Training feedforward networds with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6) :989–993, 1994. 8

[70] M. Hagenbuchner, A. Sperduti, and A. Tsoi. A self-organizing map for adaptive processing of structured data. *IEEE Transactions on Neural Networks*, 14(3) :491–505, 2003. 34

[71] Barbara Hammer, Alessio Micheli, Nicolas Neubauer, Alessandro Sperduti, and Marc Strickert. Self-organizing maps for time series. In *Proceedings of WSOM 2005*, pages 115–122, Paris, France, September 2005. 33

[72] Barbara Hammer, Alessio Micheli, Alessandro Sperduti, and Marc Strickert. Recursive self-organizing network models. *Neural Networks*, 17(8-9) :1061–1086, 2004. 34

[73] Barbara Hammer, Alessio Micheli, Marc Strickert, and Alessandro Sperduti. A general framework for unsupervised processing of structured data. *Neurocomputing*, 57 :3–35, 2004. 34

[74] Barbara Hammer and Thomas Villmann. Mathematical aspects of neural networks. In *Proceedings of ESANN'2003, European Symposium on Artificial Neural Networks*, pages 59–72, Bruges, Belgium, April 2003. 10

[75] T. S. Han. Multiple mutual informations and multiple interactions in frequency data. *Information and Control*, 46(1) :26–45, 1980. 110

[76] Mohamad H. Hassoun. *Fundamentals of artificial neural networks*. MIT Press, 1995. 8

[77] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer, New York, 2001. 8, 78

[78] Simon Haykin. *Neural Networks : A Comprehensive Foundation*. Prentice Hall, Upper Saddle River, N.J., 1998. 8

[79] R. Horowitz and L. Alvarez. Convergence properties of self-organizing neural networks. In *Proceedings of the IEEE 1995 American Control Conference*, volume 2, pages 1339–1344, Evanston, IL, USA, 1995. American Autom Control Council. 10

[80] Udo Hübner, Carl-Otto Weiss, Neal Broadus Abraham, and Dingyuan Tang. Lorenz-like chaos in NH_3-FIR lasers (data set a). In Andreas S. Weigend and Neil A. Gershenfeld, editors, *Time Series Prediction : Forecasting the Future and Understanding the Past*, pages 73–104. Addison-Wesley, Reading, M.A., 1994. 132

[81] Tsuyoshi Ide. Why does subsequence time-series clustering produce sine waves ? In *Accepted in 17th European Conference on Machine Learning and 10th European Conference on Principles and Practice of Knowledge Discovery in Databases - ECML PKDD 2006*, Berlin, Germany, September 2006. 91

[82] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988. 9

[83] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering : a review. *ACM Computing Surveys*, 31(3) :264–323, 1999. 77, 78

[84] Daniel L. James and Risto Miikkulainen. Sardnet : a self-organizing feature map for sequences. In G. Tesauro, D. Touretzky, and T. Leen, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 7, pages 577–584. MIT Press, Cambridge, M.A., 1995. 33

[85] G. James and T. Hastie. Generalizations of the bias/variance decomposition for prediction error, 1997. Technical Report, Department of Statistics, Stanford University, Stanford, CA, available at `http://www-stat.stanford.edu/~hastie/pub.htm`. 19

[86] Y. Ji, J. Hao, N. Reyhani, and A. Lendasse. Direct and recursive prediction of time series using mutual information selection. In *Computational Intelligence and Bioinspired Systems, Proceedings of International Workshop on Artificial Neural Networks - IWANN 2005*, volume 3512 of *Lecture Notes in Computer Science*, pages 1010–1017. Springer, 2005. 111

[87] Jari Kangas. *On the Analysis of Pattern Sequences by SelfOrganizing Maps*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 1994. 33

[88] H. Kantz and T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge Nonlinear Science Series 7. Cambridge University Press, Cambridge, second edition, 2004. 17, 67, 76, 95, 96, 98, 101, 134

[89] D.T. Kaplan and L. Glass. *Understanding Nonlinear Dynamics*. Springer-Verlag, New-York, 1995. 98, 134, 135

[90] Michael J. Kearns, Yishay Mansour, Andrew Y. Ng, and Dana Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27(1) :7–50, 1997. 21

[91] M. B. Kennel, R. Brown, and H. D. I. Abarbanel. Determining minimum embedding dimension using a geometrical construction. *Physical Review A*, 45 :3403–3411, 1992. 96

[92] Matthew B. Kennel and Henry D. I. Abarbanel. False neighbors and false strands : A reliable minimum embedding dimension algorithm. *Physical Review E*, 66(5) :059903, 2002. 97

[93] E. Keogh, J. Lin, and W. Truppel. Clustering of time series subsequences is meaningless : Implications for past and future research. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 115–122, Melbourne, FL., 2003.

