Université Paris I - Panthéon-Sorbonne

Thèse de doctorat

présenté par

Joseph Rynkiewicz

en vue de l'obtention du titre de

docteur en sciences

(arrêté du 30 Mars 1992)

Spécialité : **mathématiques appliquées**

Modèles hybrides intégrant des réseaux de neurones artificiels à des modèles de chaînes de Markov cachées : application à la prédiction de séries temporelles.

Sous la direction de

Marie Cottrell

Soutenue le Lundi 18 Décembre 2000 devant le jury composé de :

Hervé BOURLARD	Rapporteur
Marie COTTRELL	
Elisabeth GASSIAT	Présidente
Jean-Claude FORT	
Xavier GUYON	
Michel ROUSSIGNOL	Rapporteur
Jian-Feng YAO	

Je voudrais tout d'abord remercier Marie Cottrell pour m'avoir accueilli au SAMOS et pour avoir encadré mon travail durant ces trois dernières années. Sa grande disponibilité, ses qualités humaines et sa confiance m'ont été très précieuses.

Mes remerciements vont également à Jian Feng Yao et Xavier Guyon qui ont su m'orienter et me conseiller ; leurs recommandations et critiques m'ont permis de finaliser au mieux ce document.

Je voudrais remercier Michel Roussignol et Hervé Bourlard pour avoir accepté la lourde tâche de rapporteur et pour leur participation à ce jury.

Je remercie aussi Jean-Claude Fort et Elisabeth Gassiat de m'avoir fait l'honneur de faire partie de mon jury.

Un grand merci à tous les membres du SAMOS pour leur accueil, leur dévouement et leur bonne humeur.

Un gros merci à Cécile, pour son indispensable soutien et sans qui rien n'aurait été possible.

A Paul et Mathilde Rynkiewicz...

Table des matières

Chapitre 1

Introduction

1.1 Cadre de l'étude

Les séries temporelles à temps discret apparaissent naturellement dans la vie de tous les jours, nous suivons presque en direct les évolutions du nombre de chômeurs dans un pays donné, des cours de la bourse, du niveau de production ou de consommation d'électricité, du taux de pollution par l'ozone etc... Par exemple, les médias nous informent du nombre de chômeurs mois après mois et se réjouissent lorsqu'il baisse. Un bon modèle de la série mensuelle du chômage pourra avoir une valeur explicative et par exemple permettre de savoir si la réduction du temps de travail a une conséquence sur la réduction du taux de chômage. De même, de plus en plus de personnes regardent les cours de la bourse jour après jour pour connaître la valeur de leur portefeuille boursier et décider d'acheter ou de vendre des actions. La modélisation des cours de la bourse a aussi permis aux banques de construire une large gamme de nouveaux produits financiers (les options). D'autres exemples sont d'un grande importance économique et sociale, ainsi EDF doit pouvoir prévoir au mieux la consommation électrique heure après heure dans chaque région afin d'ajuster au mieux sa production à la demande. Enfin, les pouvoirs publics veulent prévoir le taux de pollution d'ozone du lendemain afin de prendre les mesures nécessaires si jamais celui-ci risque de se rapprocher du niveau dangereux pour la santé. Cette liste d'exemples est très loin d'être exhaustive, mais on voit déjà qu'il est important de modéliser au mieux ces phénomènes temporels.

Pour la plupart des phénomènes chronologiques, les observations futures dépendent des observations passées. Ainsi une méthode de modélisation courante consiste à tâcher d'exprimer une valeur future en fonction de valeurs passées et à déterminer ses caractéristiques statistiques. Pour des suites de données assez régulières, les propriétés asymptotiques du modèle donnent un bon aperçu de son comportement futur lorsque le nombre d'observations est assez grand. On utilise donc le passé de la série pour prévoir son avenir. Cette méthode part du principe que le comportement de la série ne varie pas beaucoup au cours du temps, ce qui se traduit en langage probabiliste par

la "stationnarité" du phénomène étudié. Hélas, cette hypothèse, bien que très souvent employée, peut se révéler totalement inadaptée au problème. L'objet de cette thèse est donc d'étudier des phénomènes qui changent de comportement au cours du temps. Cependant ces changements de comportement introduisent une difficulté supplémentaire puisque il est nécessaire de changer de modèle. On se limitera donc à un cas simple : les séries stationnaires par morceaux. Une série stationnaire par morceaux est essentiellement une série qui a un nombre fini de comportements différents. Il s'agit bien sûr d'une schématisation de la réalité, mais comme on le verra dans la suite, elle peut déjà apporter de bonnes améliorations de l'adéquation entre le modèle et les observations.

Dans ce document, on s'intéresse principalement à la modélisation paramétrique des séries temporelles, et à une méthode de modélisation des changements de comportement (régime) grâce à une chaîne de Markov dans un espace d'état fini. Ce modèle a l'avantage d'être suffisamment simple pour que les calculs sous-jacents soient faisables, bien qu'il limite le champ d'investigation à un nombre fini de régimes possibles. Lorsque le régime est fixé, le comportement de la série est modélisé par une fonction, dite d'autorégression, et on fera correspondre à chaque régime une de ces fonctions.

Dans les années 70 les modèles autorégressifs linéaires (AR) ont été largement étudiés et expérimentés. Néanmoins l'hypothèse de linéarité se révèle souvent beaucoup trop restrictive. Depuis, de nombreuses tentatives pour introduire des modèles autorégressifs non-linéaires ont abouti dans certains cas à de meilleurs modèles. Parmi ceux-ci les réseaux de neurones et plus particulièrement les perceptrons multicouches (MLP) ont connu un succès important. Cela pour des qualités théoriques comme la propriété d'approximation universelle, mais aussi pour des raisons pratiques comme la facilité du calcul de la dérivée de la fonction représentée par le MLP par rapport à ses paramètres. De même, les MLP sont séduisants pour des raisons philosophiques, essentiellement à cause de l'analogie avec la nature, car les perceptrons multicouches "apprennent" d'une manière qui imite de façon très schématique les neurones de notre cerveau. Les MLP ont cependant de sérieux inconvénients, soit théoriques, car on est confronté à des problèmes de "surapprentissage", soit pratiques comme l'existence de minima locaux.

Néanmoins, s'ils sont utilisés correctement, ils donnent des résultats satisfaisants dans la pratique. De plus, comme ils généralisent les modèles autorégressifs linéaires simples, ils seront un des outils principaux de cette thèse.

1.2 Organisation de la thèse

L'idée directrice de la thèse est la modélisation des séries non-stationnaires par morceaux. Une idée naturelle est alors, d'utiliser un mélange de N experts, chaque expert étant une fonction paramétrique de régression (par exemple un MLP, un AR,...), et de trouver un moyen de dire à un instant donné lequel de ces experts fait la prévision la plus pertinente. Le rôle de la chaîne de Markov cachée est justement de pouvoir

modéliser la probabilité qu'à l'instant t, ce soit l'expert $i \in \{1, \cdots, N\}$ qui fasse la meilleure prévision.

Nous allons donc dans un premier temps approfondir l'étude des modèles autorégressifs fonctionnels de type MLP, puis nous introduirons les modèles de chaînes de Markov cachées, pour finalement aboutir aux modèles intégrants des chaînes de Markov cachées et des MLP.

Nous abordons les sujets suivants :

- Chapitre 2
  Présentation du perceptron multicouches et des problématiques associées.
- Chapitre 3
  Etude empirique de l'estimation et de l'initialisation des paramètres d'un MLP par recuit simulé.
- Chapitre 4
  Estimation et identification des modèles autorégressifs non-linéaires multidimensionnels : Application au perceptron multicouches.
- Chapitre 5
  Introduction aux chaînes de Markov cachées (HMM) : Estimation, algorithme E.M., estimation d'un modèle hybride MLP/HMM sur une série de laboratoire.
- Chapitre 6
  Modèles autorégressifs linéaires à changements de régime markoviens : Un nouveau calcul de l'algorithme E.M.
- Chapitre 7
  Estimation directe des modèles autorégressifs à changements de régime markoviens.
- Chapitre 8
  Etude statistique de l'estimateur du maximum de vraisemblance pour des modèles à changements de régime markoviens
- Chapitre 9
  Etude des pics de pollution en niveau d'ozone.

Voilà un résumé des différents chapitres :

Présentation du perceptron multicouches et des problématiques associées (Chapitres 2 et 3).

Aspects théoriques Après un rappel de quelques résultats fondamentaux, tels que la propriété d'approximation universelle, l'identifiabilité, nous discuterons du problème du "surapprentissage" du MLP, en présentant sa signification théorique et les différentes méthodes utilisées pour contourner ce problème. On discute notamment du principe de la minimisation du risque structurel et nous faisons un parallèle entre ce principe et la méthode d'identification utilisée dans ce mémoire.

Aspects numériques La façon la plus courante de déterminer les paramètres (poids) optimaux d'un MLP sur un problème donné est de minimiser une fonction de coût, par exemple la moyenne des carrés des erreurs de prévision. C'est une tâche relativement aisée avec un perceptron multicouches, puisque le gradient de cette fonction se calcule par l'algorithme de rétro-propagation. On peut ensuite utiliser l'une des nombreuses méthodes d'optimisation différentielle pour approcher le minimum de cette fonction de coût. Nous décrivons brièvement les méthodes déterministes du premier ordre et d'ordres supérieurs, ainsi que les méthodes stochastiques.

Estimation et initialisation des paramètres d'un MLP par recuit simulé. L'algorithme de rétro-propagation permet de calculer la dérivée par rapport aux paramètres de la fonction de coût associée à un MLP. On peut ensuite utiliser une méthode différentielle pour localiser le minimum de cette fonction. Cependant cette méthode ne garantit que la convergence vers un minimum local de la fonction à minimiser, c'est pourquoi on estime le modèle en partant de nombreuses initialisations différentes. Afin d'améliorer cette méthode, nous étudions empiriquement une méthode d'estimation et d'initialisation par recuit simulé. L'avantage de cette méthode stochastique est qu'elle permet, en théorie, de converger vers l'ensemble des minima globaux d'une fonction.

Estimation et identification des modèles autorégressifs non-linéaires multidimensionnels : Application au perceptron multicouches (Chapitre 4). Dans le cas où la série à modéliser est multidimensionnelle, on utilise souvent les moindres carrés comme fonction de coût. Cela revient à minimiser la trace de la matrice de covariance empirique. Néanmoins cette méthode suppose implicitement que la matrice de covariance du bruit est l'identité, ce qui est généralement faux. L'étude du maximum de vraisemblance pour un bruit gaussien montre que le contraste à minimiser dans ce cas est le logarithme du déterminant de la matrice de covariance empirique. On montre que sous de bonnes conditions de régularité du modèle, ce contraste a de bonnes propriétés statistiques, et nous en déduisons un contraste pénalisé consistant. Dans le cas des MLP, on montre alors que les paramètres qui minimisent un bon critère d'information convergent bien vers le vrai modèle en supposant uniquement que le bruit a un moment d'ordre strictement supérieur à 12. Nous montrons de plus que ce critère d'information appartient à une famille qui contient le BIC d'Akaike.

Introduction au chaînes de Markov cachées (HMM) : Estimation, algorithme E.M., estimation d'un modèle hybride MLP/HMM sur une série de laboratoire (Chapitre 5). Les chaînes de Markov cachées ont été introduites à la fin des années 60, elles sont le modèle de prédilection pour la reconnaissance de la parole. Utilisées initialement dans des espaces d'observation finis, elles ont rapidement été généralisées au cas où l'espace des observations est continu.

En supposant que chaque régime correspond à un modèle autorégressif linéaire, on obtient une généralisation connue sous le terme : modèle autorégressif linéaire à

changements de régime markoviens. Enfin, en remplaçant les fonctions de régression linéaires par des fonctions non linéaires (par exemple des MLP), on généralise ce modèle aux modèles autorégressifs non-linéaires à changements de régime markoviens. Lorsque les fonctions de régression sont des MLP, on appelle aussi ce modèle : Modèle hybride HMM/MLP. Nous introduisons la méthode la plus populaire pour estimer ce genre de modèle, c'est-à-dire l'algorithme E.M. Ensuite nous montrons, par une étude sur la série laser, le bon comportement de ce modèle qui permet notamment de mettre à jour les régimes de la série. Ces modèles ont donc à la fois un pouvoir explicatif et prédictif.

Modèles autorégressifs linéaires à changements de régime markoviens : Un nouveau calcul de l'algorithme E.M. (Chapitre 6). On généralise ici la méthode de R. Elliot au cas des modèles autorégressifs à changement de régimes markoviens. Cela permet d'obtenir des estimateurs "récursifs" de quantités telles que les probabilités conditionnelles des états de la chaîne de Markov cachée mais aussi des statistiques exhaustives qui apparaissent lors de l'algorithme E.M. On en déduit aussi un algorithme E.M. "en ligne" qui s'avère être beaucoup plus rapide que l'algorithme E.M. classique.

Estimation directe des modèles autorégressifs à changements de régime markoviens (Chapitre 7). Dans le cas où les fonctions sont non-linéaires, l'algorithme E.M. est très lent. On montre qu'il est possible d'estimer les paramètres par un calcul direct de la log-vraisemblance et de la dérivée de la log-vraisemblance. On obtient ainsi une méthode d'estimation sans algorithme E.M., ainsi qu'une méthode d'estimation récursive. Nous montrons sur des simulations les très bonnes performances de ces algorithmes.

Etude statistique de l'estimateur du maximum de vraisemblance pour des modèles à changements de régime markoviens (Chapitre 8). Après avoir étudié les conditions pour que le modèle soit ergodique, nous montrons par une méthode originale la consistance de l'estimateur. Nous donnons notamment des conditions simples pour la consistance de l'estimateur dans le cas où le bruit est gaussien. Nous établissons aussi la normalité asymptotique de cet estimateur. Enfin, nous évoquons le problème d'identification du modèle dans le cas où le nombre d'états de la chaîne de Markov cachée est inconnu.

Etude des pics de pollution en niveau d'ozone (Chapitre 9). Ce mémoire se conclut par l'application du modèle et des méthodes étudiés à la prévision des pics de pollution en ozone sur Paris. On dispose pour cela d'une longue série chronologique d'observations des taux d'ozone. Les changements visibles de comportement de cette série mettent en lumière les avantages de la modélisation tenant compte des changements de régime.

Chapitre 2

Le perceptron multicouches

Un perceptron multicouches est, pour le statisticien, une fonction paramétrique non-linéaire. Il admet une notation qui s'inspire des neurones biologiques.

Nous ne présentons ici que le perceptron multicouches passe-avant, néanmoins il existe de nombreuses variantes.

2.1 Introduction aux modèles connexionnistes

2.1.1 Le neurone formel

Le neurone formel a été introduit par McCullogh et Pitts en 1943 [17]. Ils ont basé leur modèle sur les observations neurophysiologiques des neurones du système nerveux. Cependant il ne s'agit bien sûr que d'une approximation très schématique du neurone biologie. Il est défini de la manière suivante :

- Des entrées réelles $x_i, i \in \{1, \cdots, m\}$
- Des poids $W_i, i \in \{0, \cdots, m\}$

Le poids W_0 est relié à une entrée constante, l'opposé de W_0 peut être vu comme une valeur seuil, au-delà de laquelle le neurone est activé.

Le neurone effectue les deux opérations suivantes en calculant :

1. Son potentiel : $W_0 + \sum_{i=1}^{m} W_i x_i$
2. Son activation, grâce à une fonction d'activation ϕ : $\phi\left(W_0 + \sum_{i=1}^{m} W_i x_i\right)$

Les fonctions d'activations $\phi : \mathbb{R} \longrightarrow \mathbb{R}$ peuvent prendre des formes multiples, généralement non-linéaires, comme la fonction signe :

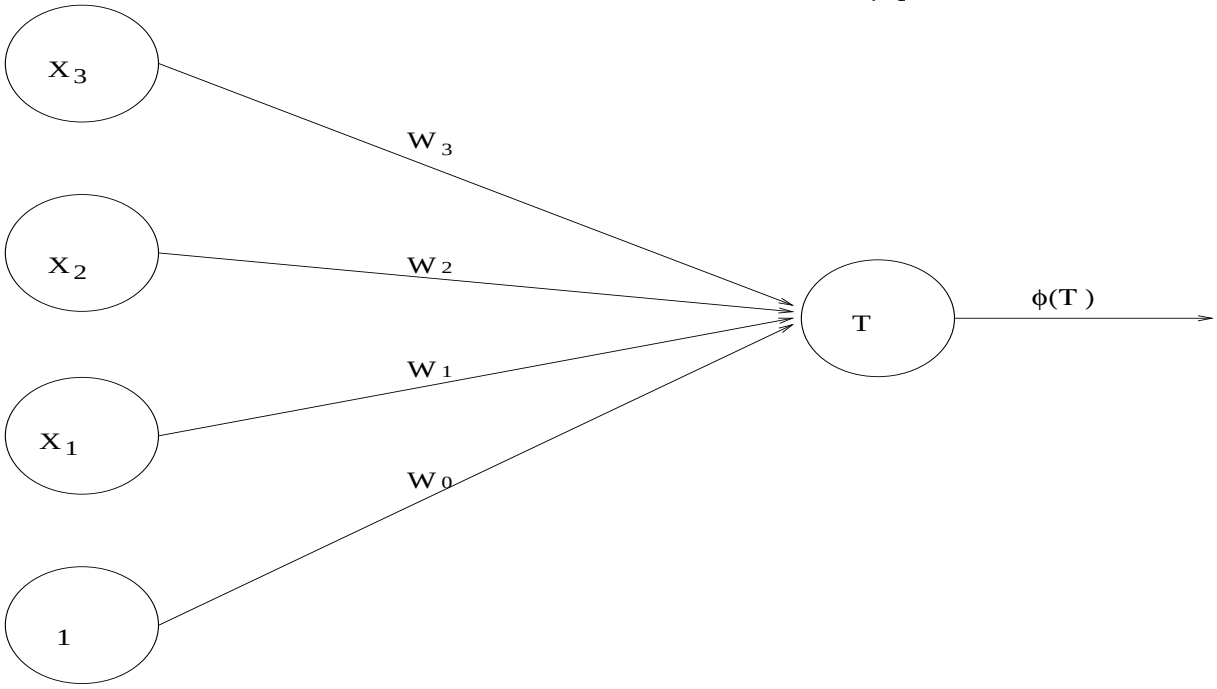
$$\begin{cases} \phi(x) = 1 \text{ si } x \geq 0 \\ \phi(x) = -1 \text{ si } x < 0 \end{cases}$$

ou bien, appartenir à la famille des fonctions sigmoïdes :

$$\phi(x) = c\frac{e^{kx}-1}{e^{kx}+1} + r \; ; \; c, k, r \in \mathbb{R} \; ; \; c, k > 0 \tag{2.1}$$

Si on pousse l'analogie avec le modèle biologique du neurone, les poids W_i représentent les poids synaptiques, $W_0 + \sum_{i=1}^{m} W_i x_i$ le potentiel et $\phi(W_0 + \sum_{i=1}^{m} W_i x_i)$ la sortie de l'axone.

FIG. 2.1 – Schéma du neurone formel (en notant $T = W_0 + \sum_{i=1}^{m} W_i x_i$, avec $m = 3$)



2.1.2 Le perceptron multicouches (MLP)

Un perceptron multicouches est un réseau de neurones formels, où l'information circule de couches en couches. Si il n'y a pas de couche cachée, et qu'il s'agit donc uniquement d'un neurone formel, on l'appelle perceptron simple. Historiquement, c'est le perceptron simple qui a été introduit en premier par Rosenblatt [55]. Il permet de séparer facilement deux ensembles linéairement séparables, à l'aide d'un algorithme d'apprentissage simple. Cependant, il échoue par exemple à modéliser un fonction non-linéaire aussi simple que le "ou exclusif" (XOR). La solution est apparue rapidement grâce à l'introduction de nouvelles couches de perceptron simples. Cependant l'algorithme d'apprentissage de Rosenblatt ne fonctionnait plus et il s'en suivit une désintérêt pour ce modèle.

Le perceptron renaît lorsque deux équipes (LeCun [43], Rumelhart et al. [56]) ont mis au point séparément l'algorithme de rétro-propagation du gradient, permettant un

apprentissage du perceptron multicouches par minimisation d'une fonction de coût (ou de risque).

2.1.2.1 Notation formelle et représentation graphique du MLP

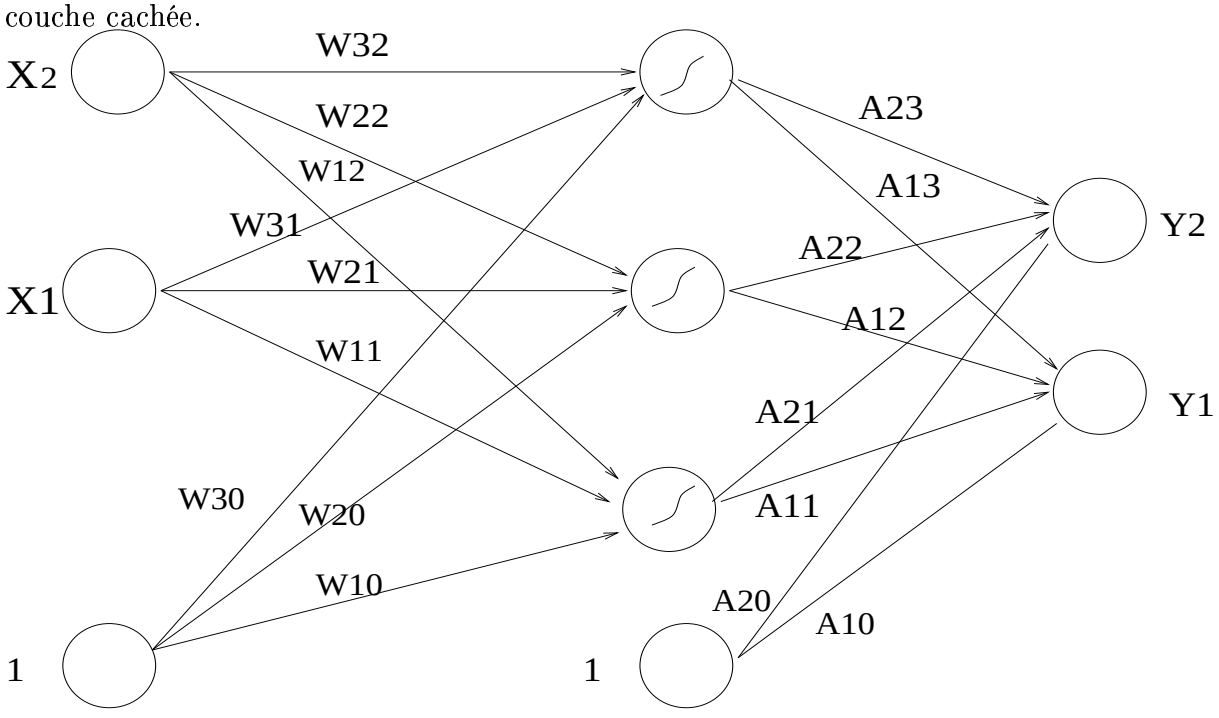
Si on note $(W_{ij})_{1\leq i\leq C, 1\leq j\leq m+1}$ les poids reliant les entrées $(X_j)_{1\leq j\leq m}$ et les unités cachées $(u_i)_{i\leq i\leq C}$ et $(A_{ij})_{1\leq i\leq s, 1\leq j\leq C+1}$ les poids reliant les unités cachées et les sorties $(Y_i)_{1\leq i\leq s}$ et si les unités cachées ont une fonction d'activation ϕ, le MLP représentera la fonction :

$$(Y_i)_{1\leq i\leq s} = \left(\sum_{j=1}^{C} A_{ij}\left(\phi\left(\sum_{k=1}^{m} W_{jk}X_k + W_{j0}\right)\right) + A_{i0}\right)_{1\leq i\leq s} \tag{2.2}$$

La figure 2.2 donne un exemple de perceptron à une couche cachée avec 2 entrées, 3 unités cachées et 2 sorties. qui représente la fonction :

$$\begin{cases} Y_1 = A_{13}\phi\left(W_{32}X_2 + W_{31}X_1 + W_{30}\right) + A_{12}\phi(W_{22}X_2 + W_{21}X_1 + W_{20}) \\ +A_{11}\phi(W_{12}X_2 + W_{11}X_1 + W_{10}) + A_{10} \\ Y_2 = A_{23}\phi\left(W_{32}X_2 + W_{31}X_1 + W_{30}\right) + A_{22}\phi(W_{22}X_2 + W_{21}X_1 + W_{20}) \\ +A_{21}\phi(W_{12}X_2 + W_{11}X_1 + W_{10}) + A_{20} \end{cases}$$

FIG. 2.2 – Exemple de Réseaux de neurones du type perceptron multicouches à une couche cachée.



Remarque 1 *Si les fonctions d'activation des unités cachées sont des sigmoïdes de paramètres* $(c_i, k_i, r_i)_{1\leq i\leq C}$ *(cf (2.1)), alors le MLP avec pour nouveaux paramètres :*

$$\begin{cases} \widetilde{W}_{ij} = \frac{k_i W_{ij}}{2},\ 1 \leq i \leq C,\ 0 \leq j \leq m \\ \widetilde{A}_{ij} = c_i A_{ij},\ 1 \leq i \leq s,\ 1 \leq j \leq C \\ \widetilde{A}_{i0} = A_{i0} + \sum_{j=1}^{C} r_j \end{cases}$$

et des tangentes hyperboliques pour fonctions d'activation, représente la même fonction. Ainsi, lorsque les fonctions d'activations appartiennent à la famille des sigmoïdes, on ne considère que des MLP où les fonctions d'activation sont des tangentes hyperboliques.

Remarque 2 *On peut aisément généraliser ces équations dans le cas où le MLP a plusieurs couches cachées (voir, par exemple, Haikin [34])*

2.1.3 Quelques propriétés du perceptron multicouches

Les MLP ont rapidement suscité l'intérêt des mathématiciens qui ont prouvé de nombreuses propriétés intéressantes. Il existe encore des problèmes non résolus liés à l'estimation par MLP, notamment en raison de leur caractère non-linéaire. En fait si les applications opérationnelles sont très développées, on ne contrôle pas encore bien ce modèle. Nous commençons cette section par le rappel de quelques propriétés fondamentales :

- La propriété d'approximateur universel
- L'identifiabilité du modèle
- Les équations de propagation et de rétro-propagation.

2.1.3.1 Un approximateur universel

La possibilité d'approcher des fonctions par des MLP a déjà suscité une abondante littérature. Cybenko [18], Hornik [37], par exemple, ont étudié la propriété d'approximation des fonctions continues à support compact par des MLP à une couche cachée, munis de fonction d'activation sigmoïdes. Ainsi on a le théorème de Hornik et al. [37].

Théorème 1 *Soit un perceptron multicouches défini par l'équation (2.2), où* ϕ *est une fonction strictement croissante et bornée. Soit* K *un compact de* $\mathbb{R}^m$ *. Alors, pour toute fonction* f *continue à support compact (*$f \in C(K)$ *),* $f : \mathbb{R}^m \longrightarrow \mathbb{R}^s$ *et pour* $\epsilon > 0$*, il existe un entier* C *et un vecteur de paramètre* $W = \left((W_{ij})_{1\leq i\leq C, 1\leq j\leq m+1}, (A_{ij})_{1\leq i\leq s, 1\leq j\leq C+1}\right) \in \mathbb{R}^{C\times(m+1)+s\times(C+1)}$ *tels que* $\forall\,(X_1, \cdots, X_m) \in \mathbb{R}^m$

$$\left\| f(X_1, \cdots, X_m) - \left(\sum_{j=1}^{C} A_{i(j+1)} \left(\phi \left(\sum_{k=1}^{m} W_{j(k+1)} X_k + W_{j1} \right) \right) + A_{i1} \right)_{1\leq i\leq s} \right\| < \epsilon$$

où $\|.\|$ *est une norme de* $\mathbb{R}^s$ *.*

Ce théorème ne fournit pas de vitesse d'approximation. C'est pourquoi différents travaux ont suivi pour préciser la vitesse de convergence. Citons par exemple Barron [4].

Théorème 2 *Soit $f : \mathbb{R}^m \longrightarrow \mathbb{R}$, une fonction et $\widetilde{f}$ sa transformé de Fourier. Posons $|\omega|_1 = \sum_{j=1}^m |\omega_j|$ la norme l_1 de ω sur $\mathbb{R}^m$. Soit un perceptron multicouches défini par l'équation (2.2), où ϕ est une fonction sigmoïde, et une seule sortie. Alors, si f vérifie*

$$\int_{\mathbb{R}^m} |\omega|_1 \left|\widetilde{f}(\omega)\right| d\omega < \infty$$

l'erreur d'approximation de la fonction par un perceptron multicouches est bornée par $O\left(\frac{1}{\sqrt{C}}\right)$

2.1.3.2 Les équations de propagation et de rétro-propagation

Une des propriétés qui a fait la popularité des perceptrons multicouches est la facilité avec laquelle s'effectuent les calculs numériques. On va voir qu'en effet les équations de propagation, qui permettent de calculer les valeurs de la fonction associé à un MLP, et les équations de rétro-propagation qui permettent le calcul de la dérivée par rapport aux paramètres de la fonction MLP s'expriment aisément sous forme matricielle. Comme cela complique peu les équations nous traiterons le cas des MLP avec un nombre de couches cachées quelconque.

Notation 1 *En notant $.^T$ l'opérateur de transposition :*

- *Si $X = (X_1, \cdots, X_n)^T \in \mathbb{R}^n$ est un vecteur réel, on désignera par $\widetilde{X}$ le vecteur $\widetilde{X} := (1, X_1, \cdots, X_n)^T \in \mathbb{R}^{n+1}$.*
- *Si $M = (m_{ij})_{1\leq i\leq n, 0\leq j\leq m} \in \mathbb{R}^{n\times(m+1)}$ est une matrice avec n lignes et $m+1$ colonnes, on notera M_* la matrice $(m_{ij})_{1\leq i\leq n, 1\leq j\leq m} \in \mathbb{R}^{n\times m}$ la matrice extraite de M en retirant la première colonne.*
- *Si f est une fonction de $\mathbb{R}$ dans $\mathbb{R}$ et $X = (X_1, \cdots, X_n)^T \in \mathbb{R}^n$ un vecteur réel, on notera $f(X)$ le vecteur $(f(X_1), \cdots, f(X_n))^T$.*
- *Enfin, si X, Y sont deux vecteurs de $\mathbb{R}^n$, on notera $X \otimes Y$ le vecteur $(X_1Y_1, \cdots, X_nY_n)$.*

Supposons qu'un MLP représente la fonction F_W avec N couches cachées, des tangentes hyperboliques pour fonctions d'activation et une sortie linéaire. Notons W^k la matrice de poids entre les couches k et $k+1$, le poids W_{ij}^k reliant l'unité j de la couche k à l'unité i de la couche $k+1$ et C^k le nombre d'unités de la couche k.

Equation de propagation Pour calculer la fonction $X \longmapsto F_W(X)$, il suffit de propager l'entrée X par l'algorithme

$$\begin{cases} X(1) = \widetilde{X} \\ X(k+1) = \tanh\left(W^k \widetilde{X}(k)\right) \end{cases}$$

Cela pour $k = 1$ jusqu'à $k = N$, $X(N+2) = W^{N+1}\widetilde{X}(N+1)$ vaudra alors $F_W(X)$. Il s'agit de la généralisation à plusieurs couches cachées de l'équation (2.2).

Equation de rétro-propagation Lorsque que l'on utilise un MLP, on cherche la plupart du temps à minimiser une fonction de risque $R(W) := W \longmapsto Q(y - F_W(x))$, où $Q(z)$ est une fonction de $\mathbb{R}^s \longrightarrow \mathbb{R}$. Notons $Q'(z)$ la fonction $\left(\frac{\partial Q(z)}{\partial z_i}\right)_{1\leq i\leq s}$. Pour minimiser $R(W)$, on peut utiliser une méthode différentielle en calculant la dérivée de $W \longmapsto Q(y - F_W(x))$.

Il faut remarquer que lorsqu'on utilise la fonction tangente hyperbolique pour fonction d'activation, la dérivée de cette fonction vérifie

$$\tanh'(x) = (1 - \tanh(x)) \times (1 + \tanh(x))$$

Finalement, notons X' le vecteur $((1 - X_i) \times (1 + X_i))_{1\leq i\leq n}$, DW^k la matrice

$$DW^k := \left(\frac{\partial R(W)}{\partial W_{ij}^k}\right)_{1\leq i\leq C^k, 1\leq j\leq C^{k-1}}$$

Soit $\psi(k), 1 \leq k \leq N+1$ des vecteurs réels de dimension $C^{k+1}, 1 \leq k \leq N+1$, on calcule la dérivée de $R(W)$ par l'algorithme :

$$\begin{cases} \psi(N+1) = Q'(y - F_W(x)) \\ \psi(k) = X'(k+1) \otimes \left(\left(W_*^{k+1}\right)^T \psi(k+1)\right) \\ DW^{k+1} = \widetilde{X}(k+1) \times \psi(k+1)^T \\ DW^1 = \widetilde{X}(1) \times \psi(1)^T \end{cases}$$

pour $k = N$ jusqu'à $k = 1$, les vecteurs $\widetilde{X}(k), 1 \leq k \leq N+1$ ayant été calculés par l'algorithme de propagation précédent. Connaissant $DW^k, 1 \leq k \leq N+1$, on connaît la dérivée de $R(W)$.

2.1.3.3 L'identifiabilité du modèle

Lorsqu'on estime un vecteur paramètre, une propriété fondamentale pour obtenir des résultats de consistance, c'est-à-dire le fait que l'estimateur converge (au moins en probabilité) vers le vrai paramètre, est l'identifiabilité du modèle. Cela signifie, que pour une fonction représentable par un MLP donné, il n'y a qu'un seul vecteur paramètre qui représente cette fonction.

Si on ne considère qu'un perceptron multicouches de dimension (m, C, s), c'est-à-dire avec m entrées, C unités cachées et s sorties, comme une fonction paramétrique sur $\mathbb{R}^D$, avec $D = (m+1) \times C + (C+1) \times s$, le modèle n'est pas identifiable. On peut

en effet trouver deux systèmes de paramètres différents qui génèrent les mêmes sorties. Ceci peut être obtenu, par exemple, en permutant l'ordre des neurones de la couche cachée.

Heureusement, pour établir les résultats de consistance, il suffit de se placer sur un ensemble polonais (métrique, complet, séparable) et en prenant un espace quotient de $\mathbb{R}^D$ convenable, les MLP deviennent identifiables.

Nous donnons dans la suite des conditions nécessaires et suffisantes pour que le modèle soit identifiable dans le cas d'un MLP à une couche cachée, avec des tangentes hyperboliques pour fonctions d'activation, car par la remarque 1, cela inclut le cas de toute les fonctions sigmoïdes.

La notation des poids adoptée ici est similaire à celle de la figure 2.2.

Notation 2 *Si $X = (X_1, \cdots, X_m)^T \in \mathbb{R}^m$ est un vecteur d'entrée, on note $v_i(X)$ l'impulsion de la i-ème unité cachée :*

$$\nu_i(X) = W_{i0} + \sum_{j=1}^{m} W_{ij} X_j$$

Fixons m. Le MLP avec C unités cachées est associé à C applications affines : $\begin{array}{c} \mathbb{R}^m \to \mathbb{R} \\ X \longmapsto \nu_j(X) \end{array}$.
On dira que deux fonctions affines ν_1, ν_2 sont "signe-équivalentes" si $\nu_1 = \nu_2$ ou $\nu_1 = -\nu_2$.

On dira qu'un MLP est réductible si et seulement si il vérifie au moins une des conditions (**R**) suivantes

1. Il existe un indice $j \in \{1, \cdots, C\}$ tel que tous les poids A_{ij} , $1 \leq i \leq s$ sont nuls.
2. Il existe au moins deux indices différents $j_1, j_2 \in \{1, \cdots, C\}$ tels que les fonctions ν_{j_1}, ν_{j_2} sont signe-équivalentes
3. Il existe au moins un indice $j \in \{1, \cdots, C\}$ tel que la fonction ν_j est constante

On dira qu'un MLP est irréductible si ce n'est pas un MLP réductible.

Notation 3 *On note $\mathcal{N}_{m,C,s}$ l'ensemble des MLP avec m entrées, C unités cachées et s sorties qui sont irréductibles. $\mathcal{N}_{m,C,s}$ est isomorphe à $\mathbb{R}^D$, avec $D = (m+1) \times C + (C+1) \times s$.*

Remarque 3 *Si $C = 0$, $\mathcal{N}_{m,0,s}$ sont les fonctions linéaires de $\mathbb{R}^m \longrightarrow \mathbb{R}^s$*

Il y a des transformations triviales qui ne changent pas la fonction MLP. Par exemple, si on choisit l'unité cachée i, que l'on change le signe de tous les poids W_{ij} pour $1 \leq j \leq C$ et que l'on change aussi le signe des A_{ij}, $1 \leq i \leq s$, comme la fonction tangente

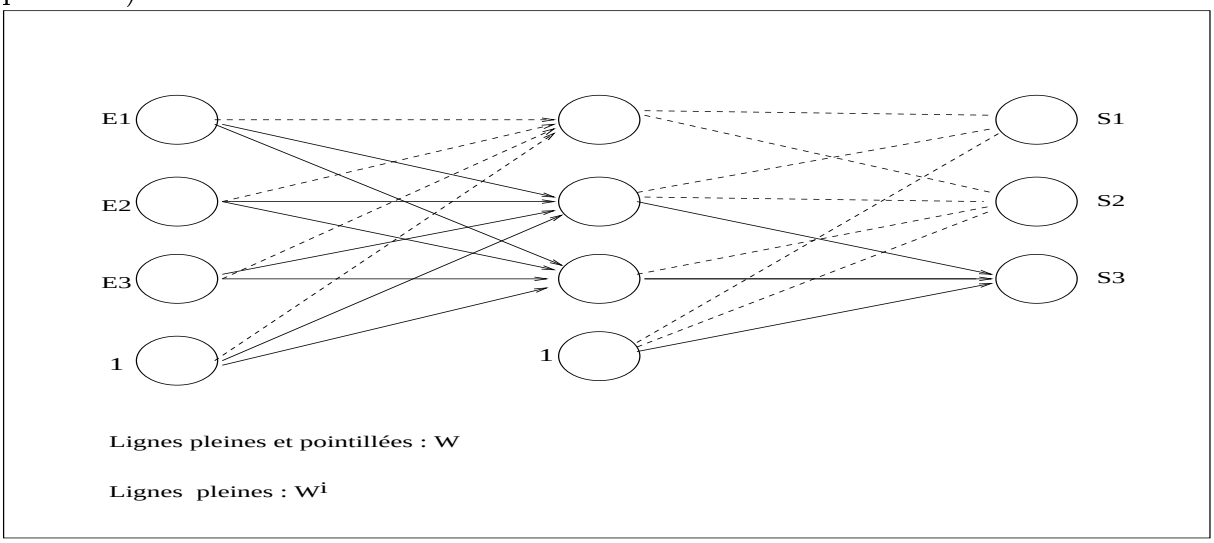
hyperbolique est impaire cela ne changera pas la fonction F_W. Notons $\zeta_j(MLP)$ le MLP résultant de cette transformation.

Une autre possibilité est d'interchanger les deux unités cachées j_1 et j_2, ainsi que les poids correspondants, on note $\eta_{j_1,j_2}(MLP)$ le MLP correspondant. Les applications η_{j_1,j_2}, ζ_j , $j_1, j_2, j \in \{1, \cdots, C\}$ génèrent un groupe fini $\mathcal{G}_{m,C,s}$ de transformations sur l'ensemble $\mathcal{N}_{m,C,s}$ (de cardinal $2^C C!$).

On dira que deux MLP M_1 et M_2 sont équivalents ($M_1 \mathcal{R} M_2$), si et seulement si il existe une transformation $\phi \in \mathcal{G}_{m,C,s}$ telle que $M_1 = \phi(M_2)$. Pour que les fonctions MLP soient identifiables, on considère donc un ensemble de paramètre inclus dans un ensemble quotient $\mathcal{N}_{m,C,s}/\mathcal{R}$, qui est évidement un espace polonais.

Notation 4 *Soit un MLP, de paramètre W de dimension (m, C, s). Appelons F_W la fonction représentée par le MLP et pour $1 \leq i \leq s$, F_{W^i} la fonction représenté par le MLP extrait à partir de la sortie "i", i.e. le MLP à sortie scalaire, qui a les mêmes poids que F_W avant la dernière unité cachée, sauf ceux qui pointent vers une unité cachée non reliée à la sortie "i" et qui ne garde que les poids pointant sur la sortie i de F_W. La figure 2.3 décrit l'exemple d'un MLP avec trois entrées et trois sorties et $i = 3$ (ce qui correspond à la troisième sortie).*

FIG. 2.3 – MLP F_{W^3} (lignes pleines) extrait du MLP F_W (lignes pleines et lignes pointillés)



Identifiabilité des MLP dans $\bigcup_{C=0}^{M} \mathcal{N}_{m,C,s}/\mathcal{R}$, $M \in \mathbb{N}$: Soit $M \in \mathbb{N}$, [58] prouve :

Théorème 3 *Si $s = 1$, les MLP sont identifiables dans $\bigcup_{C=0}^{M} \mathcal{N}_{m,C,1}/\mathcal{R}$, i.e. $F_W = F_{W'} \Leftrightarrow W = W'$.*

On en déduit le théorème :

Théorème 4 *Soit deux MLP, $F_W, F_{W'}$, appartenant à $\bigcup_{C=0}^{M} \mathcal{N}_{m,C,s}/\mathcal{R}$, irréductibles, où $m, s \in (\mathbb{N}^*)^2$, alors $F_W = F_{W'} \Leftrightarrow W = W'$.*

Preuve Supposons que ce ne soit pas le cas, donc qu'il existe $W \neq W'$ tels que $F_W = F_{W'}$. Alors il existe au moins deux MLP extraits irréductibles $F_{W^i}, F_{W'^i}$, tels que $W^i \neq W'^i$ et $F_W i = F_{W'^i}$ sinon au moins un des MLP vérifie (**R**)-(1) et n'est pas irréductible. Mais, comme les MLP extraits ont une seule sortie, cela contredit le théorème 3 ■

Le MLP est donc identifiable dans $\bigcup_{C=0}^{M} \mathcal{N}_{m,C,s}/\mathcal{R}$, pour un M donné. Les démonstrations des propriétés statistiques des estimateurs des paramètres sont souvent faites pour des paramètres réels, cependant leur transposition à $\bigcup_{C=0}^{M} \mathcal{N}_{m,C,s}/\mathcal{R}$ est immédiate.

2.2 Régression non-linéaire et dimensionnement du modèle

Les modèles de régression sont classiques lorsque qu'il faut estimer une fonction avec un bruit additif. Supposons qu'une fonction inconnue ait la forme paramétrique

$$g(x) = F_{W_0}(x)$$

où $W_0 \in \Theta$ est un paramètre inconnu.

Supposons aussi qu'en tout point x_t, on connaisse la valeur de cette fonction avec un bruit additif.

$$y_t = F_{W_0}(x_t) + \varepsilon_t$$

où ε_t est un bruit qui ne dépend pas de x_t.

Le problème est d'estimer la fonction F_{W_0} en utilisant les données obtenues en mesurant la fonction F_{W_0} corrompue par un bruit additif. Ainsi, en utilisant les observations $((x_1, y_1), \cdots, (x_n, y_n))$, on peut estimer le paramètre inconnu en minimisant une fonctionnelle $R_{emp}(W)$, associée à une fonction de coût $Q(x_t, y_t, W)$ avec

$$R_{emp}(W) = \frac{1}{n} \sum_{t=1}^{n} Q(x_t, y_t, W)$$

R_{emp} pourra être, par exemple, la moyenne de l'erreur quadratique

$$R_{emp}(W) = \frac{1}{n} \sum_{t=1}^{n} \|y_t - F_W(x_t)\|^2$$

ou bien l'opposé de la log-vraisemblance (Nous en verrons un exemple chapitre 4).

Il faut maintenant, trouver des conditions pour que si l'on choisit

$$\hat{W}_n = \arg\min R_{emp}(W),\ W \in \Theta$$

le paramètre soit "proche" du vrai paramètre W_0 et notamment que $\hat{W}_n \overset{n\to\infty}{\longrightarrow} W_0$ au moins en probabilité (propriété de consistance faible). Nous allons exposer ici les problèmes liés à la régression non-linéaire dans un cadre du à V. Vapnik [60]. L'idée principale est de contrôler la complexité du modèle pour assurer que l'erreur de celui-ci soit bonne non seulement sur les données que l'on observe, mais aussi sur des données futures, non encore observées, provenant du même phénomène.

2.2.1 Exemples de mauvaise adéquation du modèle aux données

2.2.1.1 Polynômes de degré quelconque

Supposons que $(x_t, y_t) \in \mathbb{R}^2$, et que Θ soit l'ensemble des polynômes de degré quelconque. Alors pour des observations $((x_1, y_1), \cdots, (x_n, y_n))$ telles que

$$\forall i, j \in \{1, \cdots, n\}^2, i \neq j \Rightarrow x_i \neq x_j$$

et $\frac{1}{2} > \epsilon > 0$; il existera toujours un polynôme $P_n \in \Theta$ de coefficients W_n tel que $R_{emp}(W_n) < \epsilon$.

Supposons maintenant que le vrai bruit ait une variance de 1, on aura alors

$$\lim_{n\to\infty} P_n \neq F_{W_0}$$

Il est donc évident que l'on ne peut choisir notre estimateur $\hat{W}_n$ dans un espace trop grand. Pour prendre en compte ce problème, Akaike a énoncé le principe de parcimonie, qui privilégie les modèles avec un petit nombre de paramètres. Hélas, dans un cadre général, cette précaution n'est pas suffisante.

2.2.1.2 Fonction sinus

Plaçons nous dans le même cadre que précédemment, mais supposons de plus que les données y_t sont bornées, c'est-à-dire qu'il existe $A, B \in \mathbb{R}^2$ tels que

$$\forall t \in \mathbb{N}, A < y_t < B$$

et que les écarts entre tous les x_t sont tous différents modulo 2π, par exemple $x_t = 10^{-t}, 1 \leq t \leq n$.

Supposons que les fonctions F_W soient de la forme $F_W(x) = (B - A)\sin(W \times x)$ et $W \in \mathbb{R}$. Alors pour W assez grand et bien choisi (cf Vapnik [60] section 3.6), on aura pour $\frac{1}{2} > \epsilon > 0$, $R_{emp}(W) < \epsilon$ et donc

$$\lim_{n\to\infty} F_{W_n} \neq F_{W_0}$$

Cependant cette fonction n'a qu'un seul paramètre, ainsi le principe de parcimonie est inutile dans ce cas.

Le principal problème dans ces deux exemples est que l'on cherche à estimer le modèle en minimisant un risque empirique

$$R_{emp}(W) = \frac{1}{n}\sum_{t=1}^{n} Q(x_t, y_t, W)$$

en espérant que celui-ci se rapproche du risque théorique

$$R(W) := \int Q(x, y, W)\, dP_x dP_y$$

où

$$\int f(x) dP_x$$

représente l'espérance suivant la loi de la variable aléatoire X de la fonction f .

Or, dans les exemples cités, ce n'est pas le cas. C'est pourquoi Vapnik a étudié la distance du risque empirique avec le risque théorique et en a déduit le principe de minimisation du risque structurel.

2.2.2 Principe de minimisation du risque structurel

Nous allons dans un premier temps donner des définitions essentielles.

2.2.2.1 La dimension de Vapnik-Chervonenkis (VC-dimension)

Notons $z_t = (x_t, y_t) \in \mathbb{R}^d$ et $Q(x_t, y_t, W) := Q(z_t, W)$.

VC-dimension d'un ensemble de fonctions indicatrices Soit un ensemble quelconque de fonctions indicatrices dépendant d'un paramètre W, $Q(z, W)$, $W \in \Theta$, par exemple $\Theta \subset \mathbb{R}^d$ et $Q(z, W) = 1$ si $z \in W$, $Q(z, W) = 0$ si $z \notin W$. La dimension de Vapnik-Chervonenkis d'un tel ensemble est le nombre maximal de vecteurs $z_1, \cdots z_h$ qui peuvent être séparés en deux classes suivant les 2^h façons possibles, en utilisant les fonctions de l'ensemble $Q(z, W)$, $W \in \Theta$. On dira alors que les vecteurs $z_1, \cdots, z_h$ peuvent être *éclatés* par cet ensemble de fonctions.

Si pour tout $n \in \mathbb{N}^*$, il existe un ensemble de n vecteurs qui peuvent être *éclatés* par l'ensemble $Q(z, W)$, $W \in \Theta$, alors la VC-dimension est égale à l'infini.

VC-dimension d'un ensemble de fonction réelles Soit $A \leq Q(z, W) \leq B$, $W \in \Theta$ un ensemble de fonctions réelles paramétrées par W, bornées par des constantes A et B (A et B peuvent être $-\infty$ et $+\infty$). Considérons l'ensemble des fonctions indicatrices

$$I(z, W, \beta) = S(Q(z, W) - \beta), W \in \Theta, \beta \in]A, B[\tag{2.3}$$

où $S(z)$ est le fonction signe :

$$S(z) = \begin{cases} 0 \text{ si } z < 0 \\ 1 \text{ si } z \geq 0 \end{cases}.$$

La VC-dimension de l'ensemble de fonctions réelles $Q(z, W), W \in \Theta$, est alors la VC-dimension de l'ensemble des fonctions indicatrices correspondantes (2.3) avec pour paramètres $W \in \Theta$ et $\beta \in]A, B[$.

Quelques exemples On trouvera dans Vapnik [60], la dimension de quelques ensembles de fonctions.

1. La VC-dimension de l'ensemble des fonctions linéaires :

$$Q(z, W) = \sum_{p=1}^{l} W_p z_p + W_0, \ (W_0, \cdots, W_l) \in \mathbb{R}^{l+1}$$

   est égale à $l + 1$. Notons que pour un ensemble de fonctions réelles quelconques, la dimension de Vapnik-Chervonenkis n'est pas, en général, égale au nombre de paramètres.

2. La VC-dimension de l'ensemble

$$Q(z, W) = \sin(Wz), \ W \in \mathbb{R}$$

   est infini.

2.2.2.2 Borne du risque fonctionnel

On présente ici une borne qui contrôle la capacité de "généralisation" d'un modèle (voir Vapnik [60] chapitre 4), c'est-à-dire l'écart entre le risque empirique d'un modèle et son espérance théorique.

Soit $(z_1, \cdots, z_n)$ des observations i.i.d. Considérons un ensemble de fonctions avec une VC-dimension finie h, et soit $0 \leq Q(z, W) \leq B < \infty$, une fonction positive bornée. Alors, on a avec la probabilité $1 - \eta$

$$R(W) \leq R_{emp}(W) + \frac{B\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4R_{emp}(W)}{B\varepsilon}}\right) \tag{2.4}$$

où

$$R(W) = \int Q(z, W)\, dP_z$$

$$R_{emp}(W) = \frac{1}{n} \sum_{i=1}^{n} Q(z_i, W)$$

$$\varepsilon = 4 \frac{h\left(\ln \frac{2n}{h} + 1\right) - \ln\left(\frac{\eta}{4}\right)}{n}.$$

Si la fonction $Q(z, W)$ est positive non bornée, on a avec la probabilité $1 - \eta$

$$R(W) \leq \frac{R_{emp}(W)}{\left(1 - a(p)\, \tau \sqrt{\varepsilon}\right)_+} \tag{2.5}$$

avec

$$(u)_+ = \max(u, 0)$$

$$a(p) = \left(\frac{1}{2}\left(\frac{p-1}{p-2}\right)^{p-1}\right)^{\frac{1}{p}},\ p \in \mathbb{R},\ p > 2$$

où le couple (p, τ) vérifie

$$\sup_{W \in \Theta} \frac{\left(\int Q^p(z, W)\, dP_z\right)^{\frac{1}{p}}}{\int Q(z, W)\, dP_z} \leq \tau < \infty.$$

2.2.2.3 Principe de minimisation du risque structurel (SRM)

Soit un ensemble E de fonctions $Q(z, W)$, $W \in \Theta$. Supposons qu'il existe des sous-ensembles emboîtés $E_k = \left\{Q\left(z, W^k\right),\ W^k \in \Theta_k\right\}$ tels que

$$E_1 \subset E_2 \subset \cdots \subset E_k \subset \cdots$$

où les ensembles E_k vérifient

1. La VC-dimension h_k de chaque ensemble E_k est fini, et

$$h_1 \leq h_2 \leq \cdots \leq h_k \leq \cdots$$

2. Chaque élément E_k est composé

   (a) ou bien d'un ensemble de fonctions positives bornées

$$0 \leq Q\left(z, W^k\right) \leq B_k < \infty,\ W^k \in \Theta_k$$

(b) ou bien d'un ensemble de fonctions satisfaisant les inégalités

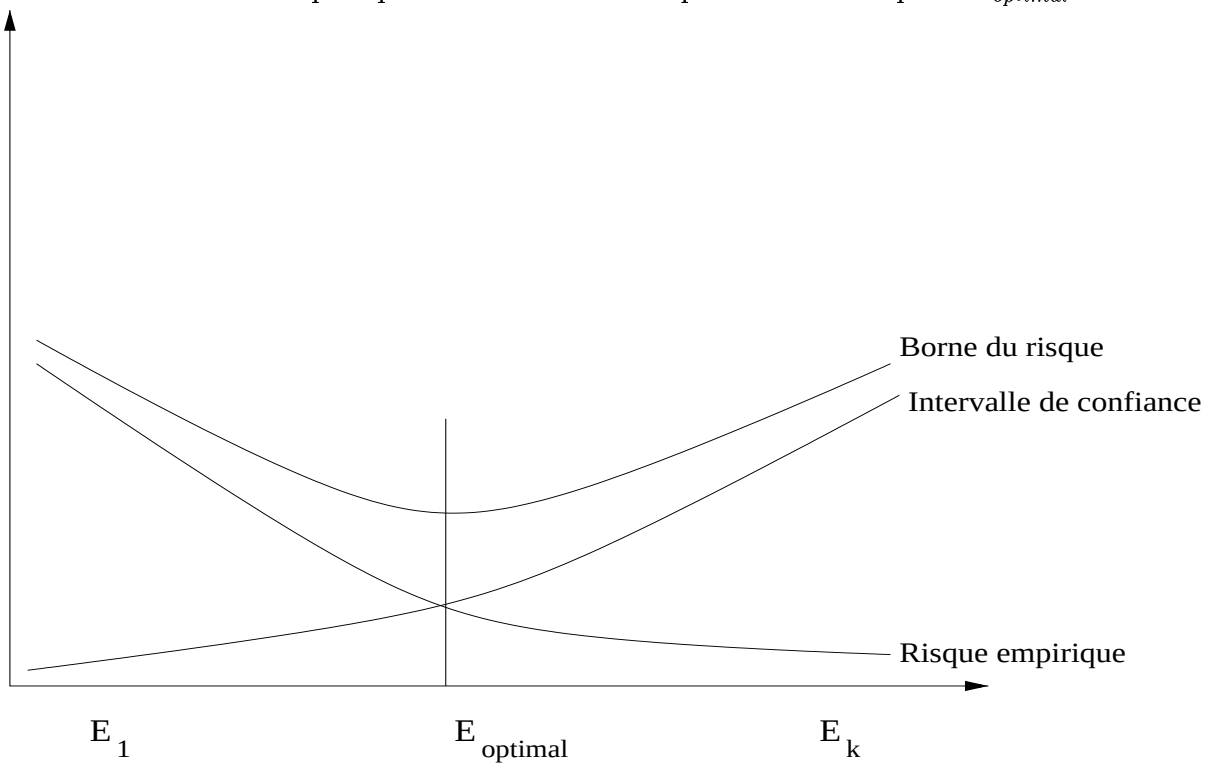
$$\sup_{W^k \in \Theta_k} \frac{\left(\int Q^p\left(z, W^k\right) dP_z\right)^{\frac{1}{p}}}{\int Q(z, W^k) dP_z} \leq \tau_k,\ p > 2 \tag{2.6}$$

pour des paires (p, τ_k).