[94] Eamonn Keogh and Jessica Lin. Clustering of time-series subsequences is meaningless : implications for previous and future research. *Knowledge and Information Systems*, 8 :154–177, 2005. 77, 78, 79, 80, 81, 82, 90, 91

[95] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of 14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1137–1145, Montréal, Canada, 1995. 20, 21

[96] Ron Kohavi and David H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In Lorenza Saitta, editor, *Machine Learning : Proceedings of the Thirteenth International Conference*, pages 275–283. Morgan Kaufmann, 1996. 19

[97] Teuvo Kohonen. *Self-organizing Maps*. Springer Series in Information Sciences. Springer, Berlin, 2nd edition, 1995. 8, 10

[98] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3) :1–6, 1998. 10

[99] T. Koskela, M. Varsta, J. Heikkonen, and K. Kaski. Recurrent SOM with local linear models in time series prediction. In *Proceedings of ESANN'1998, European Symposium on Artificial Neural Networks*, pages 167–172, Bruges, Belgium, April 1998. 34

[100] T. Koskela, M. Varsta, J. Heikkonen, and K. Kaski. Temporal sequence processing using recurrent SOM. In *Proceedings of the 2nd International Conference on Knowledge-Based Intelligent Engineering Systems*, volume 1, pages 290–297, Adelaide, Australia, 1998. 33

[101] A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Physical Review E*, 69(6) :066138, 2004. 108, 109, 110, 111

[102] Amaury Lendasse. *Analyse et prédiction de séries temporelles par méthodes non linéaires*. PhD thesis, Département d'Ingénierie Mathématique, Faculté des Sciences Appliquées, Université catholique de Louvain, Belgique, 2003. 17, 21

[103] Amaury Lendasse, Marie Cottrell, Vincent Wertz, and Michel Verleysen. Prediction of electric load using kohonen maps - application to the polish electricity consumption. In *Proceedings of American Control Conference*, pages 3684–3689, Anchorage, USA, 2002. 59

[104] Amaury Lendasse, Erkki Oja, Olli Simula, and Michel Verleysen. Time series prediction competition : The cats benchmark. In *Proceedings of IJCNN'2004, International Joint Conference on Neural Networks*, pages 1615–1620, Budapest, Hungary, July 2004. 56, 62, 65, 132

[105] Amaury Lendasse, Geoffroy Simon, Vincent Wertz, and Michel Verleysen. Fast bootstrap for least-square support vector machines. In *Proceedings of ESANN'2004, European Symposium on Artificial Neural Networks*, pages 525–530, Bruges, Belgium, April 2004. 27

[106] Amaury Lendasse, Geoffroy Simon, Vincent Wertz, and Michel Verleysen. Fast bootstrap methodology for model selection. *Neurocomputing*, 64 :161–181, 2005. 23, 27

[107] Amaury Lendasse, Michel Verleysen, Eric de Bodt, Philippe Grégoire, and Marie Cottrell. Forecasting time-series by kohonen classification. In *Proceedings of ESANN'1998, European Symposium on Artificial Neural Networks*, pages 221–226, Bruges, Belgium, 1998. 34

[108] Amaury Lendasse, Vincent Wertz, and Michel Verleysen. Model selection with cross-validations and bootstraps : Application to time series prediction with rbfn models. In O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, editors, *Artificial Neural Networks and Neural Information Processing, Proceedings of Joint International Conference on Artificial Neural Networks - ICANN/ICONIP 2003*, volume 2714 of *Lecture Notes in Computer Science*, pages 573–580. Springer-Verlag, Istanbul, Turkey, June 2003. 21

[109] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. 2006. The book is not yet published, drafts are available at `http://www.oberlin.edu/markov/`. 44

[110] T. Warren Liao. Clustering of time series data - a survey. *Pattern Recognition*, 38(11) :1857–1874, 2005. 78

[111] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communication*, 28 :84–95, 1980. 9

[112] H. R. Lindman. *Analysis of variance in complex experimental designs*. W. H. Freeman and Co, San Francisco, C.A., 1974. 25