Le principe de minimisation du risque structurel choisit la fonction $Q\left(z, W^k\right)$ pour laquelle le risque garanti (déterminé par le membre droit des inégalités (2.4) ou (2.5) suivant les circonstances) est minimal.

Ce principe définit donc un compromis entre la qualité d'approximation et la complexité des fonctions d'approximation, puisque lorsque k augmente, le risque empirique $R_{emp}\left(W^k\right)$ décroît, mais le terme responsable de l'intervalle de confiance du risque $R(W)$ (la seconde somme du membre droit de l'inégalité (2.4) ou bien le facteur du membre droit de l'inégalité (2.5)) s'accroît. Le Principe SRM prend en compte ces deux facteurs en choisissant le sous-ensemble E_k pour lequel minimiser le risque empirique fournit la meilleure borne sur le risque théorique (cf figure 2.4).

FIG. 2.4 – La plus petite borne sur le risque est obtenue pour $E_{optimal}$



Analyse asymptotique de la vitesse de convergence Soit $E^* = \bigcup_{k=1}^{\infty} E_k$, supposons que E^* soit dense dans E pour la métrique $\|.\|$:

$$\|Q(z, W_1) - Q(z, W_2)\| := \int |Q(z, W_1) - Q(z, W_2)| \, dP_z$$

Considérons une loi qui à tout $n \in \mathbb{N}^*$, associe l'indice $k = k(n)$ de l'élément E_k pour lequel on minimise le risque empirique. Alors Vapnik [60] établit le théorème suivant

Théorème 5 *Le principe de minimisation du risque structurel fournit des approximations $Q\left(z, W^{k(n)}\right)$, pour lesquelles la suite des risques $R\left(W^{k(n)}\right)$ converge vers le plus petit risque*

$$R(W_0) = \inf_{W \in \Theta} \int Q(z, W) \, dP_z$$

avec le taux asymptotique[1]

$$V(n) = r_{k(n)} + T_{k(n)} \sqrt{\frac{h_{k(n)} \ln n}{n}}$$

si la loi $k = k(n)$ est telle que

$$\lim_{n \to \infty} \frac{T_{k(n)}^2 h_{k(n)} \ln n}{n} = 0$$

où

1. *$T_k = B_k$, si on considère une structure avec des fonctions bornées $0 \leq Q\left(z, W^k\right) \leq B_k$*
2. *$T_k = \tau_k$, si on considère une structure avec des éléments satisfaisant l'inégalité (2.6)*

$r_{k(n)}$ est le taux d'approximation

$$r_k = \inf_{W^k \in \Theta_k} \int Q\left(z, W^k\right) dP_z - \inf_{W \in \Theta} \int Q(z, W) \, dP_z$$

Le SRM permet donc d'ajuster un modèle en le calibrant suivant le nombre de données. On connaît de plus son comportement asymptotique.

[1] On dit que la suite de variables aléatoires ξ_n, $n \in \mathbb{N}^*$ converge vers la valeur ξ_0 avec la vitesse asymptotique $V(n)$, si il existe une constante c telle que

$$(V(n))^{-1} |\xi_n - \xi_0| \xrightarrow{\mathbb{P}} c$$

Les limites du SRM Le principal inconvénient des bornes établies par Vapnik est qu'elles ne sont valables que pour des variables aléatoires indépendantes identiquement distribuées. Or dans ce mémoire nous traitons des séries temporelles et des modèles autorégressifs, c'est-à-dire où les données explicatives seront le passé du processus. De plus, la dimension de Vapnik-Chernovensky n'est connue que pour les MLP ayant des fonctions indicatrices sur la couche cachée. Dans le cas où les fonctions d'activation sont des tangentes hyperboliques, il n'existe que des bornes supérieures de cette dimension. C'est pourquoi, pour éviter la sur-paramétrisation de nos modèles, nous utiliserons un terme de pénalisation qui dépendra du nombre de paramètres et du nombre de données. La philosophie d'une telle approche est assez similaire au principe du SRM, mais le cadre théorique est différent, ainsi que les résultats qui en découlent.

2.2.3 Identification presque sûre des modèles par critère d'information

Nous donnons dans cette section une vue intuitive des méthodes utilisées dans le cadre des perceptrons multicouches. Ces méthodes seront justifiées théoriquement dans le chapitre 4.

2.2.3.1 Les principes de bases

Nous supposerons dorénavant qu'il existe un ensemble dominant E_{dom} telle que le vrai modèle appartienne à $E_v \subset E_{dom}$. Par exemple on pourra supposer que le vrai modèle est un perceptron multicouches avec au plus 10 entrées et 100 unités cachées. Nous devons alors choisir une architecture pour laquelle le risque empirique est petit, mais qui a un nombre raisonnable de paramètres.

Notons δ le nombre de paramètres du MLP F_W, nous choisirons alors le modèle qui minimise :

$$R_{emp}(W) + \delta \frac{c(n)}{n} \tag{2.7}$$

où $c(n)$ est la vitesse de pénalisation. Notons qu'il s'agit là aussi d'un compromis entre la qualité d'approximation ($R_{emp}(W)$) et la complexité du modèle (δ). Cependant, a priori, le parallèle avec le SRM s'arrête ici, puisque deux MLP peuvent avoir un même nombre de paramètres et des VC-dimensions différentes.

Nous montrons aussi au chapitre 4 que, sous de bonnes hypothèses, les modèles qui minimisent le critère (2.7) convergent presque sûrement vers le vrai modèle (pour le SRM, on ne suppose pas que le vrai modèle appartient à l'ensemble des fonctions considérées). Néanmoins dans toutes les applications pratiques de ce mémoire, nous utiliserons la stratégie suivante qui sera alors proche de la philosophie du SRM.

2.2.3.2 La méthode d'identification

Lorsque l'on travaille avec des données réelles, il faut prendre des précautions qui ne sont pas indispensable théoriquement, mais le sont dans la pratique. Ainsi, il faut une méthode qui conduise à un résultat en un temps "raisonnable".

Pour identifier un modèle, nous commençons par essayer de déterminer une architecture qui soit dominante (i.e. qui contienne tous les paramètres du vrai modèle), mais en étant le moins sur-paramétrisée possible. En effet si le modèle dominant est trop grand, les estimations préliminaires seront mauvaises et la méthode adoptée ici donnera de mauvais résultats. Pour déterminer cette architecture, nous procéderons d'une manière identique au SRM, à la différence près que nous n'utilisons pas les bornes du Vapnik (les inégalités (2.4) et (2.5)), mais le critère d'information (2.7).

Fixons un nombre d'entrées suffisamment grand pour être sûr que les données explicatives choisies contiennent les vraies données explicatives, puis estimons le modèle avec des MLP $F_{W_1}, \cdots F_{W_k}, \cdots$, où F_{W_k} représente un MLP, avec une couche cachée et k unités cachées. On remarquera que, si on note h_k la VC-dimension associée au MLP F_{W_k} , on aura bien

$$h_1 \leq \cdots \leq k_k \leq \cdots$$

En notant δ_k le nombre de paramètres de F_{W_k} et $BIC(k) := R_{emp}(W_k) + \delta_k \frac{c(n)}{n}$ la valeur du critère d'information associé à cette architecture on aura pour les premières architectures

$$BIC(1) \geq BIC(2) \geq \cdots \geq BIC(k).$$

On suppose alors que le MLP $F_{W_{k^*}}$ est dominant, si

$$BIC(k^* - 1) < BIC(k^*).$$

Cela revient à ajouter une unité cachée tant que le BIC du modèle décroît et à s'arrêter dès qu'il remonte, on obtient donc la suite

$$BIC(1) \geq \cdots \geq BIC(k^* - 1) < BIC(k^*).$$

Ensuite on enlève, une à une, des connexions (paramètres) du MLP F_{W_k}, suivant la méthode du "stepwise descendant" expliquée à la section suivante, créant une suite de MLP, $\left(F_{W_{k_i}}\right)_{i \in \mathbb{N}}$ avec de moins en moins de paramètres. On remarquera que si on note h_{k_i} la VC-dimension associé au MLP $F_{W_{k_i}}$, on aura :

$$h_{k_1} \geq \cdots \geq h_{k_i} \geq \cdots$$

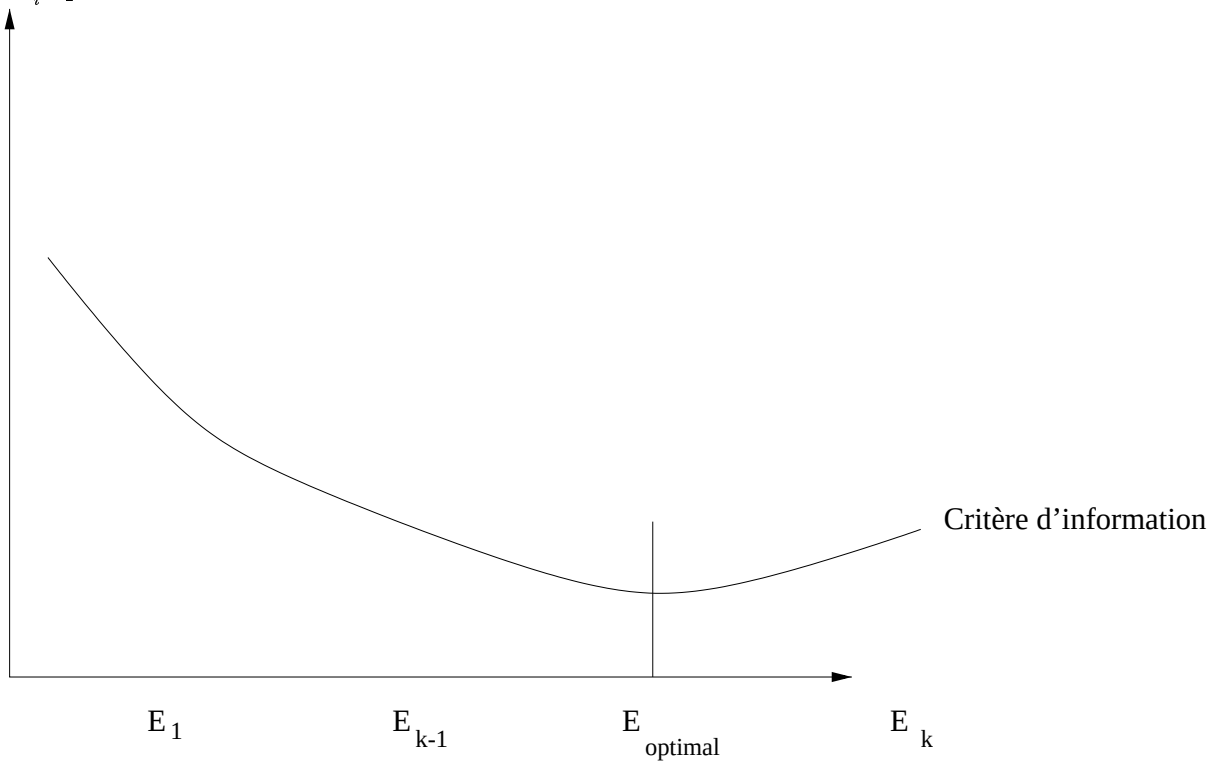
On choisira alors le MLP $F_{W_{k_{i^*}}}$ qui minimise le critère (2.7). On obtient donc la suite

$$BIC(h_{k_1}) \geq \cdots \geq BIC(h_{k_{i^*}}) < BIC\left(h_{k_{i+1}}\right).$$

On trouvera dans l'annexe, le détail sur l'implémentation de cette stratégie dans le programme d'estimation (REGRESS) développé au cours de cette thèse (cf Annexe A).

Cette recherche d'architecture correspond au schéma 2.5.

FIG. 2.5 – Minimisation du critère d'information, $E_{optimal}$ correspond à un MLP avec $\delta_{k_{i^*}}$ paramètres.



2.2.3.3 La méthode du "stepwise descendant" (cf Mangeas [49], Cottrell et al. [16])

Une fois le MLP dominant (F_{W_k}) déterminé, on a besoin d'affiner le modèle en retirant des connexions. Il faut donc déterminer quels paramètres sont inutiles (c'est-à-dire nuls). La technique est donc basée sur un test de nullité des paramètres.

On établit en effet (cf [64]) que l'estimateur des moindres carrés (ou bien celui introduit chapitre 4, associé à la vraisemblance gaussienne) est asymptotiquement gaussien (quand $n \longrightarrow \infty$), en fait on a

$$\sqrt{n}\left(\hat{W} - W\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \Sigma_0\right)$$

Tenter d'éliminer la connexion W_{ij} consiste alors à tester l'hypothèse $W_{ij} = 0$ contre l'hypothèse alternative $W_{ij} \neq 0$ suivant un test de Student (en fait un test gaussien, puisque le nombre d'observations est grand).

Enumérons l'ensemble des couples (i, j), soit l le numéro de (i, j), M le modèle courant et M^l le sous-modèle obtenu en annulant le poids W_{ij}. On suppose que M contient k paramètres.

On utilise les statistiques $(T_l)_{1\leq l\leq k}$ utilisées pour tester $W_{ij}=0$ contre $W_{ij}\neq 0$:

$$T_l = \frac{\hat{W}_{ij}}{\hat{\sigma}\left(\hat{W}_{ij}\right)} \tag{2.8}$$

avec

$$\hat{\sigma}\left(\hat{W}_{ij}\right) = \sqrt{(\Sigma_0)_{l,l}}$$

L'algorithme d'élagage s'écrit alors :

1. Calculer, pour chaque poids d'indice $l \in \{1,\cdots,k\}$, le quotient T_l correspondant à l'équation (2.8)
2. Déterminer l'indice
$$l^* = \arg \min_{l\in\{1,\cdots,k\}} T_l$$
3. Eliminer la connexion l^* .
4. Accepter l'élimination si le critère d'information (2.7) décroît.

On stoppe cet algorithme d'élagage dès que l'élimination d'une connexion ne fait plus décroître le critère d'information.

Ce type d'algorithme fait partie de la famille des "stepwise descendant", largement utilisés dans le domaine de la régression linéaire (cf [23]).

2.3 Estimation des fonctions MLP

On a vu que la modélisation statistique à l'aide de fonctions MLP implique la minimisation d'une fonction de risque (ou de coût) pour des données observées fixées. Les articles traitant des différentes méthodes d'optimisation des fonctions de risque sont innombrables et il est impossible d'en donner une vue exhaustive. Nous nous bornerons à présenter les méthodes les plus classiques issues de l'optimisation différentielle, puisque l'on a vu que le gradient était facile à calculer.

2.3.1 L'optimisation déterministe

Soit des observations $(x,y)_1^n = ((x,y)_1,\cdots,(x,y)_n) \in \left(\mathbb{R}^d\right)^n$. Soit

$$U_n(W) := f\left((x,y)_1^n, W\right)$$

une fonction réelle du vecteur paramètre W et des observations y_1^n, on cherche le paramètre W_0 qui réalise le minimum de $W \longmapsto U_n(W)$

$$W_0 = \arg\min U_n(W)$$

2.3.1.1 L'algorithme du premier ordre

Notons $\nabla U_n(W) = \left(\frac{\partial U_n(W)}{\partial W_i}\right)_{1 \leq i \leq D}$ le vecteur gradient par rapport aux paramètres de la fonction $U_n(W)$. Si on construit une suite de paramètres $(W_k)_{k \in \mathbb{N}^*}$ telle que

$$W_{k+1} = W_k - \varepsilon \nabla U_n(W_k)$$

avec $\varepsilon > 0$ suffisamment petit et que $\nabla U_n(W_k)$ reste borné, la suite W_n converge vers W^*, minimum local de la fonction $U_n(W)$. L'inconvénient principal de cette algorithme est sa lenteur. On utilise alors l'une des nombreuses techniques d'accélération.

2.3.1.2 Les algorithmes d'ordre supérieur

L'idée principal est de modifier la direction de descente, pour la replacer dans la "métrique" induite par la Hessienne. On construit une suite de paramètres $(W_k)_{k \in \mathbb{N}^*}$ telle que

$$W_{k+1} = W_k - \gamma_k H_k \nabla U_n(W_k)$$

où H_k est une matrice définie positive et γ_k le pas de descente à l'instant k. On trouvera les nombreuses façon de déterminer des suites $(\gamma_k)_{k \in \mathbb{N}^*}$ et $(H_k)_{k \in \mathbb{N}^*}$convenables dans Press [53]. On peut principalement dire que

1. H_k est une approximation de l'inverse de la matrice hessienne au point W_k
2. γ_k est un pas qui assure que la fonction à minimiser décroît bien à chaque itération, par exemple

   $$\gamma_k = \arg\min_{\gamma} U_n(W_k - \gamma H_k \nabla U_n(W_k))$$

   ou bien γ_k est tel que

   $$U_n(W_k - \gamma_k H_k \nabla U_n(W_k)) < U_n(W_k)$$

Une façon d'assurer de telles propriétés pour γ_k est d'utiliser une technique de minimisation unidimensionnelle telle que l'algorithme de Brent (cf [53]).

Parmi ces algorithmes "accélérés" citons :

- L'algorithme de gradient conjugué de Later, Polak et Ribière.
- L'algorithme quasi-newtonien de Broyden, Fletcher, Goldfarb et Shanno (BFGS).
- L'algorithme de Levenberg-Marquart

Ce sont des algorithmes classiques, dont on pourra trouver le détail dans [53].

2.3.2 L 'optimisation stochastique

L'algorithme d'optimisation stochastique met à jour le paramètre observation après observation. Il est essentiellement utilisé lorsque l'on dispose d'un grand nombre d'observations.

2.3.2.1 L'algorithme du gradient stochastique

Supposons que la dérivée de $U_n(W)$ s'écrive de façon additive :

$$\nabla U_n(W) = \frac{1}{n}\sum_{t=1}^{n} \nabla U(y_t, W)$$

On construit alors une suite de paramètres $(W_k)_{k\in\mathbb{N}^*}$ qui vérifie

$$W_{k+1} = W_k - \gamma_n \nabla U(y_k, W_k)$$

où γ_n est une suite de gains satisfaisant

$$\gamma_n > 0, \sum_{n=1}^{\infty} \gamma_n = \infty, \sum_{n=1}^{\infty} \gamma_n^2 < \infty \tag{2.9}$$

Par exemple, si U_n est la moyenne de l'erreur quadratique, on aura alors l'algorithme classique de "Least mean square" (LMS)

$$W_{k+1} = W_k - \frac{1}{n}\nabla\left((y_k - F_{W_k}(y_k))^2\right).$$

Cette technique assure sous de bonnes conditions que $(W_k)_{k\in\mathbb{N}^*}$ converge vers un minimum local de la valeur moyenne

$$\int U(y, W)\, dP_y.$$

2.3.2.2 Gradient stochastique accéléré

Le gradient stochastique est un algorithme plutôt lent, il requiert énormément de données pour converger. Il y a principalement deux façons de l'accélérer.

Modification de la direction de descente On peut modifier la direction de descente à l'aide d'une matrice définie positive comme dans la section 2.3.1.2

$$W_{k+1} = W_k - \gamma_n H_n \nabla U(y_k, W_k)$$

Le choix optimal pour H_k est l'inverse de la matrice d'information , i.e. $H_k^{-1} = I(W_k)$, où

$$I(W) = E\left[\nabla U(y, W)\nabla U(y, W)^T\right]$$

Cela correspond au gradient naturel d'Amari [2]. Le calcul de cette matrice d' information requiert une intégration numérique, ce qui est très coûteux en temps de calcul. On utilisera donc à la place une estimation de cette matrice, i. e.

$$H_k^{-1} = \frac{1}{k}\sum_{l=1}^{k} \nabla U(y_l, W_k)\nabla U(y_l, W_k)^T$$

La matrice H_k peut être estimée récursivement grâce au lemme d'inversion de matrice de Ricatti : (en notant $h_k = \nabla U(y_k, W_k)$)

$$H_{k+1} = \frac{1}{1-\gamma_k}\left(H_k - \frac{\gamma_k H_{k-1} h_k h_k^T H_{k-1}}{(1-\gamma_k) + \gamma_k h_k^T H_k h_k}\right)$$

On prouve alors que conditionnellement à la convergence de $(W_k)_{k\in\mathbb{N}^*}$ vers un minimum local, cette algorithme est asymptotiquement efficace (cf [24] ou [2]).

Moyennisation de l'algorithme Il suffit de poser

$$\begin{cases} W_{k+1} = W_k - \gamma_n \nabla U\left(y_k, W_k\right) \\ \overline{W}_{k+1} = \overline{W}_k + \frac{1}{n+1} W_{k+1} \end{cases}$$

On prouve alors que conditionnellement à la convergence de $\left(\bar{W}_k\right)_{k\in\mathbb{N}^*}$ vers un minimum local, cette algorithme est asymptotiquement efficace (cf [24] chap. 4 section III). D'après notre expérience, cet algorithme se comporte moins bien lors du régime transitoire (les premières itérations de l'algorithme) que l'algorithme précédent.

Chapitre 3

Initialisation et estimation par recuit simulé

Les algorithmes de gradient n'assurent que la convergence vers un minimum local de la fonction de risque associée à un problème d'estimation statistique. Une façon d'obtenir un minimum global est de recommencer les estimations sur de nombreuses initialisations aléatoires des paramètres. En effet, si on tire les poids suivant une loi uniforme dans l'ensemble des paramètres et que cet ensemble est compact, une infinité de tirages assure l'initialisation de l'algorithme dans le bon bassin d'attraction. Cependant, on peut espérer améliorer cette démarche naïve en utilisant une méthode de recuit simulé. C'est l'objet de cette section.

3.1 Fondements du recuit simulé

3.1.1 Une technique stochastique d'optimisation

Soit des observations $(x,y)_1^n = ((x,y)_1, \cdots, (x,y)_n) \in \left(\mathbb{R}^d\right)^n$. Soit

$$U_n(W) := f\left((x,y)_1^n, W\right)$$

une fonction réelle du vecteur paramètre W d'un MLP et des observations $(x,y)_1^n$, on cherche le paramètre W_0 qui réalise le minimum de $W \longmapsto U_n(W)$

$$W_0 = \arg\min_W U_n(W)$$

D'un point de vue mathématique, le recuit simulé est un algorithme stochastique ayant pour but de trouver W_0.

Supposons dans un premier temps que l'ensemble des paramètres possibles est un ensemble fini E[1]. On appelle E l'espace des configurations (paramètres). Nous appellerons $U_n(W)$ la fonction d'*énergie*.

Un recuit simulé séquentiel est un algorithme sur E qui génère une séquence aléatoire $W_t \in E$ de configurations qui vont se concentrer quand $t \longrightarrow \infty$ sur l'ensemble des minima absolus de U_n.

Fixons une matrice stochastique de transitions $q(i,j)$ sur $E \times E$ avec q symétrique et irréductible. Définissons $V_i = \{j \in E \mid \quad q(i,j) > 0\}$ l'ensemble des voisinages de i. Nous avons pour $i, j \in E$

$$i \in V_j \Leftrightarrow j \in V_i$$

Fixons maintenant une séquence décroissante de nombres $T_t > 0$, $t \in \mathbb{N}$. T_t s'appelle une température et la suite $(T_t)_{t\in\mathbb{N}}$ un schéma de températures, ou règle de refroidissement.

On définit maintenant un algorithme stochastique définissant une suite $W_1, \ldots, W_t, \ldots$ de configurations aléatoires $W_t \in E$ de la manière suivante :

- Choisissons premièrement dans V_{W_t} un voisin aléatoire ξ_t de W_t tels que
$$P(\xi_t = j \,|W_t) = q(W_t, j)$$
- Si $U_n(\xi_t) \leq U_n(W_t)$ alors, posons $W_{t+1} = \xi_t$
- Si $U_n(\xi_t) > U_n(W_t)$ alors soit :
$$p = \exp\left[-\frac{1}{T_t}(U_n(\xi_t) - U_n(W_t))\right]$$
On fait un choix aléatoire entre les deux décisions $(W_{t+1} = \xi_t)$ et $(W_{t+1} = W_t)$, avec
$$P(W_{t+1} = \xi_t) = p$$
$$P(W_{t+1} = W_t) = 1 - p$$

La suite $(W_t)_{t\in\mathbb{N}^*}$ est alors une chaîne de Markov inhomogène dans le temps, avec une fonction de transition définie par

$$P(W_{t+1} = j \,|W_t = i) = p_{T_t}(i,j)$$

où l'on a posé :

$$p_T(i,j) = q(i,j)exp\left[-\frac{1}{T}(U_n(j) - U_n(i))^+\right] \qquad \text{si} \qquad j \neq i$$

et

$$p_T(i,i) = 1 - \sum_{i\neq j} p_T(i,j)$$

Cet algorithme est appelé un algorithme de *Recuit Simulé* séquentiel et est associé à la fonction d'énergie U_n, la règle de refroidissement $(T_t)_{t\in\mathbb{N}^*}$ et la matrice d'exploration q.

[1] Généralement l'espace des paramètres est inclus dans un compact réel. On peut cependant considérer que, lorsqu'on calcule sur un ordinateur, on est dans un espace fini puisque la machine ne peut représenter qu'un nombre fini de réels.

3.1.1.1 Convergence du Recuit Simulé

Une configuration $W_0 \in E$ est dite "minimum global" de l'énergie U_n si :

$$U_n(W_0) = \inf_{W \in E} U_n(W)$$

Appelons E_{min} l'ensemble des configurations qui sont des minima globaux pour U_n. Si on suppose qu'il existe un entier M tel que $\forall (W, \xi) \in E^2, q^M(W, \xi) > 0$ et si on note :

$$\Delta = \max\left(U_n(W) - U_n(\xi); (W, \xi) \in E, q(W, \xi) > 0\right),$$

si la température suit un schéma logarithmique :

$$T_t = \frac{T_0}{log(t+1)},\ t \in \mathbb{N}^*$$

Hajek [32] montre qu'il suffit que $T_0 > M \times \Delta$ pour que :

$$\lim_{t \to \infty} P\left(W_t \in E_{min}\right) = 1$$

On peut donc dire que, si la température initiale est suffisamment élevée, la suite $(W_t)_{t \in \mathbb{N}^*}$ converge en probabilité vers l'ensemble des minima globaux.

3.2 Implémentation du recuit simulé pour un MLP

3.2.1 Simplification spécifique aux MLP

Nous limitons notre étude aux MLP dont la sortie est linéaire, c'est-à-dire dont la fonction d'activation des unités de sortie est l'identité[2]. De plus, on ne considère que des MLP avec une couche cachée, mais la généralisation à plusieurs couches est facile. Supposons que le MLP ait m entrées, l unités cachées et s sorties, pour des poids entre les entrées et la couche cachée fixés, on associe à chaque $x_t \in \mathbb{R}^m$ un vecteur C_t de dimension l dont les composantes sont les valeurs des unités cachées après propagation de x_t :

$$C_t = \left(\tanh\left(\sum_{k=1}^{m} W_{jk} x_t(k) + W_{j0}\right)\right)_{1 \leq j \leq l}.$$

On note C la matrice dont les lignes sont les n vecteurs $(C_t)_{t=1,..,n}$.

Soit α la matrice dont les composantes sont les poids de sortie. Pour trouver les poids de sortie optimaux, on doit minimiser l'expression :

$$\|C \times \alpha - y\|^2$$

[2]Si la fonction de sortie n'est pas l'identité mais une fonction g bijective continue, il suffit de poser $Y'_t = g^{-1}(Y_t)$ pour pouvoir appliquer le même raisonnement

où y est la matrice dont les lignes sont les n sorties y_t et $\|.\|$ est la norme euclidienne. Les poids de sortie α peuvent donc être déterminés de façon optimale par régression linéaire. Cette remarque diminue le nombre de paramètres à déterminer et améliore de façon spectaculaire les résultats du recuit simulé.

Dans la suite, le vecteur paramètre représentera donc uniquement les poids entre les entrées et les unités cachées du MLP.

3.2.2 L'espace d'état

Ici $W \in \mathbb{R}^B$, ou B est le nombre de poids du MLP entre l'entrée et la couche cachée. On notera W_{ij} le poids reliant l'entrée j et l'unité cachée i. Classiquement, le nombre de configurations possibles des paramètres doit être fini (même si il est de grand cardinal). On commence par restreindre l'ensemble de nos paramètres à un compact de $\mathbb{R}^n$, ensuite on envisage deux méthodes

- Discrétisation des poids :
  Les poids ne peuvent prendre qu'un nombre fini de valeurs.
- Probabilité de proposition continue :
  La probabilité de proposition est continue (par exemple gaussienne). Ici l'espace n'est plus fini (bien qu'il le soit pour le calcul par ordinateur). On garde cependant un schéma comparable au recuit simulé sur un espace fini.

En pratique on travaillera avec les espaces suivants

1. **Discrétisation uniforme :**

   Les poids sont discrétisés de la façon la plus simple, c'est-à-dire qu'ils ne peuvent prendre que des valeurs de la forme : $k \times V$ où V est un paramètre constant (le pas de la discrétisation) et $k \in \mathbb{Z}$.

2. **Discrétisation géométrique :**

   L'écart entre les points augmente géométriquement au fur et à mesure que l'on s'éloigne de zéro. Le facteur multiplicatif est $(1 + V)$. Ainsi les valeurs possibles pour les poids sont de la forme : $\pm I \times (1 + V)^k$, avec I valeur initiale du pas. On fait varier moins vite les poids qui sont plutôt proches de zéro, car plus ils sont grands, plus ils risquent de saturer les unités cachées. En effet, supposons que pour $x \in \mathbb{R}^m$ on ait

   $$\sum_{k=1}^{m} W_{jk}x + W_{j0} = 2$$

   alors

   $$\tanh\left(\sum_{k=1}^{m} W_{jk}x + W_{j0}\right) = 0.964.$$

Soient de nouveaux poids W' tels que

$$\sum_{k=1}^{m} W'_{jk}x + W'_{j0} = 3$$

alors

$$\tanh\left(\sum_{k=1}^{m} W'_{jk}x + W'_{j0}\right) = 0.995,$$

soit une différence de 0.031 pour la valeur d'activation de cette unité cachée. Maintenant si

$$\sum_{k=1}^{m} W_{jk}x + W_{j0} = 0$$

alors

$$\tanh\left(\sum_{k=1}^{m} W_{jk}x + W_{j0}\right) = 0$$

et si de nouveaux poids W' sont tels que

$$\sum_{k=1}^{m} W'_{jk}x + W'_{j0} = 1$$

alors

$$\tanh\left(\sum_{k=1}^{m} W'_{jk}x + W'_{j0}\right) = 0.761,$$

soit une différence de 0.761 pour la valeur d'activation de cette unité cachée. Il faut une variation plus importante des poids pour une modification significative de la valeur des unités cachées au fur et à mesure que l'on s'éloigne de 0.

3. **Espace des poids continu :**
   Les poids prennent leur valeur dans $\mathbb{R}^B$. On proposera de nouveaux poids à l'aide de la loi normale.

Enfin, pour garder des temps de calculs raisonnables, on se restreindra à $|W_{ij}| \leq 100$

3.2.3 Le schéma de températures

Un schéma de températures logarithmique converge très lentement, en effet pour que $T_t \leq \frac{T_0}{100}$ il faut que $t \geq 2.68 \times 10^{43}$, ce qui est déraisonnable. C'est pourquoi le schéma de températures le plus employé est le schéma exponentiel, c'est-à-dire $T_t = T_0 \times \alpha^t$, avec $0 < \alpha < 1$. Généralement la température reste constante sur des paliers de longueur fixe, on utilisera ce schéma dans tous les exemples traités après.

3.2.4 L'algorithme

On décrit ici l'algorithme général, commun à toutes les espaces de poids possibles. Les différents espaces de paramètres apparaissent dans la fonction "*perturb*" qui calcule la nouvelle proposition de poids.

3.2.4.1 Algorithme commun

Procédure RECUIT SIMULE
DEBUT : **Initialise** (poids courants, T_0, Erreur courante, Meilleure Erreur)
Pour $i = 0$; i < (Nombre températures)
Pour $j = 0$; j < (Longueur palier)
Perturb(poids=W)
Calcul(E_W) ;
Si $E_W <$ (Erreur courante) **Alors**
Erreur courante := E_W
Poids courants = W
Si Erreur courante < Meilleure Erreur **Alors**
Meilleure Erreur := Erreur courante
Meilleurs Poids = Poids courants
Sinon
p=exp $\left(\frac{Erreurcourante - E_W}{T_i}\right)$
Si $\mathcal{U}_{[0;1]} < p$
Erreur courante = E_W
Poids courants = W
Sinon on garde les poids courants
Prochain(j)
Calcule prochaine température T_{i+1}, **Prochain**(i)
FIN

Cette fonction fournit les meilleurs poids et la meilleure valeur de la fonction d'erreur.

Cette procédure est simple, mais elle requiert pour être mise en pratique plusieurs réglage qui sont :

- La température initiale T_0
- La longueur des paliers
- L'amplitude des voisinages

Déterminer la température initiale est un réel problème. Heuristiquement, si elle est trop petite, la région explorée par le recuit simulé est petite et on risque de ne pas pouvoir atteindre un vrai minimum global, si elle est trop grande l'algorithme explore un vaste domaine de paramètres et converge très lentement.

3.2.4.2 Les différentes fonctions "*perturb*"

Cette fonction choisit les nouveaux poids candidats $\widetilde{W_{ij}}$. Dans les trois cas, on choisit au hasard uniformément sur les indices i, j un poids W_{ij}. Puis on tire une variable aléatoire D qui peut prendre comme valeurs soit -1, soit 1, avec la même probabilité $\frac{1}{2}$. Ensuite on aura :

Discrétisation uniforme

$$\widetilde{W_{ij}} = W_{ij} + V \times D$$

Cela implique que chaque point a deux voisins (sauf aux bornes où il n'a qu'un seul voisin)

Discrétisation géométrique

$$\widetilde{W_{ij}} = W_{ij} \times (1 + V)^D,$$

sauf si $|W_{ij}| = I$ où I est la valeur absolue initiale du paramètre, dans ce cas on pose :

$$\widetilde{W_{ij}} = -W_{ij}$$

Cela implique aussi que chaque point a deux voisins, sauf aux bornes.

Proposition continue

$$\widetilde{W_{ij}} = W_{ij} + V \times \varepsilon$$

où ε suit une loi $\mathcal{N}(0, 1)$.

3.2.4.3 Schéma de températures

Le schéma exponentiel est $T_t = T_0 * \alpha^t$ avec $0 < \alpha < 1$. La température restera constante sur des paliers de longueur L. Dans les exemples ci-dessous, on prendra toujours 100 températures avec $\alpha = 0.95$, on aura donc $T_{100} = \frac{T_0}{169}$.

3.3 Un exemple d'approximation de fonction

3.3.1 La fonction

On considère un discrétisation de la fonction $\sin\left(\frac{1}{x}\right)$ pour $x \in [-1; 1]$ en prenant 201 points équi-espacés :

$$(x_1 = -1, x_2 = -0.99, \cdots, x_{201} = 1)$$

La fonction considérée n'est pas continue mais ils existe une infinité de fonctions continues passant par les 201 points ainsi déterminés. Le théorème d'approximation universelle des perceptrons multicouches assure que le MLP peut apprendre une de ces fonctions, pourvu que le nombre d'unités cachées soit suffisant. La figure 3.1 montre que les variations de cette fonction au voisinage de zéro la rendent difficile à apprendre au MLP.

Le graphique 3.2 représente la fonction de coût suivant deux poids pour un MLP avec une entrée et dix unités cachées, c'est-à-dire :

$$E_\theta = \sum_{t=1}^{201} \left(F_W(x_t) - \sin\left(\frac{1}{x_t}\right) \right)^2$$

Pour créer cette surface d'erreur, on a fait varier les poids entre l'entrée et les deux premières unités cachées entre -20 et 20 avec un pas de discrétisation de 0.2 , les autres poids restant constants. Il s'agit donc ici des poids W_{11} et W_{21}. Les poids de sortie sont déterminés par régression linéaire. On peut remarquer qu'il existe plusieurs minima locaux, alors même que cette fonction est restreinte à deux poids. Il est donc justifié d'utiliser un recuit simulé pour essayer de les éviter. Les pics au centre proviennent de la régression linéaire lorsque la matrice C^tC (voir 3.2.1) devient presque singulière.

FIG. 3.1 – La fonction à approximer

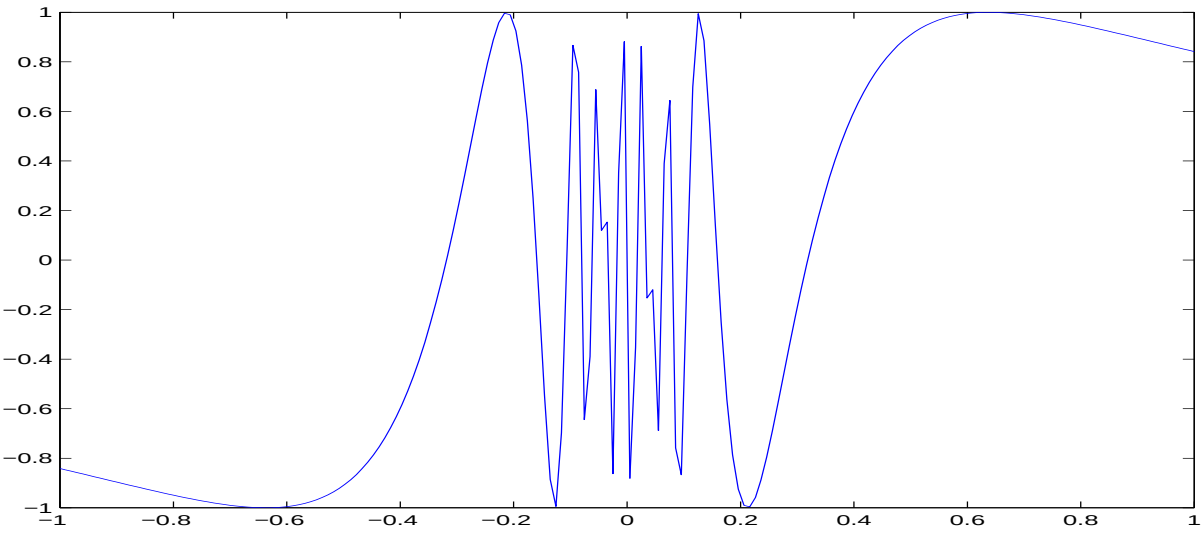
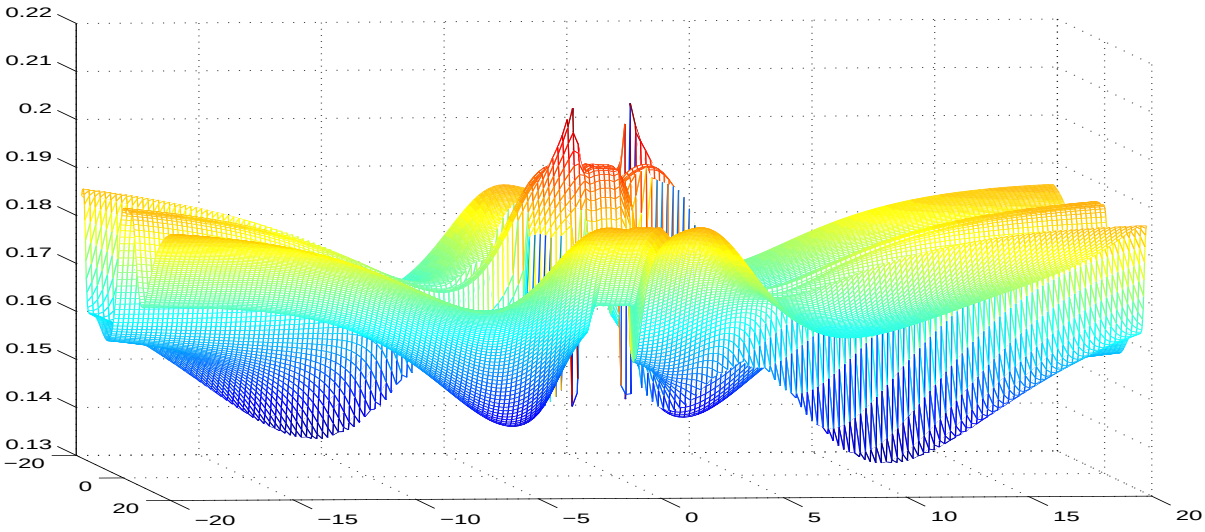


FIG. 3.2 – Fonction d'erreur suivant deux poids



3.3.2 Estimation par gradient et initialisation aléatoire

3.3.2.1 Comparaison de différents algorithmes

On utilise un MLP avec une entrée, dix unités cachées, une sortie scalaire (la fonction de sortie est l'identité) pour approximer la discrétisation de la fonction $\sin\left(\frac{1}{x}\right)$ sur 200 points. La fonction de coût est l'erreur moyenne quadratique.

On choisit les poids initiaux en les tirant suivant une loi uniforme $\mathcal{U}[-0.5; 0.5[$. En pratique on change le tirage aléatoire de l'ordinateur en changeant la "graine" du générateur aléatoire. On a recommencé l'estimation dix fois avec un tirage initial différent à chaque fois.

On va comparer, sur cette fonction, les différentes implémentations du recuit simulé et différents algorithmes de gradient (voir section 2.3.1.2) , c'est-à-dire

- Le gradient conjugué
- Le BFGS
- Le Levenberg-Marquart
- Le gradient stochastique

Pour tous les algorithmes de gradient, on fera au maximum 1000 itérations.

Le gradient conjugué On redémarre l'algorithme (c'est-à-dire qu'on réinitialise le terme de conjugaison) si l'erreur commence à décroître trop lentement. On obtient les résultats suivants :

TAB. 3.1 – Estimation par gradient conjugué

Graine	0	1	2	3	4
Erreur finale	0.0682	0.0578	0.1207	0.0614	0.0633
Graine	5	6	7	8	9
Erreur finale	0.2399	0.1207	0.0744	0.2398	0.0690

Remarque 4 *Il est utile de redémarrer l'algorithme, car s'il converge vers un mauvais minimum local, le redémarrage donne un choc stochastique au gradient, qui lui permet souvent d'en sortir, à titre de comparaison on donne les résultats sans redémarrage :*

TAB. 3.2 – Estimation par gradient conjugué sans redémarrage

Graine	0	1	2	3	4
Erreur finale	0.239	0.239	0.239	0.239	0.239
Graine	5	6	7	8	9
Erreur finale	0.239	0.239	0.239	0.239	0.120

L'algorithme BFGS On redémarre l'algorithme dans les même conditions que précédemment. On obtient les résultats suivants :

TAB. 3.3 – Estimation par BFGS

Graine	0	1	2	3	4
Erreur finale	0.059	0.070	0.064	0.050	0.051
Graine	5	6	7	8	9
Erreur finale	0.239	0.061	0.059	0.239	0.052

L'algorithme de Levenberg-Marquart On ne redémarre pas l'algorithme car l'approximation de la Hessienne n'est pas itérative. De plus la recherche unidimensionnelle est différente des deux algorithmes précédents (cf. [53]).

TAB. 3.4 – Estimation par levenberg-Marquart

Graine	0	1	2	3	4
Erreur finale	0.066	0.119	0.122	0.067	0.071
Graine	5	6	7	8	9
Erreur finale	0.242	0.082	0.057	0.237	0.076

L'algorithme du gradient stochastique On choisit au hasard uniformément un couple de point $(X_t, \sin(1/X_t))_{t=1,\ldots,201}$, puis on optimise suivant la direction opposée au gradient en ce point. Pour un pas contant $\gamma = 0.01$, on recommence ainsi 1 million de fois, on obtient les résultats suivants :

TAB. 3.5 – estimation par gradient stochastique

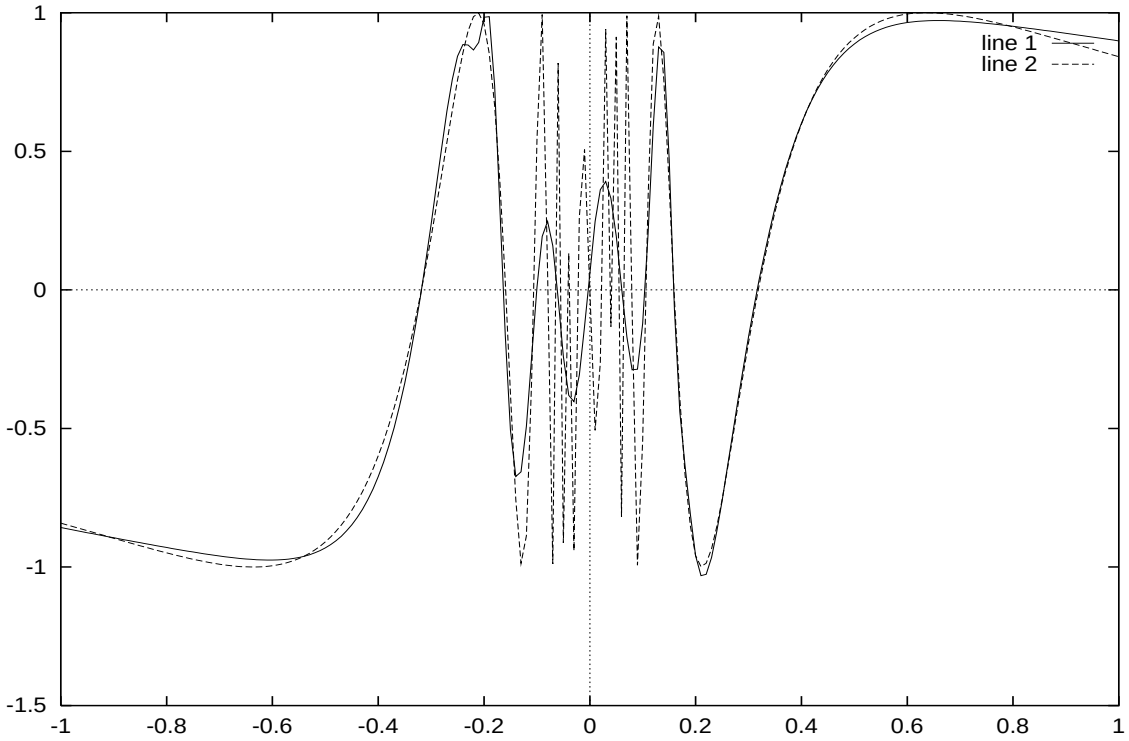
Graine	0	1	2	3	4
Erreur finale	0.197	0.236	0.228	0.242	0.239
Graine	5	6	7	8	9
Erreur finale	0.239	0.237	0.238	0.241	0.150

3.3.2.2 Conclusion de cette étude préliminaire

A part le gradient stochastique, les 3 autres algorithmes donnent des résultats à peu près similaires. On remarque que les résultats dépendent de l'initialisation, cela justifie la recherche d'une bonne initialisation. Par contre le choix de l'algorithme de calcul influe plus sur la vitesse que sur l'aptitude à éviter les minima locaux.

Représentation graphique de la fonction apprise Le graphique 3.3 montre la valeur de la fonction MLP sur les 200 points pour le meilleur de MLP optimisé par une méthode différentielle. La fonction apprise est plus régulière que la fonction à apprendre. On va utiliser le recuit simulé pour essayer d'approcher le minimum absolue de la fonction de coût.

FIG. 3.3 – Estimation par BFGS. Fonction du meilleur MLP (ligne 1 : Approximation du MLP ; ligne 2 : Fonction à apprendre)



3.3.3 Estimation par les différents recuits simulés

3.3.3.1 La méthodologie

Nous utilisons l'algorithme du recuit simulé pour essayer de trouver des paramètres du MLP qui soient plus proches du minimum global de l'erreur quadratique moyenne. On a vu que contrairement aux méthodes de gradient, dont l'utilisation est pratiquement automatique, il faut maintenant trouver le bon réglage pour :

- La température initiale T_0
- Le voisinage des propositions V

Nous allons, par tâtonnement, essayer de régler au mieux ces paramètres pour chacune des fonctions de propositions correspondant à :

- La discrétisation uniforme
- La discrétisation géométrique
- La continuité des paramètres

On fournit ici les tableaux de résultats des 3 fonctions de propositions envisagées, en fixant le nombre d'évaluations de l'erreur (it.) à dix mille (longueur des paliers égale à 100), puis à cent mille (longueur des paliers égale à 1000).

Les meilleurs résultats sont notés en gras, les colonnes des tableaux correspondent aux différentes températures de départ, les lignes aux différentes valeurs de la variable V.

3.3.3.2 Les résultats suivant les fonctions de proposition

Discrétisation uniforme Le meilleur résultat correspond à une erreur quadratique moyenne de 0.036 (pour des paliers de longueur 1000). C'est meilleur que pour les différents gradients. On remarque que pour ce type de discrétisation, il faut un pas de discrétisation assez grand (entre 1 et 5), et une température de départ petite (entre 0.01 et 0.1).

TAB. 3.6 – Discrétisation uniforme, paliers de longueur 100

10^4 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.182	0.195	0.195	0.236	0.230	0.196	0.235	0.195
$V = 0.05$	0.139	0.140	0.140	0.140	0.140	0.141	0.141	0.141
$V = 0.1$	0.095	0.123	0.091	0.110	0.139	0.136	0.140	0.133
$V = 0.5$	0.074	0.074	0.076	0.088	0.097	0.094	0.097	0.103
$V = 1.0$	0.048	**0.045**	0.071	0.081	0.072	0.092	0.099	0.082
$V = 5.0$	0.048	0.056	0.054	0.097	0.097	0.082	0.112	0.095
$V = 10$	0.057	0.061	0.063	0.076	0.088	0.103	0.092	0.095
$V = 50$	0.215	0.215	0.215	0.215	0.215	0.215	0.215	0.215

TAB. 3.7 – Discrétisation uniforme, paliers de longueur 1000

10^5 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.139	0.140	0.142	0.146	0.142	0.169	0.176	0.143
$V = 0.05$	0.091	0.077	0.086	0.095	0.101	0.123	0.134	0.096
$V = 0.1$	0.062	0.076	0.083	0.079	0.092	0.097	0.091	0.094
$V = 0.5$	0.046	0.044	0.050	0.074	0.063	0.113	0.079	0.098
$V = 1.0$	**0.036**	0.059	0.061	0.081	0.088	0.093	0.085	0.089
$V = 5.0$	0.046	0.037	0.040	0.062	0.071	0.087	0.086	0.065
$V = 10$	0.046	0.051	0.057	0.060	0.075	0.090	0.088	0.098
$V = 50$	0.215	0.215	0.215	0.215	0.215	0.215	0.215	0.215

Discrétisation géométrique La meilleure erreur est encore de 0.036 (pour des paliers de longueur 1000). On aurait pu croire que cette discrétisation était plus adaptée au MLP, puisque plus le poids est grand, plus les unités cachées sont proches de la saturation, plus il faut ajouter un grand pas pour avoir un changement significatif de la fonction représentée par le MLP.

Cependant ce raisonnement ne s'applique qu'à un seul poids et les poids dans leur ensemble peuvent se compenser les uns les autres. Ainsi cette méthode n'améliore pas les résultats obtenus par la discrétisation uniforme.

TAB. 3.8 – Discrétisation géométrique, paliers de longueurs 100

10^4 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.240	0.240	0.240	0.240	0.240	0.240	0.240	0.240
$V = 0.05$	0.240	0.239	0.239	0.239	0.239	0.240	0.240	0.240
$V = 0.1$	0.239	0.239	0.239	0.158	0.145	0.239	0.239	0.239
$V = 0.5$	0.086	0.077	0.074	0.091	0.091	0.107	0.105	0.099
$V = 1.0$	0.067	0.094	**0.059**	0.066	0.098	0.101	0.103	0.105
$V = 5.0$	0.060	0.060	0.060	0.062	0.064	0.067	0.067	0.068
$V = 10$	0.114	0.114	0.114	0.118	0.120	0.126	0.121	0.126
$V = 50$	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182

TAB. 3.9 – Discrétisation géométrique, paliers de longueurs 1000

10^5 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.240	0.240	0.240	0.240	0.240	0.240	0.240	0.240
$V = 0.05$	0.141	0.142	0.142	0.134	0.112	0.113	0.143	0.109
$V = 0.1$	0.055	0.059	0.051	0.088	0.103	0.093	0.105	0.101
$V = 0.5$	**0.036**	0.045	0.047	0.080	0.081	0.089	0.092	0.087
$V = 1.0$	0.056	0.051	0.052	0.073	0.086	0.091	0.088	0.096
$V = 5.0$	0.060	0.060	0.060	0.063	0.063	0.067	0.064	0.064
$V = 10$	0.114	0.114	0.114	0.114	0.114	0.115	0.118	0.116
$V = 50$	0.182	0.182	0.182	0.182	0.182	0.182	0.182	0.182

Proposition continue La meilleure erreur est ici de 0.026 (cf tableau 3.11). Cette technique donne donc les meilleurs résultats empiriques. Une bonne stratégie semble donc de choisir cette méthode, avec un écart type grand pour la gaussienne ($V = 10$), et une température de départ petite (entre 0.001 et 0.005). Si on veut augmenter la température de départ T_0, il faudra certainement augmenter aussi le nombre d'itérations, en jouant sur le nombre de températures et la longueur des paliers. On donne à titre de comparaison les résultats obtenus sans régression linéaire avec une proposition continue, c'est-à-dire en ajoutant les coefficients de la couche de sortie aux paramètres. Les résultats sont alors plus mauvais qu'avec les méthodes de gradient.