[113] Lennart Ljung. *System Identification, Theory for the user*. Prentice Hall, Upper Saddle River, N.J., second edition, 1999. 7, 20

[114] Xiaodong Luo and Michael Small. Geometric measures of redundance and irrelevance tradeoff exponent to choose suitable delay times for continuous systems. *ArXiv Nonlinear Sciences e-prints*, 2003. 97

[115] R. Mañé. On the dimension of the compact invariant sets of certain non-linear maps. In *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 230–242. Springer, 1981. 95

[116] T. Martinetz, S. Berkovich, and K. Schulten. Neural-gas network for vector quantization and its application to time-series prediction. *IEEE Transactions on Neural Networks*, 4(4) :558–569, 1993. 35

[117] Marshall R. III Mayberry and Risto Miikkulainen. Using a sequential SOM to parse long-term dependencies. In *Proceedings of the 21st Annual Meeting of the Cognitive Science Society (COGSCI-99)*, pages 367–372, Vancouver, Canada, 1999. Erlbaum. 33

[118] W. J. McGill. Multivariate information transmission. *Psychometrika*, 19(2) :97–116, 1954. 110

[119] R. G. Miller. *Beyond ANOVA : Basics of Applied Statistics*. Chapman and Hall, 1997. 25

[120] Isaque Q. Monteiro, Guilherme A. Barreto, and Patrícia V. Nascimento. Dynamic LVQ models for classification of spatiotemporal patterns. In *Proceedings of the VII Brazilian Congress on Neural Networks (CBRN'05)*, pages 1–6, Natal, Brazil, 2005. 35

[121] John Moody. Prediction risk and architecture selection for neural networks. In V. Cherkassky, J. H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks : Theory and Pattern Recognition Applications*. Springer, NATO ASI Series F, 1994. 19

[122] John E. Moody and Christian Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1 :281–294, 1989. 8, 13

[123] M. Mozer. Neural network architectures for temporal pattern processing. In A. Weigend and N. Gershenfeld, editors, *Time series prediction : Forecasting the future and understanding the past*, pages 243–264. Addison-Wesley, Reading, M.A., 1993. 33

[124] James R. Norris. Markov chains. In R. Gill, B. D. Ripley, S. Ross, B. W. Silverman, and M. Stein, editors, *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1997. 44

[125] Mark J. Orr. Optimising the widths of radial basis functions. In *Proceedings of Vth Brazilian Symposium on Neural Networks*, Belo Horizonte, Brazil, December 1998. 14

[126] S. Osowski, K. Siwek, and L. Tran Haoi. Short term load forecasting using neural networks. In *Proceedings of III Ukrainian-Polish Workshop*, pages 72–77, Aluszta, Poland, 2001. 59

[127] A. M. Albano A. Passamante and Mary Eileen Farrell. Using higher-order correlations to define an embedding window. *Physica D*, 54 :85–97, 1991. 105

[128] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9) :1481–1497, 1990. 8, 13

[129] N. Reyhani, J. Hao, Y. Ji, and A. Lendasse. Mutual information and gamma test for input selection. In *Proceedings of European Symposium on Artificial Neural Networks - ESANN2005*, pages 503–508, Bruges, Belgium, April 2005. 111

[130] Helge Ritter. Self-organizing maps on non-euclidian spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–110. MIT Elsevier, Amsterdam, 1999. 34

[131] Helge Ritter and K. Schulten. Convergence properties of kohonen's topology conserving mappings : Fluctuations, stability and dimension selection. *Biological Cybernetics*, 60 :59–71, 1988. 10

[132] Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3) :400–407, 1951. 10, 12

[133] Frank Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6) :386–408, 1958. 8

[134] Michael T. Rosenstein, James J. Collins, and Carlo J. De Luca. Reconstruction expansion as a geometry-based framework for choosing proper delay times. *Physica D*, 73 :82–98, 1994. 84, 98

[135] Patrick Rousset. *Applications des algorithmes d'auto-organisation à la classification et à la prévision.* PhD thesis, Université Paris I - Panthéon Sorbonne, Paris, France, 1999. 34

[136] Patrick Rousset. Curve forecast with the som algorithm : Using a tool to follow the time on a kohonen map. In *Proceedings of ESANN'2000, European Symposium on Artificial Neural Networks*, pages 353–358, Bruges, Belgium, April 2000. 34

[137] D. E. Rumelhart and D. Zipser. Feature discovery by competitive learning. *Cognitive Science*, 9 :75–112, 1985. 9

[138] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, volume I, pages 318–362. MIT Press, Cambridge, 1986.