TAB. 3.10 – Proposition continue, paliers de longueurs 100

10^4 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.195	0.196	0.239	0.195	0.218	0.239	0.196	0.238
$V = 0.05$	0.150	0.140	0.140	0.141	0.145	0.141	0.141	0.143
$V = 0.1$	0.121	0.109	0.140	0.132	0.136	0.138	0.130	0.131
$V = 0.5$	0.069	0.072	0.075	0.097	0.095	0.089	0.089	0.103
$V = 1.0$	0.060	0.043	0.071	0.070	0.092	0.078	0.104	0.111
$V = 5.0$	0.051	0.050	0.072	0.071	0.089	0.088	0.091	0.117
$V = 10$	0.040	0.042	0.054	0.068	0.074	0.106	0.088	0.086
$V = 50$	**0.037**	0.041	0.053	0.056	0.075	0.101	0.108	0.108

TAB. 3.11 – Proposition continue, paliers de longueurs 1000

10^5 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.137	0.140	0.143	0.145	0.145	0.145	0.141	0.147
$V = 0.05$	0.088	0.082	0.114	0.116	0.105	0.135	0.117	0.130
$V = 0.1$	0.070	0.077	0.083	0.089	0.094	0.091	0.091	0.101
$V = 0.5$	0.038	0.058	0.055	0.059	0.072	0.089	0.083	0.090
$V = 1.0$	0.035	0.032	0.054	0.070	0.081	0.086	0.087	0.089
$V = 5.0$	0.026	0.032	0.038	0.062	0.066	0.088	0.091	0.093
$V = 10$	0.027	**0.026**	0.041	0.058	0.071	0.084	0.076	0.088
$V = 50$	0.028	0.037	0.034	0.061	0.064	0.082	0.086	0.090

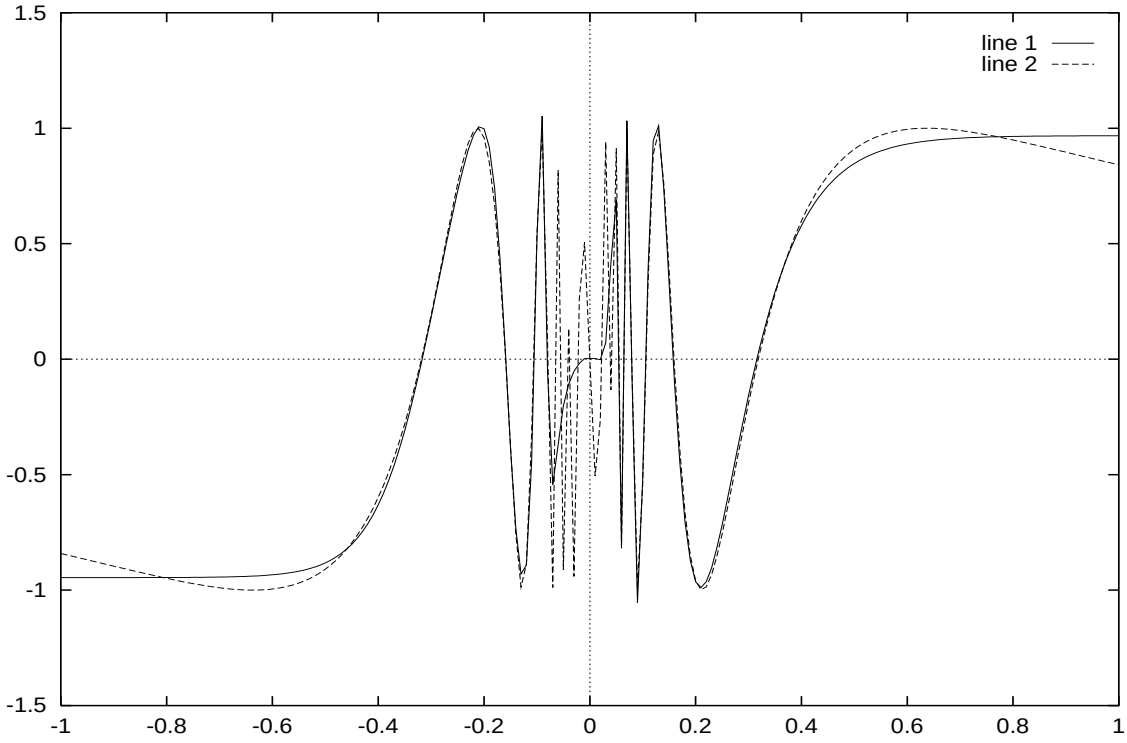
TAB. 3.12 – Proposition continue, sans régression linéaire, paliers de longueurs 1000

10^5 it.	$T_0 = 0.01$	$T_0 = 0.05$	$T_0 = 0.1$	$T_0 = 0.5$	$T_0 = 1.0$	$T_0 = 5$	$T_0 = 10$	$T_0 = 50$
$V = 0.01$	0.094	0.066	**0.060**	0.098	0.093	0.074	0.065	0.078
$V = 0.05$	0.143	0.105	0.070	0.138	0.105	0.133	0.078	0.085
$V = 0.1$	0.152	0.127	0.166	0.105	0.126	0.120	0.089	0.100
$V = 0.5$	0.237	0.136	0.123	0.194	0.155	0.204	0.155	0.155
$V = 1.0$	0.239	0.172	0.155	0.216	0.208	0.157	0.147	0.157
$V = 5.0$	0.241	0.219	0.149	0.188	0.200	0.266	0.261	0.195
$V = 10$	0.247	0.187	0.173	0.171	0.239	0.300	0.293	0.165
$V = 50$	0.247	0.269	0.241	0.285	0.272	0.282	0.264	0.255

Représentation graphique de la fonction apprise et de l'erreur Le graphique 3.4 montre la valeur de la fonction MLP sur les 200 points pour le meilleur des MLP optimisés par le recuit simulé. Comme on peut le voir, la fonction apprise est un peu plus irrégulière que la fonction apprise par le gradient, même si les oscillations proches de zéro restent difficiles à modéliser.

Un MLP avec seulement 10 unités cachées a déjà de bonne capacités d'approximation, mais il faut un algorithme d'apprentissage qui évite le plus possible les mauvais minima locaux.

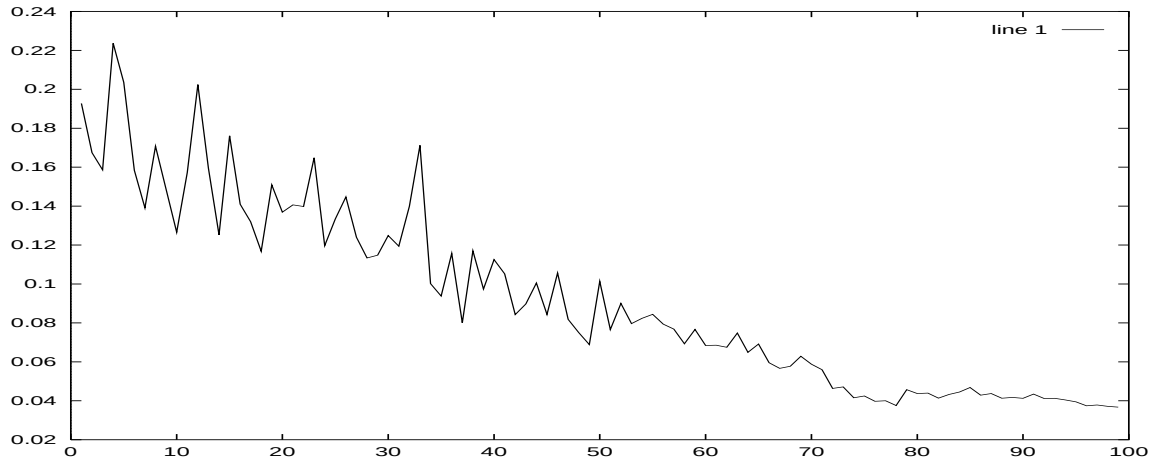
FIG. 3.4 – Meilleure estimation par recuit simulé (ligne 1 : Approximation du MLP ; ligne 2 : Fonction à apprendre)



La caractéristique essentielle du recuit simulé est que l'erreur peut remonter, c'est ce qui permet de visiter différents minima et d'augmenter ainsi les chances de trouver un minimum global. Par contre, c'est aussi pour cette raison que le recuit simulé est bien plus lent qu'une méthode de gradient pour converger vers le minimum d'un bassin d'attraction.

Le graphique 3.5 montre la valeur de l'erreur courante à la fin de chaque palier.

FIG. 3.5 – Evolution de l'erreur obtenue avec le recuit simulé



3.3.3.3 Apprentissage par une technique de gradient avec initialisation par recuit simulé

Les résultats précédents montrent que, sur cet exemple, le recuit simulé améliore l'apprentissage. Néanmoins c'est une technique très coûteuse en temps de calcul, on va donc essayer de l'utiliser ici uniquement pour initialiser les poids.

On commence donc tous les apprentissages de gradient par un recuit simulé sur un espace d'états continu de 100 températures, sans palier, avec une température initiale assez haute de 1 pour explorer largement le domaine et un écart type de 10.

Comme on peut le voir sur le tableau 3.13, cette initialisation ne permet pas à l'algorithme de trouver un meilleur minimum que par initialisation aléatoire. Par contre, elle limite le risque de converger vers un mauvais minimum local pour une seule initialisation.

TAB. 3.13 – Estimation de la série, suivant les différentes graines initiales, pour une initialisation par recuit simulé

Algo.	g 0	g 1	g 2	g 3	g 4	g 5	g 6	g 7	g 8	g 9
Grad. conj.	0.075	0.086	0.064	0.099	0.069	0.090	0.066	0.072	0.057	0.083
BFGS	0.065	0.064	0.067	0.098	0.068	0.088	0.056	0.071	0.057	0.073
Lev. Mar.	0.058	0.061	0.069	0.105	0.066	0.090	0.077	0.059	0.061	0.070
Grad. stoch.	0.066	0.119	0.092	0.075	0.094	0.106	0.0983	0.0766	0.107	0.117

3.3.3.4 Conclusion de l'estimation de fonction

On a montré sur un exemple que les techniques de recuit simulé peuvent amélioré l'apprentissage d'un MLP par rapport à une technique de gradient. Cette étude empirique montre que la meilleure implémentation semble être une méthode avec une probabilité de proposition continue gaussienne sur l'ensemble des paramètres, un écart type plutôt grand ($V = 5$ ou $V = 10$) et une température de départ plutôt basse ($T_0 = 0.01$ ou $T_0 = 0.05$). Bien sûr, si on a le temps d'attendre plus longtemps on pourra toujours essayer une température plus élevée en augmentant aussi le nombre d'itérations, ce qui augmentera les chances de trouver un minimum global.

Si on veut un résultat rapide, l'initialisation par recuit simulé, qui permet d'éviter à chaque fois les mauvais minima, est peut être un bon compromis entre performance et temps de calcul. En effet comme le recuit simulé visite un vaste domaine de paramètres, on peut espérer qu'il va visiter un bon bassin d'attraction pour débuter l'algorithme de gradient.

Néanmoins, comme on ne peut pas savoir si on est dans un bon bassin d'attraction, on est souvent obligé de continuer le recuit simulé si on veut vraiment un bon minimum et les temps de calculs peuvent devenir rapidement prohibitifs.

3.4 Application aux série temporelles

On va maintenant appliquer la technique du recuit simulé pour la prévision d'une série temporelle. Au vu des résultats précédents, la technique choisie sera la méthode avec une proposition gaussienne, un paramètre $V = 10$ et un température initial $T_0 \leq 0.005$.

3.4.1 La série

Nous allons simuler un modèle autorégressif non linéaire d'ordre 1 : on se donne un nombre X_0 et on construit récursivement les observations par la formule :

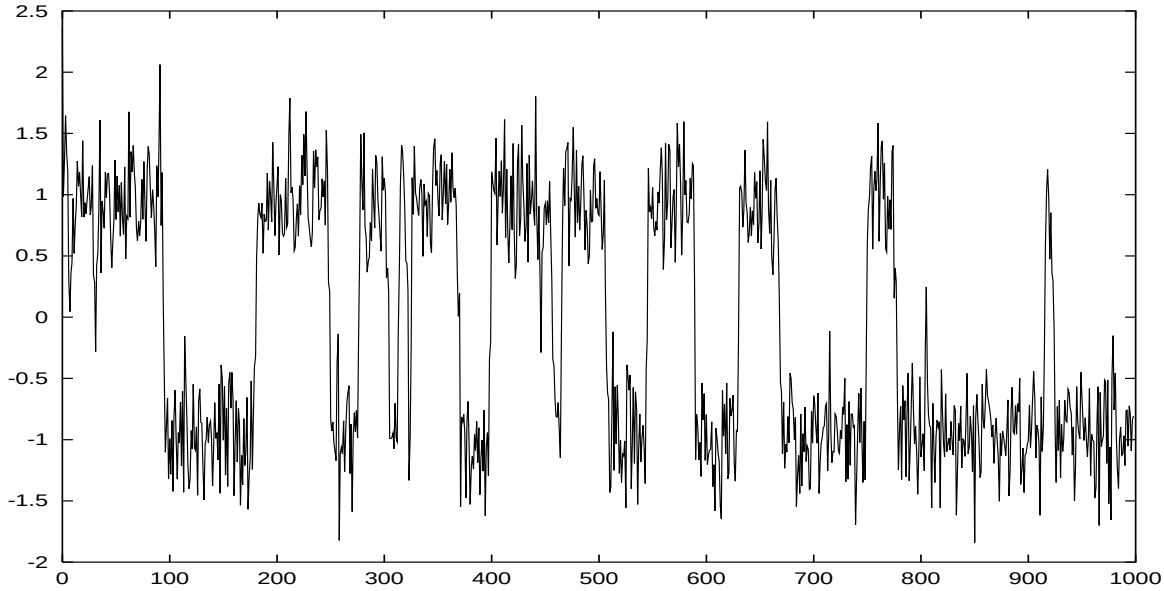
$$X_{t+1} = F_W(X_t) + \varepsilon_{t+1}$$

où F_W est une fonction perceptron avec 1 entrée, 10 unités cachées, 1 sortie linéaire.

Afin de générer une série qui ne soit pas trop facilement modélisable par une méthode de gradient classique, nous choisirons une fonction suffisamment irrégulière. Nous réutiliserons le MLP ayant le mieux appris la fonction $\sin\left(\frac{1}{x}\right)$. Le bruit ε suit une loi normale d'écart type 0.3. La série ainsi générée est représentée figure 3.6.

Comme on peut le remarquer, elle est particulièrement peu régulière, on pourrait même croire qu'il y a 2 régimes différents alors qu'il n'y en a qu'un seul.

FIG. 3.6 – Série simulée



3.4.2 Apprentissage par des techniques de gradient

On procède de la même façon que pour l'apprentissage de fonction.

On donne ici la valeur moyenne de l'erreur quadratique, en théorie elle doit être le plus proche possible de 0.09. On peut voir que les résultats sont acceptables, mis à part le gradient stochastique qui surestime plus que les autres la variance du bruit, certainement parce que le nombre d'itérations (10^6) n'est pas suffisant.

TAB. 3.14 – Estimation de la série, suivant les différentes graines initiales, pour une initialisation aléatoire

Graine	0	1	2	3	4	5	6	7	8	9
Grad. conj.	0.096	0.096	0.099	0.096	0.096	0.096	0.096	0.096	0.134	0.134
BFGS	0.134	0.133	0.134	0.106	0.134	0.134	0.133	0.133	0.134	0.134
Lev. Mar.	0.102	0.127	0.135	0.132	0.099	0.097	0.101	0.122	0.130	0.097
Grad. stoch.	0.134	0.134	0.134	0.133	0.134	0.134	0.133	0.134	0.134	0.134

3.4.3 Estimation et initialisation par recuit simulé

3.4.3.1 Résultat du recuit simulé seul

On fait un recuit simulé avec 100 températures et des paliers de longueur 1000, où la règle de refroidissement est $T_k = T_0 \times 0.95^k$. On recommencera quatre fois le recuit en prenant $T_0 = 0.005$, $T_0 = 0.001$ puis $T_0 = 0.0005$ et $T_0 = 0.0001$. Les résultats sont résumés tableau 3.15.

TAB. 3.15 – Estimation par recuit simulé

T_0	0.005	0.001	0.0005	0.0001
Erreur finale	0.095	0.093	0.092	0.093

3.4.3.2 Poursuite de l'apprentissage grâce au gradient

On peut continuer l'apprentissage du meilleur MLP, optimisé par le recuit simulé pour essayer de diminuer encore plus l'erreur quadratique. Après 1000 itérations, l'erreur qui était de 0.09190 descend jusqu'à 0.09189. Le gradient n'améliore pratiquement plus l'erreur quadratique moyenne. Le recuit simulé a donc convergé tout près d'un minimum.

3.4.3.3 Initialisation par recuit simulé

L'initialisation par recuit simulé permet de trouver un meilleur point de départ à l'algorithme et limite le risque de converger vers un mauvais minimum local.

TAB. 3.16 – Estimation de la série, suivant les différentes graines initiales, pour une initialisation par recuit simulé

Graine	0	1	2	3	4	5	6	7	8	9
Grad. conj.	0.095	0.096	0.096	0.095	0.097	0.094	0.094	0.097	0.107	0.111
BFGS	0.095	0.096	0.097	0.095	0.099	0.095	0.099	0.097	0.108	0.111
Lev. Mar.	0.100	0.095	0.096	0.095	0.095	0.095	0.094	0.097	0.107	0.095
Grad. stoch.	0.104	0.100	0.098	0.096	0.102	0.099	0.100	0.098	0.109	0.115

3.4.4 Conclusion de l'estimation de la série

Le recuit simulé permet essentiellement d'éviter des résultats très loin du vrai minimum. Cependant, on voit que les algorithmes de gradient sont bien souvent proches

de la valeur désirée de l'erreur (si on prend la meilleure valeur de l'erreur pour 10 initialisations aléatoires). Les avantages du recuit sont donc moins clairs pour cette étude. Pourtant l'allure de la série est bien plus chaotique que nombre de séries réelles. C'est pourquoi un nombre raisonnable d'initialisations aléatoires peut être suffisant dans de nombreux cas.

3.5 Conclusion

On a vu que pour un problème difficile, le recuit simulé pouvait améliorer l'estimation. La technique donnant les meilleurs résultats, parmi celles testées, est un recuit avec une densité de proposition gaussienne de variance plutôt grande et une température initiale "faible".

L'algorithme du recuit est plus coûteux que celui du gradient. Ainsi pour l'expérience sur une série temporelle de longueur 1000, le temps de calcul sur un PC de bureau (266 Mhz) est de l'ordre de l'heure pour 100000 itérations. Une technique de gradient calcule beaucoup moins de fois la fonction de coût et le temps de calcul pour 10 initialisations aléatoires différentes est de 5 minutes.

A moins de vouloir à tout prix la valeur la plus basse possible pour la fonction d'erreur, un bon compromis semble être d'initialiser avec un recuit simulé rapide. Cela fournit une assez bonne initialisation de l'algorithme, mais n'est pas forcément moins coûteux en temps de calcul que d'augmenter le nombre d'initialisations aléatoires.

La principal difficulté avec l'initialisation par recuit simulé réside dans l'impossibilité de dire quand le recuit a trouvé un bon bassin d'attraction pour débuter la descente de gradient et cela en limite beaucoup l'intérêt. On peut donc dire que le recuit devra être appliqué essentiellement si on veut vraiment un très bon minimum et qu'alors il faudra en payer le prix en temps de calcul.

Chapitre 4

Estimation et identification de modèles autorégressifs non-linéaires multidimensionnels

Un modèle autorégressif fonctionnel (généralement non linéaire) correspond à l'idée de régression à chaque instant sur l'espace des observations passées. On suppose ici que la fonction de régression est une fonction paramétrique dérivable par rapport à ses paramètres (par exemple une fonction représentée par un perceptron multicouches). Pour une série scalaire, l'estimateur du maximum de vraisemblance d'un modèle dont l'innovation est gaussienne, coïncide avec l'estimateur des moindres carrés si on suppose que la variance de l'innovation est strictement positive. Néanmoins, cette remarque n'est, en général, plus vraie dans le cas où la série observée est vectorielle. En calculant l'estimateur du maximum de vraisemblance dans le cas gaussien, on s'aperçoit que la fonction à minimiser n'est pas la trace, mais le déterminant de la matrice de covariance empirique du bruit (cf Gallant [30]). On montrera d'abord comment on peut calculer facilement cet estimateur. Ensuite on montrera, en suivant la même démarche que Yao [64], que l'estimateur lié à la vraisemblance gaussienne fournit un estimateur consistant, qui vérifie les propriétés de normalité asymptotique et une loi du logarithme itéré, si on suppose que le bruit a un moment d'ordre suffisant. Cela fournira un critère d'identification presque sûre du modèle qui inclut le BIC.

4.1 Le modèle

Pour une série (Y_t), $t \in \mathbb{N}^*$ on notera Y_t^{t+l} le vecteur $(Y_t, \cdots, Y_{t+l})^T$ avec $l \in \mathbb{N}^*$.

On considère le modèle suivant :

$$Y_{t+1} = F_W(Y_{t-p+1}^t) + \varepsilon_{t+1} \tag{4.1}$$

où

- p est l'ordre de régression du modèle.
- (Y_t), $t \in \mathbb{Z}$, $t \geq -p+1$ est une suite de variables aléatoires de $\mathbb{R}^d$.
- F_W est une fonction paramétrique, continûment dérivable par rapport à ses paramètres, avec pour vecteur paramètre $W \in \mathbb{R}^D$,
- (ε_t), $t \in \mathbb{N}^*$, est une suite de variables aléatoires vectorielles indépendantes identiquement distribuées de matrice de covariance $\Gamma \in \mathbb{R}^{(d+1)d/2}$ inversible et inconnue.
- On supposera les observations initiales y^0_{-p+1} connues et fixées

Notation 5 *On note le vecteur paramètre $\theta = (W, \Gamma) \in \Theta = \Theta^W \times \Theta^\Gamma$, où $W \in \Theta^W \subset \mathbb{R}^D$, $\Gamma \in \Theta^\Gamma \subset \mathbb{R}^{\frac{d(d+1)}{2}}$ donc en posant $B = D + (d+1)\,d/2$, B est le nombre total de paramètres et $\theta \in \Theta \subset \mathbb{R}^B$. Pour simplifier les calculs on estime Γ^{-1} à la place de Γ, puisque cette matrice est supposée inversible, les deux paramétrisations sont équivalentes.*

Notation 6 *Dans toute la suite, si X est un vecteur multidimensionnel, $X(i)$ désignera sa i ème coordonnée.*

Notation 7 *Si X est une matrice inversible on notera x^{-1}_{ij} les coefficients de X^{-1}.*

Notation 8 *On note $\mathbb{F}_t$ la tribu engendrée par $Y_{-p+1}, \cdots, Y_t$, $t > 0$*

Remarque 5 *Toute cette étude est valable si on considère que Θ est inclus dans un espace polonais, c'est-à-dire un espace métrique complet et séparable, mais par souci de clarté on exposera les résultats pour des paramètres réels.*

Nous supposons, dans un premiers temps, que l'innovation ε est gaussienne, on peut alors calculer la log-vraisemblance du modèle. En effet, la vraisemblance s'écrit en fonction des observations $(y_1, \cdots, y_n)$ et du paramètre θ.

$$L_\theta(y_1, ..., y_n) = \frac{1}{(2\pi)^d \det(\Gamma)}^{\frac{n}{2}} \exp\left(-\frac{1}{2}\sum_{t=1}^{n}(y_t - F_W(y_{t-p}^{t-1}))^T\Gamma^{-1}(y_t - F_W(y_{t-p}^{t-1}))\right)$$

d'où la log-vraisemblance :

$$l_\theta(y_1, \cdots, y_n) := \ln(L_\theta(y_1, ..., y_n))$$

$$= -\frac{n}{2}\ln(\det(\Gamma)) - \frac{1}{2}\sum_{t=1}^{n}(y_t - F_W(y_{t-p}^{t-1}))^T\Gamma^{-1}(y_t - F_W(y_{t-p}^{t-1})) + Cte. \qquad (4.2)$$

4.2 Maximisation de la log-vraisemblance

On va d'abord montrer que pour maximiser cette fonction, on peut simplement trouver les paramètres $\hat{W}_n$ qui minimisent le déterminant de la covariance empirique du bruit, et poser $\hat{\Gamma}_n^{-1}$ égale à l'inverse de la covariance empirique calculée grâce aux paramètres $\hat{W}_n$.

4.2.1 Expression de $\hat{\Gamma}_n^{-1}$ en fonction de $\hat{W}_n$

On rappelle trois formules classiques que l'on utilisera par la suite :

- Si A de coefficient a_{ij}, est une matrice constante et X une matrice de coefficients x_{ij} :

$$\frac{\partial}{\partial x_{ij}} Tr(AX) = a_{ji}. \tag{4.3}$$

- En supposant maintenant X inversible on a :

$$\frac{\partial}{\partial x_{ij}} \ln(\det(X)) = x_{ji}^{-1}. \tag{4.4}$$

- Si A, B, C sont trois matrices de taille convenable, la trace de leur produit est invariante par permutation circulaire :

$$Tr(ABC) = Tr(BCA) = Tr(CAB). \tag{4.5}$$

On a le résultat suivant :

Proposition 1 *Notons $\Gamma_n(W)$ la covariance empirique :*

$$\Gamma_n(W) = \frac{1}{n}\sum_{t=1}^{n}(y_t - F_W(y_{t-p}^{t-1}))(y_t - F_W(y_{t-p}^{t-1}))^T$$

et

$$\Gamma_n^{-1}(W) = \left(\frac{1}{n}\sum_{t=1}^{n}(y_t - F_W(y_{t-p}^{t-1}))(y_t - F_W(y_{t-p}^{t-1}))^T\right)^{-1}.$$

Soit $\hat{\theta}_n = (\hat{W}_n, \hat{\Gamma}_n^{-1})$, l'estimateur du maximum de vraisemblance, on a :

$$\hat{W}_n = \arg\min_{W\in\Theta^W}\left(\frac{1}{2}\ln\det\left(\Gamma_n(W)\right)\right) \tag{4.6}$$

et

$$\hat{\Gamma}_n^{-1} = \left(\Gamma_n(\hat{W}_n)\right)^{-1}$$

en supposant que $\Gamma_n(\hat{W}_n)$ est bien inversible.

Preuve Fixons les paramètres W, la dérivée de la log-vraisemblance (multipliée par $\frac{2}{n}$) par rapport au coefficient Γ_{ij}^{-1} s'écrit :

$$\frac{\partial}{\partial \Gamma_{ij}^{-1}}(\ln(\det(\Gamma_n^{-1}(W))) - \frac{1}{n}\sum_{t=1}^{n}\frac{\partial}{\partial \Gamma_{ij}^{-1}} Tr\left((y_t - F_W(y_{t-p}^{t-1}))^T \Gamma^{-1}(y_t - F_W(y_{t-p}^{t-1}))\right)$$

grâce aux formules (7.7) et (7.8) cela s'écrit :

$$\Gamma_{ji} - \frac{1}{n}\sum_{t=1}^{n}\frac{\partial}{\partial \Gamma_{ij}^{-1}} Tr\left((y_t - F_W(y_{t-p}^{t-1}))(y_t - F_W(y_{t-p}^{t-1})^T \Gamma^{-1}\right).$$

En utilisant la formule (7.6) on obtient :

$$\frac{\partial \frac{2}{n} l_\theta(y_1, \cdots, y_n)}{\partial \Gamma_{ij}^{-1}} = \Gamma_{ji} - \frac{1}{n}\sum_{t=1}^{n}\left(y_t - F_W(y_{t-p}^{t-1})\right)(j) \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(i)$$

Les fonctions

$$\Gamma^{-1} \longmapsto (\ln(\det(\Gamma^{-1})) - \frac{1}{n}\sum_{t=1}^{n} Tr\left((y_t - F_W(y_{t-p}^{t-1}))^T \Gamma^{-1}(y_t - F_W(y_{t-p}^{t-1}))\right)$$

et

$$\Gamma^{-1} \longmapsto (\ln(\det(\Gamma^{-1}))$$

ont la même matrice hessienne (cf formules (7.6) et (7.8)). De plus la fonction $\Gamma^{-1} \longmapsto \ln\det(\Gamma^{-1})$ est une fonction strictement concave sur l'ensemble convexe des matrices symétriques définies positives (cf [36] théorème 7.6.7) donc

$$\Gamma^{-1} \longmapsto (\ln(\det(\Gamma^{-1})) - \frac{1}{n}\sum_{t=1}^{n} Tr\left((y_t - F_W(y_{t-p}^{t-1}))^T \Gamma^{-1}(y_t - F_W(y_{t-p}^{t-1}))\right)$$

aussi.

Elle atteint son maximum en l'unique point où sa dérivée s'annule. Ainsi, pour W fixé, le maximum de la log-vraisemblance en fonction de la matrice Γ^{-1} s'exprime en fonction de W :

$$\hat{\Gamma}_n^{-1}(W) = \left(\frac{1}{n}\sum_{t=1}^{n}(y_t - F_W(y_{t-p}^{t-1}))(y_t - F_W(y_{t-p}^{t-1}))^T\right)^{-1}.$$

Le vecteur paramètre $\hat{\theta}_n = (\hat{W}_n, \hat{\Gamma}_n)$ qui maximise la log-vraisemblance vérifie donc :

$$\hat{W}_n = \arg\min_{W\in\Theta^W}\left(\frac{n}{2}\ln(\det(\Gamma_n(W))) + \frac{1}{2}\sum_{t=1}^{n} Tr(y_t - F_W(y_{t-p}^{t-1})^T \Gamma_n^{-1}(W)(y_t - F_W(y_{t-p}^{t-1}))\right) \tag{4.7}$$

et

$$\hat{\Gamma}_n = \Gamma_n(\hat{W}_n).$$

Ce qui est équivalent, en remplaçant $\Gamma_n(W)$ par $\Gamma_n(\hat{W}_n)$ dans (4.7) à :

$$\hat{W}_n = \arg \min_{W \in \Theta^W} \left(\frac{1}{2} \ln \det \left(\Gamma_n(W) \right) \right)$$

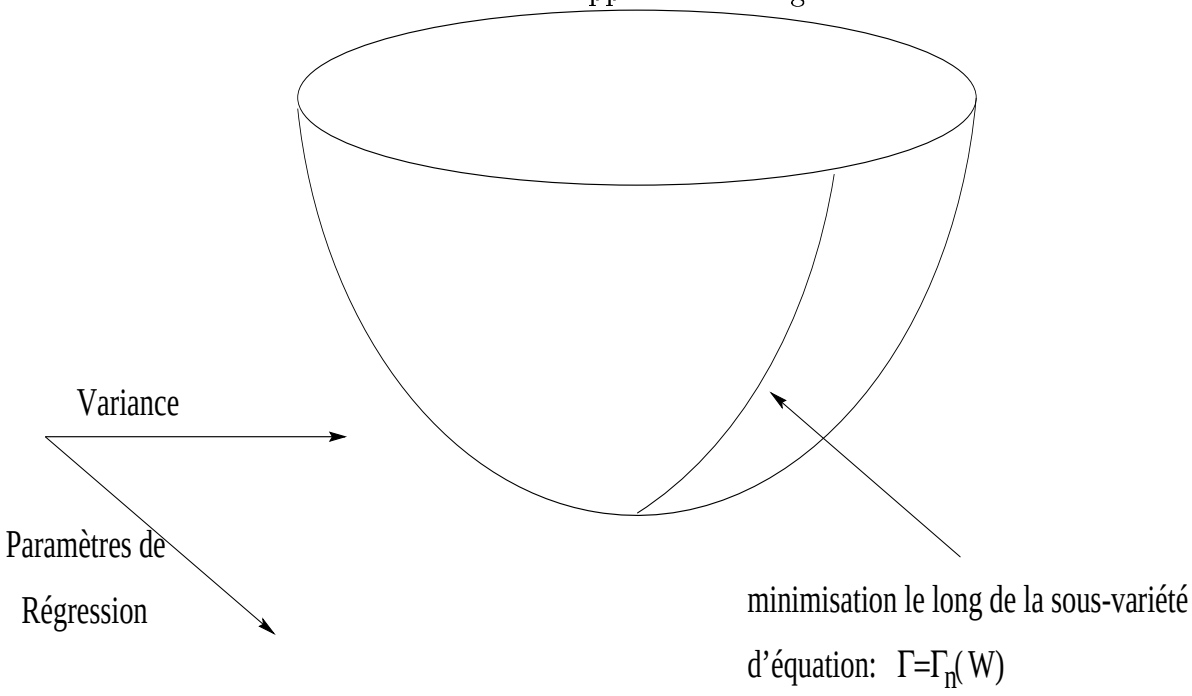
et

$$\hat{\Gamma}_n = \Gamma_n(\hat{W}_n)$$

■

En général, on n'a pas de forme explicite de $\hat{W}_n = \arg\min_{W \in \Theta^W} \left(\ln \det \left(\Gamma_n(W) \right) \right)$, néanmoins une solution acceptable est de savoir calculer la dérivée de cette fonction et d'approcher ce minimum par optimisation différentielle. Ainsi, pour maximiser la log-vraisemblance, il suffit d'optimiser cette fonction le long de la sous-variété de $(W, \Gamma) := \mathbb{R}^{(D+(d+1)*d/2)}$ définie par $\Gamma = \Gamma_n(W)$.

FIG. 4.1 – Minimisation de l'opposée de la log-vraisemblance



4.2.2 Dérivée de $\ln \det \left(\Gamma_n(W) \right)$

On suppose ici que $\forall y_1^p \in \left(\mathbb{R}^d\right)^p$, la fonction $W \longmapsto F_W\left(y_1^p\right)$ est continûment dérivable.

Notation 9 *Dans la suite, si X est une matrice symétrique, la notation :*

$$(X_{ij})_{ind} := (X_{ij})_{1\leq i,j\leq d}$$

indiquera le vecteur :

$$(X_{11}, X_{12}, \cdots, X_{1d}, X_{21}, X_{22}, \cdots X_{2d}, \cdots, X_{dd})^T.$$

La fonction $\ln(\det(\Gamma_n(W)))$ s'exprime comme la fonction composée $f(g(W))$ avec :

- $g : \mathbb{R}^D \longrightarrow \mathbb{R}^{d(d+1)/2}$ telle que :

$$\Gamma_{ij} = \Gamma_{ji} = g_{ij}(W) = g_{ji}(W) := \frac{1}{n}\sum_{t=1}^{n}\left(y_t - F_W(y_{t-p}^{t-1})\right)(i) \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j).$$

- $f : \mathbb{R}^{d(d+1)/2} \longrightarrow \mathbb{R}$ telle que :

$$f(\Gamma) = \ln(\det(\Gamma)).$$

Grâce à la formule de dérivée de fonction composée (Cartan [14] théorème 2.2.1), on aura pour tout $k \in \{1, ..., D\}$[1]

$$\frac{\partial}{\partial W_k}(\ln(\det(\Gamma_n(W)))) = \left(\frac{\partial}{\partial \Gamma_{ij}}(\ln(\det(\Gamma_n(W)))\right)^T_{ind}\left(\frac{\partial \Gamma_{ij}}{\partial W_k}\right)_{ind}$$

avec

$$\frac{\partial \Gamma_{ij}}{\partial W_k} = \frac{1}{n}\sum_{t=1}^{n}\frac{\partial}{\partial W_k}\left[\left(y_t - F_W(y_{t-p}^{t-1})\right)(i) \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j)\right]$$

soit

$$\frac{\partial \Gamma_{ij}}{\partial W_k} = \frac{1}{n}\sum_{t=1}^{n}\left[-\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j) - \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k}\left(y_t - F_W(y_{t-p}^{t-1})\right)(i)\right]$$

et grâce à la formule (7.7) :

$$\frac{\partial}{\partial \Gamma_{ij}}\ln(\det(\Gamma_n(W)) = \Gamma_{ij}^{-1} = \Gamma_{ji}^{-1},$$

car la matrice $\Gamma_n^{-1}(W)$ est symétrique.

[1] La notation $\frac{\partial f(X)}{\partial X_k}$ signifie la k-ème coordonnée de la dérivée de $f(X)$ au point X

On en déduit la dérivée de la log-vraisemblance par rapport à l'élément W_k du vecteur paramètre W :

$$\frac{\partial}{\partial W_k}(\ln(\det(\Gamma_n(W)))) = \left(\Gamma_{ij}^{-1}\right)_{ind}^T \left(\frac{\Gamma_{ij}}{\partial W_k}\right)_{ind}$$

et

$$\frac{\partial \Gamma_{ij}}{\partial W_k} = \frac{1}{n}\sum_{t=1}^{n}\left[-\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j) - \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k}\left(y_t - F_W(y_{t-p}^{t-1})\right)(i)\right]$$

d'où la formule de la dérivée :

$$\frac{\partial}{\partial W_k}(\ln(\det(\Gamma_n(W)))) =$$

$$\left(\Gamma_{ij}^{-1}\right)_{ind}^T \left(\frac{1}{n}\sum_{t=1}^{n} -\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j) - \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k}\left(y_t - F_W(y_{t-p}^{t-1})\right)(i)\right)_{ind}. \tag{4.8}$$

Connaissant la dérivée de $\ln\det(\Gamma_n(W))$, il est maintenant facile de la minimiser par une technique d'optimisation différentielle.

4.3 Propriétés statistiques de l'estimateur

On appellera le contraste $U_n(\theta) = -\frac{1}{n}l_\theta(y_1, \cdots, y_n)$ (cf 4.3.2.1) : contraste associé à la vraisemblance. Nous allons montrer que l'estimateur du minimum de contraste converge presque sûrement vers le bon paramètre (consistance) et qu'il converge suffisamment vite (théorème de la limite centrale, loi du logarithme itéré) pour pouvoir aussi identifier le modèle par un contraste pénalisé. Cela, même dans le cas où l'innovation n'est plus gaussienne, mais garde un moment fini d'ordre suffisamment grand.

4.3.1 Hypothèses de base

La chaîne vectorisée $\left(Y_{t-p+1}^t\right)_{t>0}$vérifie l'équation :

$$Y_{t-p+1}^t = \begin{pmatrix} Y_t \\ Y_{t-1} \\ \vdots \\ Y_{t-p+1} \end{pmatrix} = \begin{pmatrix} F_W(Y_{t-1}, \cdots, Y_{t-p}) \\ Y_{t-1} \\ \vdots \\ Yt-p+1 \end{pmatrix} + \begin{pmatrix} \varepsilon_t \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4.9}$$

La loi forte des grands nombres est assurée par le théorème suivant : (cf Duflo [24])

Théorème 6 *Soit le modèle vérifiant (4.9). supposons que le bruit* (ε_t) *a une densité positive par rapport à la mesure de Lebesgue sur* $\mathbb{R}^d$ *avec un moment d'ordre* $a \geq 1$ *. Si il existe des nombres positifs* $\nu_1, \cdots, \nu_p$ *tels que* $\nu_1 + \cdots + \nu_p < 1$*, une constante* $\kappa \geq 0$ *et une norme* $\|.\|$ *de* $\mathbb{R}^d$ *satisfaisant pour tout* $y \in \left(\mathbb{R}^d\right)^p$ *:*

$$\|F_{W_0}(y_1^p)\| \leq \nu_1 \|y_1\| + \cdots + \nu_p \|y_p\| + \kappa,$$

alors la chaîne vectorisée $\left(Y_{t-p+1}^t\right)_{t>0}$ *est stable, géométriquement ergodique et sa mesure invariante* μ_0 *a une densité par rapport à la mesure de Lebesgue qui admet un moment d'ordre* a*.*

Remarque 6 *Soit* $(Y_t)_{t\in\mathbb{N}^*}$ *un processus ergodique à valeurs dans* $\mathbb{R}^d$*, alors* $\forall q > 0$*,* $\forall g$ *fonction intégrable, le processus* $\left(g(Y_{t-q}^t)\right)_{t\in\mathbb{N}^*}$ *est un processus ergodique.*

Dans la suite, on supposera que le modèle vérifie les hypothèses (**H**) :

1. Le processus vérifie les hypothèses du théorème 6, avec un moment d'ordre $a \geq 2$ et pour tout $W \in \Theta^W$ $F_W\left(Y_1^p\right)$ a un moment d'ordre a.
2. Θ est un compact de $\mathbb{R}^B$ et le vrai paramètre $\theta_0 = \left(W_0, \Gamma_0^{-1}\right)$ appartient à l'intérieur de Θ .
3. Pour tout $Y_1^p \in \left(\mathbb{R}^d\right)^p$, $F_W(Y_1^p)$ est continue sur Θ^W par rapport au paramètre W.
4. Toute matrice de covariance $\Gamma \in \Theta^\Gamma$ est définie positive. On note, dans la suite λ_{max} (resp. λ_{min}) la plus grande valeur propre de Γ pour $\Gamma \in \Theta^\Gamma$(resp. la plus petite valeur propre de Γ pour $\Gamma \in \Theta^\Gamma$). On a $0 < \lambda_{min}, \lambda_{max} < \infty$ car, pour tout $\Gamma \in \Theta^\Gamma$, Γ est définie positive et Θ^Γ est compact.
5. Le modèle est supposé identifiable, c'est-à-dire : $F_{W'} = F_W \Leftrightarrow W' = W$.

4.3.2 Consistance de l'estimateur.

4.3.2.1 Vérification des propriétés de contraste :

La fonction $U_n(\theta)$ associée à la vraisemblance est définie par :

$$U_n(\theta) = -\frac{2}{n} l_\theta(y_1, \cdots, y_n)$$

$U_n(\theta)$ est un processus contraste relatif à une fonction $\theta \longmapsto K\left(\theta_0, \theta\right)$ si $U_n\left(\theta\right)$ est $\mathcal{F}_n$-adapté et si :

$$\lim_{n\to\infty} U_n\left(\theta\right) - U_n\left(\theta_0\right) \xrightarrow{p.s.} K\left(\theta_0, \theta\right) \geq 0$$

avec

$$K\left(\theta_0, \theta\right) = 0 \Leftrightarrow \theta = \theta_0$$

On a, en notant $\varepsilon_t^W = Y_t - F_W(Y_{t-p}^{t-1})$:

$$U_n(\theta) - U_n(\theta_0) = \frac{1}{n}(l_{\theta_0}(y_1, \cdots, y_n) - l_\theta(y_1, \cdots, y_n))$$

$$= \ln((\frac{\det \Gamma}{\det \Gamma_0})) + \frac{1}{n}\left(\sum_{t=1}^{n} \left(\varepsilon_t^W\right)^T \Gamma^{-1} \left(\varepsilon_t^W\right) - \sum_{t=1}^{n} \left(\varepsilon_t^{W_0}\right)^T \Gamma_0^{-1} \left(\varepsilon_t^{W_0}\right)\right).$$

Mais

$$\begin{aligned}\varepsilon_t^W = Y_t - F_W\left(Y_{t-p}^{t-1}\right) &= Y_t - F_{W_0}\left(Y_{t-p}^{t-1}\right) + F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right) \\ &= \varepsilon_t^{W_0} + F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\end{aligned}$$

donc

$$\left(\varepsilon_t^W\right)^T \Gamma^{-1}\varepsilon_t^W - \left(\varepsilon_t^{W_0}\right)^T \Gamma_0^{-1}\varepsilon_t^{W_0} = tr\left(\Gamma^{-1}\varepsilon_t^W \left(\varepsilon_t^W\right)^T\right) - tr\left(\Gamma_0^{-1}\varepsilon_t^{W_0}\left(\varepsilon_t^{W_0}\right)^T\right)$$

$$= tr\left(\left(\Gamma^{-1} - \Gamma_0^{-1}\right)\varepsilon_t^{W_0}\left(\varepsilon_t^{W_0}\right)^T\right) + 2tr\left(\Gamma^{-1}\varepsilon_t^{W_0}\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)^T\right)$$

$$+tr\left(\Gamma^{-1}\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)^T\right).$$

La condition **H**-1 assure que pour tout θ et toute norme $\|.\|$, $\left\|\widetilde{\varepsilon}_t^W\right\|^2$est intégrable par rapport à la mesure invariante μ_0. Par la loi forte des grands nombres on aura p.s. :

$$\lim_{n\to\infty}\frac{1}{n}\sum_{t=1}^{n}\left(\varepsilon_t^W\right)^T\Gamma^{-1}\varepsilon_t^W - \left(\varepsilon_t^{W_0}\right)^T\Gamma_0^{-1}\varepsilon_t^{W_0} \overset{p.s.}{=} tr\left(\left(\Gamma^{-1} - \Gamma_0^{-1}\right)\Gamma_0\right) + 0$$

$$+tr\left(\Gamma^{-1}E_{\theta_0}\left[\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)^T\right]\right)$$

car $\varepsilon_t^{W_0}$ et $\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)$ sont indépendants et $\varepsilon_t^{W_0}$ est de moyenne nulle.

Ainsi

$$\lim_{n\longrightarrow\infty} U_n(\theta) - U_n(\theta_0) := K(\theta_0, \theta)$$

avec

$$K(\theta_0,\theta) = \frac{1}{2}\left(\ln\frac{\det\Gamma}{\det\Gamma_0} + tr\left(\Gamma^{-1}\Gamma_0 - Id\right)\right)$$

$$+\frac{1}{2}E_{\theta_0}\left[\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)\Gamma^{-1}\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)^T\right].$$

- Le premier terme : $\frac{1}{2}\left(\ln\frac{\det\Gamma}{\det\Gamma_0} + tr\left(\Gamma^{-1}\Gamma_0 - Id\right)\right)$ est égale à la distance de Kullback $K\left(\mathcal{N}\left(0,\Gamma\right),\mathcal{N}\left(0,\Gamma_0\right)\right) \geq 0$ et valant 0 si et seulement si $\Gamma = \Gamma_0$.
- Comme toute matrice $\Gamma \in \Theta^\Gamma$ est définie positive, le deuxième terme :

$$\frac{1}{2}E_{\theta_0}\left[\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)\Gamma^{-1}\left(F_{W_0}\left(Y_{t-p}^{t-1}\right) - F_W\left(Y_{t-p}^{t-1}\right)\right)^T\right]$$

est toujours positif et ne vaut 0 que si $F_W \overset{p.s.}{=} F_{W_0}$.

L'hypothèse d'identifiabilité **H**-5 assure donc que $K(\theta_0, \theta) = 0$ seulement pour $\theta = \theta_0$.

4.3.2.2 Consistance forte

Dans le cadre des hypothèses (**H**), une condition suffisante assurant la consistance forte de $(\hat{\theta}_n)$ est (cf Guyon [31] section 3.4) :

Lemme 1 *Pour $\eta > 0$, posons*

$$\omega_n(\eta) = \sup \left\{ \left| U_n(\theta_\alpha) - U_n(\theta_\beta) \right| ; \left\| \theta_\alpha - \theta_\beta \right\| \leq \eta \right\}.$$

Si $\theta \longmapsto K(\theta_0, \theta)$ est continue et si il existe une suite (ϵ_k) réelle et décroissante vers 0 telle que, pour tout entier $k > 0$,

$$\lim_{n \longrightarrow \infty} P_{\theta_0} \left[\limsup_{n \to \infty} \left(\omega_n(\frac{1}{k}) > \epsilon_k \right) \right] = 0,$$

alors l'estimateur du minimum de contraste est fortement consistant.

On a :

$$l_\theta(y_1^{p+1}) = \ln \det \Gamma + \left(y_{p+1} - F_W(y_1^p) \right)^T \Gamma^{-1} \left(y_{p+1} - F_W(y_1^p) \right).$$

Posons $\Theta^* = \Theta \cap \mathbb{Q}^B$, où $\mathbb{Q}$ est l'ensemble des nombres rationnels. Cet ensemble est dense dans Θ et dénombrable. Soit la famille de fonctions (g_η) , $\eta > 0$.

$$\begin{array}{l} g_\eta(y_1^{p+1}) := \\ \sup \left\{ \left| l_{\theta_\alpha}(y_1^{p+1}) - l_{\theta_\beta}(y_1^{p+1}) \right| ; \left\| (\theta_\alpha) - (\theta_\beta) \right\| \leq \eta,\ (\theta_\alpha, \theta_\beta) \in \Theta^* \times \Theta^* \right\}. \end{array}$$

La fonction g_η est une variable aléatoire.

Pour tout $y_1^{p+1} \in \left(\mathbb{R}^d \right)^{p+1}$, par continuité de la fonction

$$(\theta_\alpha, \theta_\beta) \longmapsto \left| l_{\theta_\alpha}(y_1^{p+1}) - l_{\theta_\beta}(y_1^{p+1}) \right|$$

et la densité de Θ^*dans Θ, on aura :

$$\begin{array}{l} \sup \left\{ \left| l_{\theta_\alpha}(y_1^{p+1}) - l_{\theta_\beta}(y_1^{p+1}) \right| ; \left\| (\theta_\alpha) - (\theta_\beta) \right\| \leq \eta,\ (\theta_\alpha, \theta_\beta) \in \Theta^* \times \Theta^* \right\} \\ = \sup \left\{ \left| l_{\theta_\alpha}(y_1^{p+1}) - l_{\theta_\beta}(y_1^{p+1}) \right| ; \left\| (\theta_\alpha) - (\theta_\beta) \right\| \leq \eta,\ (\theta_\alpha, \theta_\beta) \in \Theta \times \Theta \right\}. \end{array}$$

Sous les hypothèses (**H**), on a la majoration :

$$\sup_{\theta \in \Theta} \left| l_\theta(y_1^{p+1}) \right| \leq h(y_1^{p+1}) \tag{4.10}$$

avec

$$h(y_1^{p+1}) := d \times \sup \left(\left| \ln(\lambda_{max}) \right|, \left| \ln(\lambda_{min}) \right| \right) + \frac{1}{\lambda_{min}} \left(\left\| y_{p+1} \right\|^a + \xi_1 \left\| y_1 \right\|^a + \cdots + \xi_p \left\| y_p \right\|^a + \kappa \right),$$

$\xi_1, \cdots, \xi_p$ et κ étant des constantes positives finies dont l'existence est assurée grâce à la condition **H**-1, alors $g_\eta \leq 2h$ avec h intégrable par rapport à la mesure invariante.

La continuité uniforme (car Θ est compact) de

$$(\theta_\alpha, \theta_\beta) \longmapsto \left| l_{\theta_\alpha}(y_1^{p+1}) - l_{\theta_\beta}(y_1^{p+1}) \right|$$

implique, par convergence dominée, que

$$\theta \longmapsto K(\theta_0, \theta)$$

est continue.

De même on aura

$$\forall (y_1, \cdots, y_{p+1}) \in \left(\mathbb{R}^d\right)^{p+1}, \lim_{\eta \longrightarrow 0} g_\eta(y_1, \cdots, y_{p+1}) = 0$$

donc par convergence dominée :

$$\lim_{\eta \longrightarrow 0} E_{\theta_0}\left[g_\eta(Y_1, \cdots, Y_{p+1})\right] = 0.$$

Finalement on aura P_{θ_0} p.s. :

$$\omega_n(\eta) \leq \frac{1}{n} \sum_{t=1}^{n} g_\eta(Y_t, \cdots, Y_{t-p}).$$

Il suffit donc de choisir pour $k \in \mathbb{N}^*$: $\epsilon_k = 2 \times E_{\theta_0}\left[g_{\frac{1}{k}}(Y_1, \cdots, Y_{p+1})\right]$ (ϵ_k décroît bien vers 0) pour que (en notant $i.s.$ pour infiniment souvent)

$$\limsup_{n \to \infty} \left\{ \omega_n\left(\frac{1}{k}\right) \geq \epsilon_k \right\} = \left\{ \omega_n\left(\frac{1}{k}\right) \geq \epsilon_k \, i.s. \right\}$$

$$\subseteq \frac{1}{n} \sum_{t=1}^{n} g_{\frac{1}{k}}(Y_t, \cdots, Y_{t-p}) \geq \epsilon_k \, i.s.$$

sur $A := \left\{ \frac{1}{n} \sum_{t=1}^{n} g_{\frac{1}{k}}(Y_t, \cdots, Y_{t-p}) \geq 2 \times E_{\theta_0}\left[g_{\frac{1}{k}}(Y_1, \cdots, Y_{p+1})\right] \; i.s. \right\}$, $\frac{1}{n} \sum_{t=1}^{n} g_{\frac{1}{k}}(Y_t, \cdots, Y_{t-p})$, ne peut converger vers $E_{\theta_0}\left[g_{\frac{1}{k}}(Y_1, \cdots, Y_{p+1})\right]$, A est donc un ensemble de mesure nulle. Cela montre le théorème suivant :

Théorème 7 *Dans le cadre des hypothèses (**H**), l'estimateur du minimum de contraste $U_n(\theta)$ est fortement consistant.*

4.3.3 Normalité asymptotique

Le Théorème de la Limite Centrale pour $\hat{\theta}_n$ nécessite des hypothèses supplémentaires sur les dérivées de $\theta \longmapsto U_n(\theta)$. Il faut s'assurer que les dérivées secondes, les carrés des dérivées premières sont bien intégrables, et avoir un contrôle sur la croissance des dérivées secondes.

4.3.3.1 Les dérivées d'ordre 1 et 2

On a

$$\frac{\partial U_n(\theta)}{\partial W_k} =$$

$$(\Gamma_{ij}^{-1})_{ind}^T \left(\frac{1}{n} \sum_{t=1}^{n} \frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1}) \right)(j) + \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k} \left(y_t - F_W(y_{t-p}^{t-1}) \right)(i) \right)_{ind}$$

et

$$\frac{\partial U_n(\theta)}{\partial \Gamma_{ij}^{-1}} = (\Gamma ij) - \left(\frac{1}{n} \sum_{t=1}^{n} \left(y_t - F_W(y_{t-p}^{t-1}) \right)(i) \left(y_t - F_W(y_{t-p}^{t-1}) \right)(j) \right).$$

On en déduit les dérivées d'ordre 2 :

$$\frac{\partial^2 U_n(\theta)}{\partial W_k \partial W_l} =$$

$$(\Gamma_{ij}^{-1})_{ind}^T \left(\frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 F_W(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l} \times \left(y_t - F_W(y_{t-p}^{t-1}) \right)(j) - \frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_l} \right)_{ind}$$

$$+(\Gamma_{ij}^{-1})_{ind}^T \left(\frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 F_W(y_{t-p}^{t-1})(j)}{\partial W_k \partial W_l} \times \left(y_t - F_W(y_{t-p}^{t-1}) \right)(i) - \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k} \frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_l} \right)_{ind}$$

ainsi que

$$\frac{\partial^2 U_n(\theta)}{\partial \Gamma_{ij}^{-1} \partial \Gamma_{kl}^{-1}} = \frac{\partial (\Gamma ij)}{\partial \Gamma_{kl}^{-1}}$$

et

$$\frac{\partial^2 U_n(\theta)}{\partial W_k \partial \Gamma_{ij}^{-1}} = \frac{1}{n} \sum_{t=1}^{n} \frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1}) \right)(j) + \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k} \left(y_t - F_W(y_{t-p}^{t-1}) \right)(i).$$

La dérivée de $\theta \longmapsto U_n(\theta)$ (resp. $W \longmapsto F_W$) sera noté $\nabla U_n(\theta)$ (resp. ∇F_W). De même la dérivée seconde de $\theta \longmapsto U_n(\theta)$ (resp. $W \longmapsto F_W$) sera noté $HU_n(\theta)$ (resp. HF_W).

4.3.3.2 Hypothèses supplémentaires

On suppose qu'il existe un voisinage V de θ_0 tel que les hypothèses (**N**) suivantes soient vérifiées.

1. Le bruit a un moment d'ordre $2a$ avec $a \geq 2$ et par le théorème 6, $(Y_t)_{t\in\mathbb{N}^*}$ a alors un moment d'ordre $2a$.
2. Pour tout $W \in \Theta^W \cap V$, les dérivées d'ordres 1, 2 et 3 de $W \longmapsto F_W$ sont μ_0-p.s. continues par rapport à y_1^p .
3. Pour tout $y_1^p \in \left(\mathbb{R}^d\right)^p$, $\forall k, j \,:\, 1 \leq k, j \leq D$

$$\left\| \frac{\partial F_{W_0}(y_1^p)}{\partial W_k} \right\| \leq Cte \times \left(1 + \|y_1^p\|^{a/2}\right)$$

et

$$\left\| \frac{\partial^2 F_{W_0}(y_1^p)}{\partial W_k \partial W_j} \right\| \leq Cte \times \left(1 + \|y_1^p\|^{a/2}\right).$$

4. Pour tout $y_1^p \in \left(\mathbb{R}^d\right)^p$, pour tout $W \in \Theta^W \cap V$,$\forall k, j, l \,:\, 1 \leq k, j, l \leq D$

$$\left\| \frac{\partial^3 F_W(y_1^p)}{\partial W_k \partial W_j \partial W_l} \right\| \leq Cte \times \left(1 + \|y_1^p\|^{a/2}\right).$$

Remarque 7 *La condition **N**-4 implique qu'il existe un module de continuité ζ tel que $\forall W \in \Theta^W \cap V$:*

$$\|HF_W(y_1^p) - HF_{W_0}(y_1^p)\| \leq \zeta(\|W - W_0\|)\left(1 + \|y_1^p\|^{a/2}\right).$$

Remarque 8 *Notons que la compacité de Θ, la remarque 7 et la condition **N**-3 impliquent qu'il existe une constante $\gamma > 0$ telle que $\forall W \in \Theta^W \cap V$:*

$$\|HF_W(y_1^p)\| \leq \gamma(1 + \|y_1^p\|^{a/2}).$$

Remarque 9 *La remarque précédente nous permet de déduire un contrôle sur les accroissements de la dérivée première $\forall W \in \Theta^W \cap V$:*

$$\forall k, i \left\| \frac{\partial F_W(y_1^p)}{\partial W_k} - \frac{\partial F_{W_0}(y_1^p)}{\partial W_k} \right\| \leq \gamma \|W - W_0\| \times \left(1 + \|y_1^p\|^{a/2}\right).$$

Donc on aura aussi l'existence d'une constante finie γ telle que $\forall W \in \Theta^W \cap V$:

$$\forall k \,:\, \left\| \frac{\partial F_W(y_1^p)}{\partial W_k} \right\| \leq Cte \times \left(1 + \|y_1^p\|^{a/2}\right).$$

Remarque 10 *De la même façon, le contrôle sur cette dérivée nous donne un contrôle sur les accroissements de la fonction de régression elle-même $\forall W \in \Theta^W \cap V$:*

$$\forall k, i, \, \|F_W(y_1^p) - F_{W_0}(y_1^p)\| \leq \gamma \|W - W_0\| \times \left(1 + \|y_1^p\|^{a/2}\right).$$

On aura alors

Proposition 2 *Sous les hypothèses (**H**) et (**N**), on a pour toute loi initiale de la chaîne $\left(Y_{t-p+1}^t\right)_{t\in\mathbb{N}}$:*

$$HU_n(\theta_0) \xrightarrow{p.s.} I_0 \tag{4.11}$$

où I_0 est une matrice symétrique.