[139] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323 :533–536, 1986. 8

[140] T. Sauer, J. Yorke, and M. Casdagli. Embeddology. *Journal of Statistical Physics*, 65 :579–616, 1991. 93, 95

[141] Bernhard Schölkopf, Christopher J.C. Burges, and Alexander J. Smola, editors. *Advances in Kernel Methods : Support Vector Learning*. MIT Press, Cambridge, M.A., 1999. 8

[142] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, Cambridge, M.A., 2002. 8

[143] Thomas Schreiber and Andreas Schmitz. Discrimination power of measures for nonlinearity in a time series. *Physical Review E*, 55 :5443–5447, 1997. 105

[144] SIDC, 2005. RWC Belgium, World Data Center for the Sunspot Index, Royal Observatory of Belgium, `http://sidc.oma.be/sunspot-data/index.php`. 2, 131

[145] Geoffroy Simon, John A. Lee, Marie Cottrell, and Michel Verleysen. Double quantization forecasting method for filling missing data in the cats time series. In *Proceedings of IJCNN'2004, International Joint Conference on Neural Networks*, pages 1635–1640, Budapest, Hungary, July 2004. 62, 64

[146] Geoffroy Simon, John Aldo Lee, and Michel Verleysen. On the need of unfolding preprocessing for time series clustering. In *Proceedings of Workshop on Self-Organizing Maps - WSOM 2005*, pages 251–258, Paris, France, September 2005. 79, 80, 87, 90, 91

[147] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort, and Michel Verleysen. Double quantization of the regressor space for long-term time series prediction : Method and proof of stability. *Neural Networks*, 17(8-9) :1169–1181, 2004. 31, 35

[148] Geoffroy Simon, Amaury Lendasse, Marie Cottrell, Jean-Claude Fort, and Michel Verleysen. Time series forecasting : Obtaining long term trends with self-organizing maps. *Pattern Recognition Letters*, 26(12) :1795–1808, 2005. 31, 35

[149] Geoffroy Simon, Amaury Lendasse, and Michel Verleysen. Bootstrap for model selection : Linear approximation of the optimism. In J. Mira and J.R. Alvarez, editors, *Computational Methods in Neural Modeling, Proceedings of International Work-Conference on Artificial and Natural Neural Networks*, volume 2686 of *Lecture Notes in Computer Science*, pages I182–I189. Springer-Verlag, Mao, Menorca (Spain), June 2003. 23

[150] Geoffroy Simon, Amaury Lendasse, Vincent Wertz, and Michel Verleysen. Fast approximation of the bootstrap for model selection. In *Proceedings of ESANN'2003, European Symposium on Artificial Neural Networks*, pages 475–480, Bruges, Belgium, April 2003. 23, 26, 57

[151] Geoffroy Simon and Michel Verleysen. Lag selection for regression models using high-dimensional mutual information. In *Proceedings of European Symposium on Artificial Neural Networks - ESANN2006*, pages 395–400, Bruges, Belgium, April 2006. 111

[152] A. Sorjamaa, J. Hao, and A. Lendasse. Mutual information and k-nearest neighbors approximator for time series predictions. In *Artificial Neural Networks : Formal Models and Their Applications, Proceedings of International Conference on Artificial Neural Networks - ICANN'05*, volume 3697 of *Lecture Notes in Computer Science*, pages 553–558. Springer, 2005. 111

[153] A. Sorjamaa, N. Reyhani, and A. Lendasse. Input and structure selection for k-nn approximator. In *Computational Intelligence and Bioinspired Systems, Proceedings of International Workshop on Artificial Neural Networks - IWANN 2005*, volume 3512 of *Lecture Notes in Computer Science*, pages 985–991. Springer, 2005. 111

[154] Antti Sorjamaa and Amaury Lendasse. Time series prediction using dirrec strategy. In *Proceedings of ESANN'2006, European Symposium on Artificial Neural Networks*, pages 143–148, Bruges, Belgium, December 2006. 67

[155] A. Sperduti. Neural networks for adaptive processing of structured data. In Springer, editor, *Proceedings of ICANN 2001*, pages 5–12, 2001. 34

[156] Marc Strickert. *Self-Organizing Neural Networks for Sequence Processing*. PhD thesis, University of Osnabrück, Institute of Computer Science, Osnabrück, Germany, 2004. 34, 35