Preuve Il suffit de montrer que chaque terme de la Hessienne $(HU_n(\theta_0))_{ij}$, $1 \leq i \leq j \leq B$ converge presque sûrement vers $(I_0)_{ij}$ et pour cela montrer qu'ils sont tous dominés par une fonction intégrable :

Terme de la forme $\frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial W_l}$: On a

$$\left\| \frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial W_l} \right\| \leq$$

$$\left\| (\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(j) - \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_l} \right)_{ind} \right\|$$

$$+ \left\| (\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k \partial W_l} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(i) - \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_l} \right)_{ind} \right\|$$

qui est majoré grâce à la condition **H**-4 par :

$$\frac{2}{\lambda_{min}} \sum_{1 \leq i,j \leq d} \left\| \frac{1}{n} \sum_{t=1}^{n} \frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(j) - \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_l} \right\|$$

ce qui est majoré par

$$\frac{2}{\lambda_{min}} \sum_{1 \leq i,j \leq d} \frac{1}{n} \sum_{t=1}^{n} \left\| \frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(j) \right\| + \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(i) \right\| \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \right\|$$

Mais grâce à la remarque 8 et à la condition (**H**)-1, sachant que $a \geq 2$, il existe une constante $\gamma > 0$ finie telle que pour tout $t \in \mathbb{N}^*$:

$$\left\| \frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1}) \right)(j) \right\| < \gamma(1 + \left\| y_{t-p}^t \right\|^a).$$

Donc par la loi forte des grands nombres, $\frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial W_l}$ converge presque sûrement vers le nombre fini :

$$E_{\theta_0} \left[(\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{\partial^2 F_W(Y_1^p)(i)}{\partial W_k \partial W_l} \times \left(Y_{p+1} - F_W(Y_1^P) \right)(j) - \frac{\partial^2 F_W(Y_1^p)(j)}{\partial W_k \partial W_l} \times \left(Y_{p+1} - F_W(Y_1^P) \right)(i) \right)_{ind} \right.$$

$$\left. + (\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{\partial^2 F_W(Y_1^p)(j)}{\partial W_k \partial W_l} \times \left(Y_{p+1} - F_W(Y_1^P) \right)(i) - \frac{\partial^2 F_W(Y_1^p)(i)}{\partial W_k \partial W_l} \times \left(Y_{p+1} - F_W(Y_1^P) \right)(j) \right)_{ind} \right].$$

Terme de la forme $\frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial \Gamma_{ij}^{-1}}$: On a

$$\left\| \frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial \Gamma_{ij}^{-1}} \right\| =$$

$$\left\| \frac{1}{n} \sum_{t=1}^{n} \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j) + \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i) \right\|,$$

ce qui est majoré par

$$\frac{1}{n} \sum_{t=1}^{n} \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j) \right\| + \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i) \right\|$$

et grâce aux conditions (**H**)-1 et (**N**)-3, il existe une constante $\gamma > 0$ finie telle que pour tout $t \in \mathbb{N}^*$:

$$\left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j) \right\| + \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i) \right\|$$

$$\leq \gamma(1 + \left\| y_{t-p}^{t} \right\|^a).$$

On peut encore appliquer la loi des grands nombres et $\frac{\partial^2 U_n(\theta_0)}{\partial W_k \partial \Gamma_{ij}^{-1}}$ converge presque sûrement vers :

$$E_{\theta_0} \left[\frac{\partial F_{W_0}(Y_1^p)(i)}{\partial W_k} \times \left(Y_{p+1} - F_{W_0}(Y_1^P)\right)(j) - \frac{\partial F_{W_0}(Y_1^p)(j)}{\partial W_k} \times \left(Y_{p+1} - F_W(Y_1^P)\right)(i) \right] < \infty.$$

Terme de la forme $\frac{\partial^2 U_n(W,\Gamma^{-1})}{\partial \Gamma_{ij}^{-1} \partial \Gamma_{kl}^{-1}} = \frac{\partial(\Gamma ij)}{\partial \Gamma_{kl}^{-1}}$: Ce terme est constant en y, donc il est intégrable. ■

Démontrons maintenant la proposition qui établit la normalité asymptotique du processus $\nabla U_n(\theta_0)$:

Proposition 3 *Sous les hypothèses (**H**) et (**N**), on a pour toute loi initiale de la chaîne* $\left(Y_{t-p+1}^t\right)_{t \in \mathbb{N}}$

$$\sqrt{n} \nabla U_n(\theta_0) \xrightarrow{Loi} N(0, J_0) \tag{4.12}$$

où J_0 est une matrice symétrique.

Pour montrer cette proposition, posons

$$M_n = -n \nabla U_n(\theta_0) \tag{4.13}$$

M_n est une martingale, montrons qu'elle est de carré intégrable. Pour cela on va montrer que chaque terme de son crochet est intégrable.

Notation 10 *Notons* $\widetilde{U}_\theta(y_{t-p}^t)$ *la fonction :*

$$\widetilde{U}_\theta(y_{t-p}^t) = \ln\det\Gamma + \left(y_t - F_W(y_{t-p}^{t-1})\right)^T \Gamma^{-1} \left(y_t - F_W(y_{t-p}^{t-1})\right)$$

et $\nabla\widetilde{U}_\theta(y_{t-p}^t)$ *la dérivée de* $\widetilde{U}_\theta(y_{t-p}^t)$ *par rapport aux paramètres.*

on a : $M_t - M_{t-1} = \nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)$,

Lemme 2 *Il existe une constante strictement positive* γ *telle que pour tout* $t \in \mathbb{N}^*$

$$\left\|\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)^T \nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)\right\| < \gamma\left(1 + \left\|y_{t-p}^t\right\|^{2a}\right) \tag{4.14}$$

Preuve On va examiner chaque terme.

Terme de la forme $\frac{\partial\widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_k} \times \frac{\partial\widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_l}$ **:** Il vaut

$$\begin{array}{l}(\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j) + \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k}\left(y_t - F_W(y_{t-p}^{t-1})\right)(i)\right)_{ind} \times \\ (\Gamma_{0_{ij}}^{-1})_{ind}^T \left(\frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_l} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j) + \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_l}\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\right)_{ind}\end{array}$$

La norme de ce terme est majorée par :

$$\left(\frac{1}{\lambda_{min}} \sum_{1\le i,j\le d} \left\|\frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l}\right\| \left\|\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j)\right\| + \left\|\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\right\| \left\|\frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k}\right\|\right)^2$$

Mais par la remarque 8 :

$$\left\|\frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l}\right\| \left\|\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j)\right\| < \gamma(1 + \left\|y_{t-p}^t\right\|^a)$$

donc on aura

$$\begin{array}{c}\left(\frac{1}{\lambda_{min}} \sum_{1\le i,j\le d} \left\|\frac{\partial^2 F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k \partial W_l}\right\| \left\|\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j)\right\| + \left\|\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\right\| \left\|\frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k}\right\|\right)^2 \\ \le \left[\frac{2}{\lambda_{min}} \sum_{1\le i,j\le d} \gamma(1 + \left\|y_{t-p}^t\right\|^a)\right]^2\end{array}$$

ce qui assure qu'il existe une constante γ_1 positive finie telle que :

$$\left\|\frac{\partial\widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_k} \times \frac{\partial\widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_l}\right\| \le \gamma_1\left(1 + \left\|y_{t-p}^t\right\|^{2a}\right).$$

Terme de la forme $\frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_k} \times \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{ij}^{-1}}$ **:** Il vaut

$$\left(\Gamma_{0_{ij}}^{-1}\right)_{ind}^{T}\left(\frac{\partial F_{W_0}(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j) + \frac{\partial F_{W_0}(y_{t-p}^{t-1})(j)}{\partial W_k}\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\right)_{ind} \times \left((\Gamma_{0_{ij}}) - \left(\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j)\right)\right).$$

Le module de ce terme sera alors majoré par

$$\frac{1}{\lambda_{min}} \sum_{1\leq i,j\leq d} \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right) \right\|^3 + \frac{2\lambda_{max}}{\lambda_{min}} \sum_{1\leq i,j\leq d} \left\| \frac{\partial F_{W_0}(y_{t-p}^{t-1})}{\partial W_k} \right\| \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right) \right\|$$

et grâce aux conditions (**N**)-3, (**N**)-1 et (**H**)-1, on en déduit l'existence d'un constante $\gamma_2 > 0$ telle que

$$\left\| \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial W_k} \times \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{ij}^{-1}} \right\| \leq \gamma_2 \left(1 + \left\| y_{t-p}^t \right\|^{2a}\right).$$

Terme de la forme $\frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{ij}^{-1}} \times \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{kl}^{-1}}$ **:** Il vaut

$$\left((\Gamma_{0_{ij}}) - \left(\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(i)\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(j)\right)\right) \times \left((\Gamma_{0_{kl}}) - \left(\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(k)\left(y_t - F_{W_0}(y_{t-p}^{t-1})\right)(l)\right)\right)$$

dont le module est majoré par

$$2\lambda_{max} \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right) \right\|^2 + \left\| \left(y_t - F_{W_0}(y_{t-p}^{t-1})\right) \right\|^4.$$

Grâce aux conditions (**H**)-1 et (**N**)-1, on en déduit l'existence d'une constante $\gamma_3 > 0$ telle que

$$\left\| \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{ij}^{-1}} \times \frac{\partial \widetilde{U}_{\theta_0}(y_{t-p}^t)}{\partial \Gamma_{kl}^{-1}} \right\| \leq \gamma_3 \left(1 + \left\| y_{t-p}^t \right\|^{2a}\right).$$

Maintenant, en prenant $\gamma_0 = \sup\{\gamma_1, \gamma_2, \gamma_3\}$, on a bien

$$\left\| \nabla \widetilde{U}_{\theta_0}(y_{t-p}^t) \nabla \widetilde{U}_{\theta_0}(y_{t-p}^t)^T \right\| < B \times \gamma_0 \left(1 + \left\| y_{t-p}^t \right\|^{2a}\right)$$

■

La proposition 3 sera donc prouvée si (M_n) satisfait la condition de Lindeberg suivante (Duflo [25]) :

Proposition 4 *En notant $\mathcal{F}_t$ la tribu engendrée par Y_{-p+1}^t, pour tout $\epsilon > 0$ on a :*

$$L_n := \frac{1}{n} \sum_{t=1}^{n} E\left[\left\| \nabla \widetilde{U}_{\theta_0}(y_{t-p}^t) \right\|^2 \mathbb{I}_{\left\{ \left\| \nabla \widetilde{U}_{\theta_0}(y_{t-p}^t) \right\| \geq \epsilon \sqrt{n} \right\}} | \mathcal{F}_{t-1} \right] \xrightarrow{P_{\theta_0}} 0.$$

Preuve Soit $A > 0$ et :

$$F_n(A) := \frac{1}{n}\sum_{t=1}^{n} E\left[\left\|\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)\right\|^2 \mathbb{I}_{\left\{\left\|\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)\right\|\geq\epsilon A\right\}}|\mathcal{F}_{t-1}\right] := \frac{1}{n}\sum_{t=1}^{n} h\left(y_{t-p}^{t-1}, A\right)$$

avec

$$h\left(y_{t-p}^{t-1}, A\right) = E\left[\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)^T\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)\mathbb{I}_{\left\{\left\|\nabla\widetilde{U}_{\theta_0}(y_{t-p}^t)\right\|\geq\epsilon A\right\}}|\mathcal{F}_{t-1}\right].$$

D'après le lemme 2, il existe une constante γ_0 telle que

$$h(y_{t-p}^{t-1}, A) \leq \gamma_0\left(1 + \left\|y_{t-p}^{t-1}\right\|^{2a}\right).$$

Par la loi forte des grands nombres, on a :

$$F_n(A) \xrightarrow{p.s.} \Phi(A) = \int_{\left(\mathbb{R}^d\right)^p} h(y_1^p, A)\mu_0(dy_1^p).$$

Φ est décroissante et positive. Le théorème de convergence dominée montre que, quand A tend vers ∞, $\Phi(A)$ tend vers 0 . Enfin, pour A fixé, on a, si n est assez grand : $\epsilon\sqrt{n} > A$, et $L_n = F_n\left(\epsilon\sqrt{n}\right) \leq F_n(A)$, donc p.s. $\limsup_n L_n \leq \Phi(A)$. Finalement, en faisant tendre $A \longrightarrow \infty$, on obtient p.s. :

$$\lim_{n\longrightarrow\infty} L_n = 0$$

■

On peut maintenant établir le théorème de normalité asymptotique :

Théorème 8 *On suppose satisfaites les hypothèses (**H**) et (**N**) et que le bruit a un moment d'ordre $2a$ avec $a \geq 2$. Alors pour toute loi initiale de la chaîne vectorisée $\left(Y_{t-p+1}^t\right)_{t>0}$:*

$$\sqrt{n}I_0\left[(\hat{\theta}_n) - (\theta_0)\right] \xrightarrow{Loi} N(0, J_0).$$

Preuve Soit V un voisinage de θ_0. Puisque $\hat{\theta}_n \longrightarrow \theta_0$ p.s., il existe $n_0(\omega)$ tel que $\hat{\theta}_n \in V$, pour tout $n \geq n_0(\omega)$. Par la formule de Taylor (avec reste intégral) :

$$0 = \nabla U_n\left(\hat{\theta}_n\right) = \nabla U_n(\theta_0) + \Delta_n(\hat{\theta}_n)(\hat{\theta}_n - \theta_0) \tag{4.15}$$

avec

$$\Delta_n(\hat{\theta}_n) = \int_0^1 HU_n\left[\hat{\theta}_n + u\left(\hat{\theta}_n - \theta_0\right)\right]du.$$

Supposons vérifiée la condition suivante (voir lemme 3) :

$$\Delta_n(\hat{\theta}_n) - HU_n(\theta_0) \xrightarrow{P_{\theta_0}} 0.$$

D'après la proposition 2 qui assure : $HU_n(\theta_0) \xrightarrow{p.s.} I_0$, le théorème est prouvé, puisque alors :

$$\sqrt{n}\Delta_n(\theta_n)\left(\hat{\theta}_n - \theta_0\right) = -\sqrt{n}\nabla U_n(\theta_0)$$

assure que

$$\lim_{n\longrightarrow\infty} \sqrt{n} I_0 \left(\hat{\theta}_n - \theta_0\right) \xrightarrow{en\ loi} N(0, J_0).$$

Lemme 3 *Dans le cadre du théorème 8, on a :*

$$\Delta_n(\hat{\theta}_n) - HU_n(\theta_0) \xrightarrow{p.s.} 0$$

Soit par la proposition 2, ($HU_n(\theta_0) \xrightarrow{p.s.} I_0$) :

$$\Delta_n(\hat{\theta}_n) \xrightarrow{p.s.} I_0$$

Nous allons d'abord d'établir le lemme :

Lemme 4 *Il existe un module de continuité β tel que pour tout $\theta \in V$,*

$$\left\| H\widetilde{U}_\theta(y_1^{p+1}) - H\widetilde{U}_{\theta_0}(y_1^{p+1})) \right\| \leq \beta(\|\theta - \theta_0\|)(1 + \left\|y_1^{p+1}\right\|^a)$$

ce qui implique l'existence d'une constante γ telle que pour tout $\theta \in V$:

$$\left\| H\widetilde{U}_\theta(y_1^{p+1}) \right\| \leq \gamma(1 + \left\|y_1^{p+1}\right\|^a).$$

Preuve Il suffit de vérifier que chaque terme de la dérivée d'ordre 3 par rapport au paramètre de $\widetilde{U}_\theta(y_1^{p+1})$ est dominé par une expression du type $\gamma(1 + \left\|y_1^{p+1}\right\|^a)$, pour $\Theta \in V$.

Terme de la forme $\frac{\partial^3 \widetilde{U}_\theta(y_1^{p+1})}{\partial W_k \partial W_l \partial W_m}$: On a

$$\left\| \frac{\partial^3 \widetilde{U}_\theta(y_1^{p+1})}{\partial W_k \partial W_l \partial W_m} \right\| =$$

$$\left\| (\Gamma_{ij}^{-1})_{ind}^T \left(\frac{\partial^3 F_W(y_1^p)(i)}{\partial W_k \partial W_l \partial W_m} \times \left(y_{p+1} - F_W(y_1^p)\right)(j) - \frac{\partial^2 F_W(y_1^p)(i)}{\partial W_k \partial W_l} \frac{\partial F_W(y_1^p)(j)}{\partial W_m} \right)_{ind} \right.$$

$$-(\Gamma_{ij}^{-1})_{ind}^T \left(\frac{\partial^2 F_W(y_1^p)(j)}{\partial W_k \partial W_m} \times \frac{\partial F_W(y_1^p)(i)}{\partial W_l} + \frac{\partial F_W(y_1^p)(j)}{\partial W_k} \frac{\partial^2 F_W(y_1^p)(i)}{\partial W_l \partial W_m} \right)_{ind}$$

$$\left. +(\Gamma_{ij}^{-1})_{ind}^T \left(\frac{\partial^3 F_W(y_1^p)(j)}{\partial W_k \partial W_l \partial W_m} \times \left(y_{p+1} - F_W(y_1^p)\right)(i) - \frac{\partial^2 F_W(y_1^p)(j)}{\partial W_k \partial W_l} \frac{\partial F_W(y_1^p)(i)}{\partial W_m} \right)_{ind} \right\|$$

$$-(\Gamma_{ij}^{-1})_{ind}^{T}\left(\frac{\partial^2 F_W(y_1^p)(i)}{\partial W_k \partial W_m}\times\frac{\partial F_W(y_1^p)(j)}{\partial W_l}+\frac{\partial F_W(y_1^p)(i)}{\partial W_k}\frac{\partial^2 F_W(y_1^p)(j)}{\partial W_l \partial W_m}\right)_{ind}\Bigg\|,$$

ce qui est majoré par

$$\frac{2}{\lambda_{min}}\sum_{1\le i,j\le d}\left\|\frac{\partial^3 F_W(y_1^p)(i)}{\partial W_k\partial W_l\partial W_m}\times\left(y_{p+1}-F_W(y_1^p)\right)(j)+\frac{\partial^2 F_W(y_1^p)(i)}{\partial W_k\partial W_l}\frac{\partial F_W(y_1^p)(j)}{\partial W_m}\right\|$$
$$+\frac{2}{\lambda_{min}}\sum_{1\le i,j\le d}\left\|\frac{\partial^2 F_W(y_1^p)(j)}{\partial W_k\partial W_m}\times\frac{\partial F_W(y_1^p)}{\partial W_l}+\frac{\partial F_W(y_1^p)(j)}{\partial W_k}\frac{\partial^2 F_W(y_1^p)(i)}{\partial W_l\partial W_m}\right\|.$$

En tenant compte de la condition (**N**)-4, des remarques 8 et 9, ainsi que de la condition (**H**)-1, il existe une constante finie γ telle que :

$$\left\|\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial W_l\partial W_m}\right\|\le\gamma(1+\left\|y_1^{p+1}\right\|^a).$$

Terme de la forme $\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial W_l\partial\Gamma_{ij}^{-1}}$: On a

$$\left\|\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial W_l\partial\Gamma_{ij}^{-1}}\right\|=$$

$$\left\|\frac{\partial^2 F_W(y_{t-p}^{t-1})(i)}{\partial W_k\partial W_l}\times\left(y_t-F_W(y_{t-p}^{t-1})\right)(j)-\frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k}\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_l}\right.$$
$$\left.\frac{\partial^2 F_W(y_{t-p}^{t-1})(j)}{\partial W_k\partial W_l}\times\left(y_t-F_W(y_{t-p}^{t-1})\right)(i)-\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k}\frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_l}\right\|.$$

Grâce aux remarques 8 et 9, ainsi qu'à la condition (**H**)-1, il existe une constante finie γ telle que :

$$\left\|\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial W_l\partial\Gamma_{ij}^{-1}}\right\|\le\gamma(1+\left\|y_1^{p+1}\right\|^a).$$

Termes de la forme $\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial\Gamma_{cd}^{-1}\partial\Gamma_{ij}^{-1}}$ et $\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial\Gamma_{kl}\partial\Gamma_{cd}^{-1}\partial\Gamma_{ij}^{-1}}$: On a

$$\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial W_k\partial\Gamma_{cd}^{-1}\partial\Gamma_{ij}^{-1}}=0$$

et

$$\frac{\partial^3\widetilde{U}_\theta(y_1^{p+1})}{\partial\Gamma_{kl}^{-1}\partial\Gamma_{cd}^{-1}\partial\Gamma_{ij}^{-1}}=\frac{\partial^2(\Gamma_{kl})}{\partial\Gamma_{cd}^{-1}\partial\Gamma_{ij}^{-1}}$$

qui sont constants en y, donc majorés par une expression du type $\gamma(1+\left\|y_1^{p+1}\right\|^a)$ pour $\Theta\in V$ et γ fini.

Les majorations de la dérivée troisième du contraste implique immédiatement l'existence du module de continuité du lemme 4 ■

Preuve du lemme 3 : Pour $\theta \in V$, considérons :

$$n \left\| HU_n(\theta) - HU_n(\theta_0) \right\|$$

Grâce au lemme 4, on a l'inégalité pour tout $\theta \in V$:

$$n \left\| HU_n(\theta) - HU_n(\theta_0) \right\| \leq 2 \sum_{t=1}^{n} \beta(\|\theta - \theta_0\|)(1 + \left\| y_{t-p}^t \right\|^a)$$

donc

$$\left\| \Delta_n(\hat{\theta}_n) - HU_n(\theta_0) \right\| = \left\| \int_0^1 \left\{ HU_n \left[\hat{\theta}_n + u \left(\hat{\theta}_n - \theta_0 \right) \right] - HU_n(\theta_0) \right\} du \right\|$$

est majoré par

$$\beta(\|\theta - \theta_0\|) \frac{2}{n} \sum_{t=1}^{n} (1 + \left\| y_{t-p}^t \right\|^a).$$

Par la loi forte des grands nombres $\frac{2}{n} \sum_{t=1}^{n} (1 + \left\| y_{t-p}^t \right\|^a)$ converge p.s. et comme $\hat{\theta}_n \longrightarrow \theta_0$ p.s., on en déduit la convergence p.s. vers 0 de $\Delta_n(\hat{\theta}_n) - HU_n(\theta_0)$ ■

Remarque 11 *Dans le cas gaussien on peut préciser la forme des matrices I_0 et J_0. En effet, dans ce cas, la martingale (M_n) (cf égalité (4.13)) est la dérivée de l'opposée de la log-vraisemblance, elle est de carré intégrable et le processus croissant qui lui est associé est :*

$$J_n(\theta_0) := \sum_{t=1}^{n} \left(\frac{\left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)} \right) \left(\frac{\left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)} \right)^T$$

De plus la dérivée seconde de l'opposée de la log-vraisemblance est

$$Z_n(\theta_0) := \sum_{t=1}^{n} - \frac{\left(\frac{\partial}{\partial \theta_l} \left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B} \right)_{1 \leq l \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)}$$

$$+ \left(\frac{\left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)} \right) \left(\frac{\left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)} \right)^T$$

*Mais, comme sous les hypothèses (**N**), on peut échanger espérance et dérivation, on a :*

$$E_{\theta_0} \left[\frac{\left(\frac{\partial}{\partial \theta_l} \left(\frac{\partial}{\partial \theta_k} L_{\theta_0}\left(y_{t-p}^t\right) \right)_{1 \leq k \leq B} \right)_{1 \leq l \leq B}}{L_{\theta_0}\left(y_{t-p}^t\right)} \right] = \int_{\left(\mathbb{R}^d\right)^{p+1}} \frac{\partial^2}{\partial \theta_l \theta_k} L_{\theta_0}\left(y_1^{p+1}\right) dy_1^{p+1} = 0$$

donc

$$\lim_{n\to\infty}\frac{1}{n}Z_n(\theta_0) = I_0 = \lim\frac{1}{n}J_n(\theta_0) = J_0$$

où $I_0 = J_0$ est la matrice d'information de Fisher du modèle.

Donc, dans le cas gaussien, $\hat{\theta}_n$ est l'estimateur du maximum de vraisemblance, il est fortement consistant et asymptotiquement efficace.

4.3.4 Vitesse et loi du logarithme itéré

Dans cette section, pour une matrice réelle et symétrique A, $\lambda_{max}A$ (resp. $\lambda_{min}A$) désignera la plus grande (resp. la plus petite) valeur propre de A. Montrons le théorème suivant :

Théorème 9 *Sous les hypothèses (**H**) et (**N**), si le bruit a un moment d'ordre $> 2a$, $a \geq 2$ et si les matrices I_0 et J_0 sont inversibles, on a presque sûrement :*

$$\limsup_n \sqrt{\frac{n}{2\ln\ln n}}\,\|DU_n(\theta_0)\| \leq \sqrt{\lambda_{max}J_0} \tag{4.16}$$

$$\limsup_n \sqrt{\frac{n}{2\ln\ln n}}\,\left\|\hat{\theta}_n - \theta_0\right\| \leq \sqrt{\frac{\lambda_{max}J_0}{\lambda_{min}I_0}}. \tag{4.17}$$

Preuve C'est une adaptation de la preuve de [49].

Pour u, un vecteur de $\mathbb{R}^B$ notons

$$\widetilde{M_n} := \langle M_n, u\rangle = \sum_{t=1}^{n}\left\langle \nabla\widetilde{U}_{\theta_0}(y_{t-p}^t), u\right\rangle$$

C'est une martingale de puissance $2+2\alpha$ intégrable pour tout $\alpha \in]0, \frac{a}{2}-1]$. Notons :

$$T_t = E\left[\left|\widetilde{M}_{t+1} - \widetilde{M}_t\right|^{2+2\alpha}|F_t\right]^{1/(2+2\alpha)}$$

et

$$\tau_n = \sum_{t=1}^{n} T_t^2 = u^T\langle M\rangle_n u.$$

En vertu de (4.14), $\frac{\tau_n}{n} \longrightarrow u^T J_0 u$ presque sûrement, qui est strictement positif car J_0 est supposée inversible. On aura alors $\tau_n \longrightarrow \infty$ presque sûrement. La loi du logarithme itéré pour une martingale de puissance $2+2\alpha$ intégrable (Duflo[26], Corollaire 6) assure que presque sûrement :

$$\limsup_n \frac{\left|\widetilde{M_n}\right|}{\sqrt{2\tau_{n-1}\ln\ln\tau_{n-1}}} \leq 1$$

si la série $\sum \left(\frac{T_n^2}{\tau_n}\right)^{1+\alpha}$ est p.s. convergente.