[157] Marc Strickert, Thorsten Bojer, and Barbara Hammer. Generalized relevance lvq for time series. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of ICANN 2001, International Conference on Artificial Neural Networks*, pages 677–683. Springer, 2001. 35, 98

[158] Marc Strickert and Barbara Hammer. Neural gas for sequences. In *Proceedings of Workshop on Self-Organizing Maps WSOM'03*, Kitakyushu, Japan, Septembre 2003. 35

[159] Marc Strickert and Barbara Hammer. Merge som for temporal data. *Neurocomputing*, 64 :39–72, 2005. 34

[160] Marc Strickert, Barbara Hammer, and Sebastian Blohm. Unsupervised recursive sequence processing. *Neurocomputing*, 63 :69–98, 2005. 34

[161] Z. Struzik. Time series rule discovery : Tough, not meaningless. In *Proceedings of International Symposium on Methodologies for Intelligent Systems (ISMIS)*, volume 2871 of *Lecture Notes in Artificial Intelligence*, pages 32–39, Maebashi City, Japan, October 2003. Springer-Verlag. 90

[162] Johan A. K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002. 27

[163] Johan A. K. Suykens, Lukas Lukas, and Joos Vandewalle. Sparse approximation using least squares support vector machines. In *Proceedings of the IEEE International Symposium on Circuits and Systems - ISCAS*, pages II757–II760, 2000. 27

[164] Johan A. K. Suykens, Lukas Lukas, and Joos Vandewalle. Sparse least squares support vector machine classifiers. In *Proceedings of ESANN'2000, European Symposium on Artificial Neural Networks*, pages 37–42, Bruges, Belgium, April 2000. 27

[165] F. Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381. Springer, 1981. 93, 94, 95

[166] F. Takens. On the numerical determination of the dimension of an attractor. In *Dynamical Systems and Bifurcations*, volume 1125 of *Lecture Notes in Mathematics*, pages 99–106. Springer, 1985. 96

[167] Howell Tong. Some comments on nonlinear time series analysis. In Colleen D. Cuttler and Daniel Kaplan, editors, *Non linear dynamics and time series : building a bridge between the natural and statistical sciences*. 67

[168] Peter D. Turney. A theory of cross-validation error. *Journal of Experimental and Theoretical Artificial Intelligence*, 6 :361–391, 1994. 19

[169] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998. 8

[170] Markus Varsta, Jukka Heikkonen, Jouko Lampinen, and José del R. Millán. Analytical comparison of the temporal kohonen map and the recurrent self organizing map. In *Proceedings of ESANN'2000, European Symposium on Artificial Neural Networks*, pages 273–280, Bruges, Belgium, April 2000. 33

[171] Markus Varsta, Jukka Heikkonen, Jouko Lampinen, and José del R. Millán. Temporal kohonen map and the recurrent self-organizing map : Analytical and experimental comparison. *Neural Processing Letters*, 13 :237–251, 2001. 33

[172] J. Vesanto. Using the som and local models in time-series prediction. In *Proceedings of Workshop on Self-Organizing Maps WSOM'97*, pages 209–214, Espoo, Finland, 1997. 34

[173] Thomas Voegtlin. Context quantization and contextual self-organizing maps. In *Proceedings of the IJCNN'2000*, volume 6, pages 20–25, 2000. 34

[174] Thomas Voegtlin and Peter F. Dominey. Recursive self-organizing maps. *Neural Networks*, 15(8-9) :979–991, 2002. 33

[175] J. Walter, H. Ritter, and K. Schulten. Non-linear prediction with self-organising maps. In *Proceedings of International Joint Conference on Neural Networks - IJCNN*, pages 589–594, San Diego, CA, July 1990. 34

[176] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4 :66–82, 1960. 110

[177] Andreas S. Weigend and Neil A. Gershenfeld. *Time Series Prediction : Forecasting the Future and Understanding the Past*. Addison-Wesley, Reading, M.A., 1994. 25, 56, 131, 132

[178] Hassler Whitney. Differentiable manifolds. *The Annals of Mathematics*, 37(3) :645–680, July 1936. 95

[179] Bernard Widrow and Michael A. Lehr. 30 years of adaptive neural networks : Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, 78(9) :1415–1445, 1990. 8

[180] David Wolpert. On bias plus variance. *Neural Computation*, 9(6) :1211–1243, 1997. 19