Posons $s_n := T_1^{2+2\alpha} + \cdots + T_n^{2+2\alpha}$. Pour $\alpha < a/2 - 1$, grâce au lemme 2, on a la loi forte des grands nombres pour $(T_n^{2+2\alpha})$, donc $\frac{s_n}{n} \longrightarrow \gamma \geq 0$ presque sûrement. Par ailleurs $\left(\frac{T_n^2}{\tau_n}\right)^{1+\alpha} \sim Cte \times \frac{T^{2+2\alpha}}{n^{1+\alpha}}$ et par la transformation d'Abel

$$\sum_{t=1}^{n} \frac{T_t^{2+2\alpha}}{t^{1+\alpha}} = \frac{s_n}{n^{1+\alpha}} + \sum_{t=1}^{n-1} \left[\frac{1}{t^{1+\alpha}} - \frac{1}{(t+1)^{1+\alpha}}\right] - s_1.$$

Puisque $\frac{s_n}{n^{1+\alpha}} \longrightarrow 0$ p.s. et

$$\frac{1}{t^{1+\alpha}} - \frac{1}{(t+1)^{1+\alpha}} \sim \frac{1+\alpha}{t^{2+\alpha}},$$

la série $\sum \frac{T^{2+2\alpha}}{n^{1+\alpha}}$ est presque sûrement convergente et il en est de même pour $\sum \left(\frac{T_n^2}{\tau_n}\right)^{1+\alpha}$.

Comme $2\tau_{n-1} \ln\ln \tau_{n-1} \sim 2nu^T J_0 u \ln\ln n$ on obtient presque sûrement :

$$\limsup_n \sqrt{\frac{n}{2\ln\ln n}} \, |\langle DU_n(\theta_0), u\rangle| \leq \sqrt{u^T J_0 u}$$

d'où la L.L.I.

$$\limsup_n \sqrt{\frac{n}{2\ln\ln n}} \, \|DU_n(\theta_0)\| \leq \sqrt{\lambda_{max} J_0}.$$

Pour la seconde L.L.I.

$$\limsup_n \sqrt{\frac{n}{2\ln\ln n}} \left\|\hat{\theta}_n - \theta_0\right\| \leq \sqrt{\frac{\lambda_{max} J_0}{\lambda_{min} I_0}},$$

elle se déduit de la première grâce au développement de Taylor (4.15) et le lemme 3 ■

4.3.5 Identification presque sûre

Suivant la présentation de Guyon [31], on suppose que l'espace des paramètres $\Theta \subset \mathbb{R}^M$ correspond au modèle majorant. Soit $\mathbb{O}$ une famille finie de sous-espaces de $\mathbb{R}^M$, $\delta \in \mathbb{O}$ l'élément générique de $\mathbb{O}$, $|\delta|$ sa dimension et $\Theta_\delta := \Theta \cap \delta$ le sous-espace (sous-modèle) paramétrique associé. On suppose que la vraie valeur est $\theta_{0,\delta_0} \in \Theta_{\delta_0}$, $\delta_0 \in \mathbb{O}$ étant le sous-espace minimal associé à θ_{0,δ_0}. Soit $(c(n))$ une suite positive. Au vu de la réalisation $(Y_t)_{-p<t\leq n}$, on utilise comme fonction de décision le contraste pénalisé à la vitesse $c(n)$ par la dimension du modèle :

$$CP_{n,\delta}(\theta) := U_n(\theta_\delta) + \frac{c(n)}{n} |\delta| \tag{4.18}$$

pour $\delta \in \mathbb{O}$ et $\theta_\delta \in \Theta_\delta$.

Notons : $\overline{CP}_{n,\delta} = \overline{U}_{n,\delta} + \frac{c(n)}{n} |\delta|$ avec $\overline{U}_{n,\delta} = U_n(\hat{\theta}_{n,\delta})$ et $\hat{\theta}_{n,\delta} = \arg\min_{\theta_\delta \in \Theta_\delta} U_n(\theta_\delta)$. On choisira $\hat{\delta}_n = \arg\min_{\delta \in \mathbb{F}} \overline{CP}_{n,\delta}$, qui répond au principe de parcimonie d'Akaiké avec la vitesse $c(n)$.

Appliquant les résultats de (Senoussi [57], Guyon [31]), nous avons le résultat suivant d'identification presque sûre du vrai modèle δ_0.

Théorème 10 *On se place dans le cadre du théorème 9. Si la vitesse de pénalisation $c(n)$ est telle que :*

$$\lim_n \frac{c(n)}{n} = 0$$

et

$$\liminf_n \frac{c(n)}{2 \ln \ln n} > \frac{\lambda_{max} J_0}{2 \lambda_{min} I_0}$$

alors, le couple $\left(\hat{\delta}_n, \hat{\theta}_{n,\delta_n}\right)$ converge P_{θ_0}-p.s. vers la vraie valeur $(\delta_0, \theta_{0,\delta_0})$.

Preuve Il suffit d'appliquer le théorème (3.4.8) de (Guyon [31]) dont les conditions d'application se vérifient immédiatement ici grâce au théorème 9.

4.4 Application au perceptron multicouches

Nous allons montrer comment les résultats obtenus s'appliquent au MLP.

4.4.1 Calcul de la dérivée du contraste

Pour obtenir les paramètres (poids) du MLP, $W \in \mathbb{R}^D$, qui minimisent le contraste associé à la vraisemblance, il faut minimiser la fonction $W \to \ln(\det(\Gamma))$.

On rappelle que par l'équation 4.8, le gradient de cette fonction s'écrit :

$$\left(\frac{\partial}{\partial W_k}(\ln(\det(\Gamma)))\right)_{1 \le k \le D}$$

avec

$$\frac{\partial}{\partial W_k}(\ln(\det(\Gamma))) = \left(\Gamma_{ij}^{-1}\right)_{ind}^T \left(\frac{\Gamma_{ij}}{\partial W_k}\right)_{ind}$$

et

$$\frac{\partial \Gamma_{ij}}{\partial W_k} = \sum_{t=1}^{n} \left[-\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k} \times \left(y_t - F_W(y_{t-p}^{t-1})\right)(j) - \frac{\partial F_W(y_{t-p}^{t-1})(j)}{\partial W_k} \left(y_t - F_W(y_{t-p}^{t-1})\right)(i) \right].$$

Il suffit donc de savoir calculer $\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k}$ pour pouvoir exprimer ce gradient.

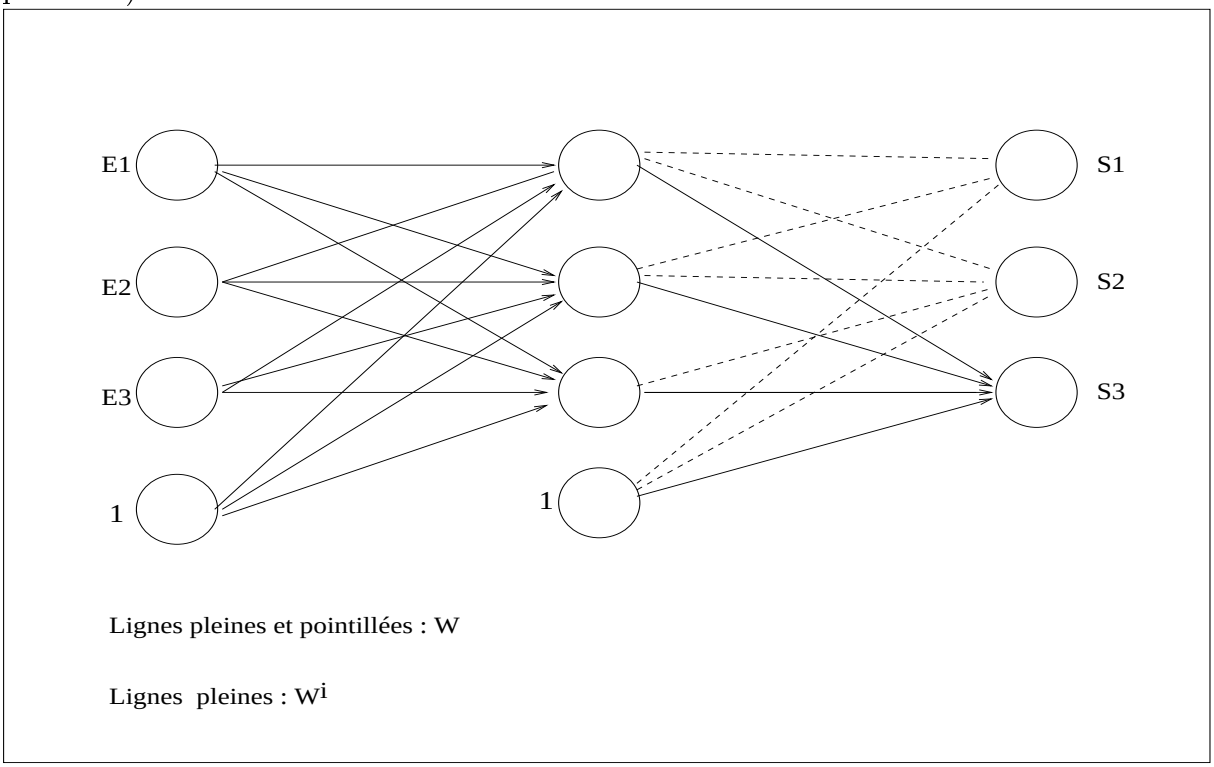
Calcul de $\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k}$: Soit un MLP à sortie vectorielle F_W, considérons le MLP extrait F_{Wi} qui a les mêmes poids que F_W avant la dernière unité cachée, mais qui ne garde que les poids pointant sur la sortie i de F_W . Le MLP extrait n'a donc plus qu'une sortie.

La figure 4.2 décrit un MLP avec trois entrées, trois sorties et $i = 3$ (ce qui correspond à la troisième sortie).

Pour calculer $\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k}$, il suffit de calculer par rétro-propagation classique la dérivée [2] de la fonction représentée par le MLP extrait F_{Wi} : $\frac{\partial F_{Wi}(y_{t-p}^{t-1})}{\partial W_k^i}$. Finalement $\frac{\partial F_W(y_{t-p}^{t-1})(i)}{\partial W_k}$sera égale à $\frac{\partial F_{Wi}(y_{t-p}^{t-1})}{\partial W_k^i}$pour les poids en commun et sera nulle pour les autres poids.

On peut alors approcher le minimum du contraste associé à la vraisemblance par optimisation différentielle.

FIG. 4.2 – MLP extrait F_W3 (lignes pleines) du MLP F_W (lignes pleines et lignes pointillés)



[2] la dérivée $\frac{\partial F_{Wi}(y_{t-p}^{t-1})}{\partial W_k^i}$ s'obtient en commençant la rétropropagation en fixant l'erreur de prévision égale à 1.

4.4.2 Identification presque sûre du modèle

Considérons des MLP dans l'espace où ils sont identifiables (cf section 2.1.3). L'étude des propriétés statistiques de l'estimateur de minimum de contraste associé à la vraisemblance, nous permet de déduire le théorème suivant :

Théorème 11 *Soit un modèle correspondant au modèle (8.1) avec un MLP pour fonction F_W. Supposons satisfaites les conditions suivantes :*

1. *$(\varepsilon_t)_{t\in\mathbb{N}^*}$ est une suite de variables aléatoires de $\mathbb{R}^d$ centrées, de matrice de covariance Γ_0 définie positive, i.i.d. et indépendantes de l'état initial de la chaîne $\left(Y_{t-p+1}^t\right)_{t\in\mathbb{N}}$. ε_1 a une densité positive par rapport à la mesure de Lebesgue sur $\mathbb{R}^d$, avec, pour $\eta > 0$, $E\left(\|\varepsilon_1\|^{12+\eta}\right) < \infty$.*
2. *$\theta = (W, \Gamma^{-1})$ appartient à un sous-ensemble compact Θ :*

$$\Theta = \Theta^W \times \Theta^{\Gamma^{-1}} \subset \left(\bigcup_{C=0}^{M} \mathcal{N}_{d\times p,C,d}/\mathcal{R}\right) \times \mathbb{R}^{(d+1)d/2}$$

   où M est le nombre maximal d'unités cachées pour les MLP considérés.
3. *Pour toute matrice $\Gamma^{-1} \in \Theta^{\Gamma^{-1}}$, $\rho(\Gamma) > 0$.*
4. *On suppose qu'il existe $0 \leq m_0 \leq M$ tel que le vrai modèle appartient à $\Theta \cap \mathcal{N}_{d\times p,m_0,d}$.*
5. *Les matrices I_0 et J_0 du théorème 8 sont supposées définies positives.*
6. *La vitesse de pénalisation $c(n)$ du contraste pénalisé $CP_{n,\delta}(\theta)$ est telle que :*

$$\lim_{n\longrightarrow\infty} \frac{c(n)}{n} = 0$$

   et

$$\liminf_{n\longrightarrow\infty} \frac{c(n)}{2\ln\ln n} > \frac{\lambda_{max} J_0}{2\lambda_{min} I_0}.$$

Alors le couple $\left(\hat{\delta}_n, \hat{\theta}_{n,\delta_n}\right)$ converge P_{θ_0}-p.s. vers la vraie valeur $(\delta_0, \theta_{0,\delta_0})$.

Preuve : Il est aisé de montrer que la fonction F_W est bornée pour tout $W \in \Theta^W$, on aura de plus pour tout $y_1^p \in \left(\mathbb{R}^d\right)^p$(cf Yao et Mangeas [64]) :

$$\|\nabla F_{W_0}(y_1^p)\| \leq Cte(1 + \|y_1^p\|)$$

$$\|HF_{W_0}(y_1^p)\| \leq Cte(1 + \|y_1^p\|^2)$$

et en notant TF_W la dérivée troisième par rapport aux paramètres de F_W , en remarquant que Θ^W est compact, il existera une constante γ telle que :

$$\forall W \in \Theta^W,\ \|TF_W(y_1^p)\| \leq \gamma(1 + \|y_1^p\|^3).$$

Comme on doit avoir $\|TF_W(y_1^p)\| \leq \gamma(1 + \|y_1^p\|^{a/2})$, avec un bruit ayant un moment d'ordre strictement supérieur à $2a$, il faut donc que le bruit possède un moment strictement supérieur à 12 dans le cas du perceptron. Il est facile de voir que les hypothèses relatives à la continuité sont satisfaites. De plus le modèle est identifiable grâce au théorème 4 du chapitre 2. Le théorème 11 est alors une conséquence du théorème 10 ■

Remarque 12 *Si γ est une constante positive, un terme de pénalisation tel que $c(n) = \gamma \ln(n)$ satisfait les conditions du théorème 11. Si on choisit $\gamma = 1$, on aboutit alors à un critère de sélection de modèle du type (on rappelle que B est le nombre de paramètres du modèle) :*

$$CP_{n,\delta}(\theta) = U_n(\theta) + \frac{\ln(n)}{n} B = \ln \det \left(\Gamma_n(W)\right) + \frac{\ln(n)}{n} B + Cte$$

ce qui revient à minimiser :

$$\widetilde{CP}_{n,\delta}(\theta) = \ln \det \left(\Gamma_n(W)\right) + \frac{\ln(n)}{n} B$$

et cela correspond exactement au critère BIC.

4.5 Conclusion

On a montré que la fonctionnelle à minimiser pour tenir compte des termes non diagonaux de la matrice de covariance du bruit est le logarithme du déterminant de sa matrice de covariance empirique. Cet estimateur est celui du maximum de vraisemblance pour un bruit gaussien. Cependant, même si le bruit n'est pas gaussien, on a montré que cet estimateur avait de bonnes propriétés statistiques. En suivant la même démarche que Yao et Mangeas [64], nous avons montré qu'un contraste pénalisé de type BIC, convergeait p.s. vers le vrai modèle. Cela permet, par exemple pour les perceptrons multicouches, d'estimer et d'identifier le modèle grâce à un algorithme de type stepwise descendant (cf section 2.2.3).

Chapitre 5

Introduction aux modèles autorégressifs à changements de régime markoviens.

5.1 Chaînes de Markov cachées

La théorie des chaînes de Markov cachées et leurs premières applications en reconnaissance de la parole datent de plus de 30 ans. La théorie de base a été publiée dans une suite d'articles de Baum et ses collègues ([5], [6], [7], [8] et [9]) à la fin des années 60 et au début des années 70. Les chaînes de Markov cachées ont ensuite été appliquées à la reconnaissance de la parole dans les années 70 par Baker [3] puis Jelinek [38] ainsi que Bourlard et Morgan [12]. Ces modèles ont été aussi utilisés en génétique, biologie et biochimie (cf Krogh et al. [42], Leroux et Putterman [47]).

5.1.1 Chaînes de Markov dans un espace discret

On considère (X_t), $t \in \mathbb{Z}$ une chaîne de Markov homogène à valeurs dans un espace d'état fini $\mathbb{E} = \{e_1, ..., e_N\}$, $N \in \mathbb{N}^*$. Sans perte de généralité, on identifie l'espace d'état $\mathbb{E} = \{e_1, \cdots, e_N\}$ avec le simplexe de $\mathbb{R}^N$ où e_i est un vecteur unité de $\mathbb{R}^N$ avec 1 sur la i-ème composante et 0 partout ailleurs.

La chaîne X_t est caractérisée par sa matrice de transition [1] $A = (a_{ij})$ qui est telle que :

$$P(X_{t+1} = e_i \,|X_t = e_j) = a_{ij}$$

[1] La notation traditionnelle d'un matrice de transition est plutôt $a_{ij} = P(X_{t+1} = e_j / X_t = ei)$ cependant la notation transposée utilisée ici et empruntée à Elliott permet une notation du modèle complet plus confortable.

Ainsi si on définit : $V_{t+1} := X_{t+1} - AX_t$, on obtient l'équation du modèle :

$$X_{t+1} = AX_t + V_{t+1}$$

ce qui est similaire à celle d'un processus autorégressif.

5.1.2 Observations dans un espace discret

Le processus (X_t), $t \in \mathbb{Z}$, n'est pas observé directement. On suppose qu'il existe une fonction $c(.,.)$ à valeurs finies telle que l'on observe les valeurs

$$Y_t = c(X_t, \varepsilon_t), \; t \in \mathbb{Z} \tag{5.1}$$

où (ε_t), $t \in \mathbb{Z}$, est une suite i.i.d. et indépendantes de V_t, $t \in \mathbb{Z}$. Supposons que l'image de $c(.,.)$ consiste en M points, alors sans perte de généralité on peut identifier cette image avec l'ensemble de vecteurs unités

$$S_Y = \{f_1, \cdots, f_M\}$$

où $f_j = (0, \cdots, 0, 1, 0, \cdots, 0) \in \mathbb{R}^M$, avec l'élément unité en position j. On suppose de plus que $c(.,.)$ est indépendant du temps t. Maintenant (5.1) implique

$$\mathbb{P}(Y_t = f_i \,|X_l, Y_l, \, l \leq t) = \mathbb{P}(Y_t \,|X_t).$$

notons

$$C = (c_{ij}) \in \mathbb{R}^{M \times N}, \; c_{ij} = \mathbb{P}(Y_t = f_i \,|X_t = e_j)$$

avec $\sum_{i=1}^{M} c_{ij} = 1$ et $c_{ij} \geq 0$. On a alors

$$E[Y_t \,|X_t] = CX_t.$$

Si on définit $W_{t+1} := Y_{t+1} - CX_t$ on obtient l'équation

$$Y_t = CX_t + W_t.$$

Le modèle de chaîne de Markov cachée discrète aura alors pour équations

$$\begin{cases} X_t = AX_{t-1} + V_t \\ Y_t = CX_t + W_t \end{cases}$$

avec $t \in \mathbb{Z}$.

5.1.3 Observations dans un espace continu

On peut généraliser les modèles précédents de la manière suivante. On suppose toujours que (X_t) , $t \in \mathbb{Z}$ n'est pas observé directement, mais influence le processus $(Y_t)_{t\in\mathbb{Z}}$ de $\mathbb{R}^d$ tel que le modèle ait pour équations

$$\begin{cases} X_t = AX_{t-1} + V_t \\ Y_t = c(X_t) + \varepsilon_t(X_t) \end{cases}$$

où $t \in \mathbb{Z}$ et où $c(.)$ est une fonction de $\mathbb{E} \longrightarrow \mathbb{R}^d$ déterminée par son graphe $(X_i, f_i = c(X_i))_{1\leq i\leq N}$.

Enfin, pour tout $e_i \in \mathbb{E}$, $(\varepsilon_t(e_i))$, $t \in \mathbb{Z}$ est une suite i.i.d. centrée de $\mathbb{R}^d$, les suites $(\varepsilon_t(e_i))_{1\leq i\leq N}$ étant indépendantes.

5.1.4 Modèle à changements de régime markoviens

On généralise encore les modèles précédents en supposant maintenant que Y_t dépend non seulement de X_t mais aussi des observations $Y_{t-p}^{t-1} = (Y_{t-p}, \cdots, Y_{t-1})$ pour $p \in \mathbb{N}^*$. A un instant t fixé : $Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + \varepsilon_{t+1}(X_{t+1})$ avec

1. Pour tout $e_i \in \mathbb{E}$, $F_{e_i} \in \{F_{e_1}, ..., F_{e_N}\}$ est une fonction borélienne de $\left(\mathbb{R}^d\right)^p \mapsto \mathbb{R}^d$.
2. Pour tout $e_i \in \mathbb{E}$, $(\varepsilon_t(e_i))_{t\in\mathbb{Z}}$ est une suite i.i.d., centrées, de $\mathbb{R}^d$. Les suites $(\varepsilon_t(e_i))_{t\in\mathbb{Z}}$ sont indépendantes.

Les équations du modèle sont donc

$$\begin{cases} X_{t+1} = AX_t + V_{t+1} \\ Y_{t+1} = F_{X_{t+1}}(Y_{t-q+1}^t) + \varepsilon_{t+1}(X_{t+1}) \end{cases}$$

avec $t \in \mathbb{Z}$.

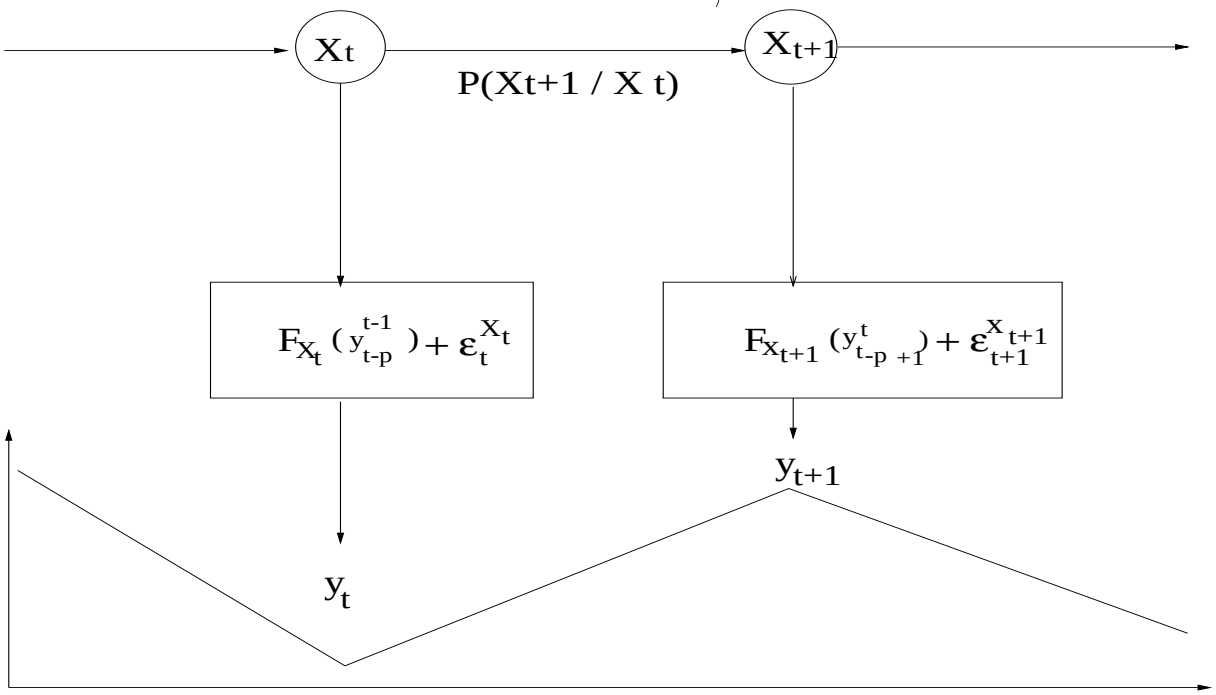
Le principal intérêt de ce modèle est de pouvoir modéliser des séries non-stationnaires par morceaux. Ainsi ces modèles ont déjà été appliqués essentiellement lorsque les fonctions de régression sont linéaires pour des modélisations économiques et physiques.

Hamilton [33] a étudié de tels modèles afin de modéliser des séries temporelles sujettes à des changements discrets de régimes pour analyser la série GNP (Gross National Product) aux Etats-Unis. Il a utilisé ce modèle en supposant que cette série a un régime pour les périodes de croissance économique, un autre pour les périodes de récession.

Un autre domaine d'application de ces modèles est la reconnaissance de la parole (cf Poritz [52]). Par contre il n'y a pratiquement aucune étude pour des fonctions de régression non-linéaires, alors que cette généralisation semble très naturelle. Cela pourrait notamment être très utile, lorsque le processus semble fortement non-linéaire, ce qui est le cas en reconnaissance de la parole.

La figure 5.1 schématise ce modèle.

FIG. 5.1 – Modèle HMM/MLP



5.2 Modèles hybrides MLP/HMM

L'idée est d'utiliser un mélange de N experts, chaque expert étant un MLP, et de trouver un moyen de dire pour un instant donné lequel de ces experts fait la prévision la plus pertinente. Ce problème a par exemple été traité par A.S Weigend, M. Mangeas, A.N. Srivastava [61], grâce à un MLP "porte" qui utilise la série et les erreurs de prédiction de chaque expert pour deviner dans quel état on se trouve (c'est-à-dire quel expert fait la prévision la plus pertinente).

Ici, on choisit de remplacer le MLP "porte" par une chaîne de Markov cachée. L'avantage potentiel des chaînes de Markov cachées sur un MLP "porte" est que la segmentation est locale avec le MLP "porte" (la probabilité des états ne dépend que de ses entrées), mais est globale avec la chaîne de Markov cachée (la probabilité des états dépend de toutes les observations). Nous allons donc appliquer ce modèle pour la prévision de séries temporelles.

5.2.1 Le modèle considéré pour cette étude

Par souci de clarté, nous nous restreignons ici au cas scalaire, mais le cas multidimensionnel s'en déduit facilement et sera étudié ultérieurement.

Soit (X_t), $t \in \mathbb{N}$ une chaîne de Markov homogène dans le temps à valeurs dans un espace d'état fini $\mathbb{E} = \{e_1, ..., e_N\}$, et (Y_t) la série des observations. On considère le modèle suivant à un instant t fixé :

$$Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + \sigma_{X_{t+1}}\varepsilon_{t+1}$$

avec $F_{X_{t+1}} \in \{F_{e_1}, ..., F_{e_N}\}$ une fonction de régression d'ordre p représentée par l'un des MLP expert. Dans ce cas F_{e_i} représente le i-ème MLP, paramétré par le vecteur des poids W_i. On notera indifféremment F_{e_i} ou F_{W_i} la fonction de régression d'ordre p représenté par le i-ème MLP. $\sigma_{X_{t+1}} \in \{\sigma_{e_1}, ..., \sigma_{e_N}\}$ est un nombre réel strictement positif et $(\varepsilon_t)_{t\in\mathbb{N}^*}$ une suite i.i.d gaussienne centrée réduite, la densité de probabilité de $\sigma_{e_i}\varepsilon$ sera notée Φ_i.

Avec les notations de la section précédente, on obtient les équations générales du modèle :

$$\begin{cases} X_{t+1} = AX_t + V_{t+1} \\ Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + \sigma_{X_{t+1}}\varepsilon_{t+1} \end{cases} \tag{5.2}$$

Pour estimer une série qui vérifie (5.2), on prendra pour distribution initiale des états de la chaîne de Markov cachée π_0 la loi uniforme sur $\mathbb{E}$. De plus le conditionnement par rapport aux observations initiales y_{-p}^0 sera toujours implicite.

5.2.2 Estimation du modèle

5.2.2.1 La vraisemblance

Les paramètres θ du modèle sont ici les poids des N experts (W_i), les N variances des modèles (σ_i), les coefficients de la matrice de transition (a_{ij}) que l'on suppose tous strictement positif.

La vraisemblance du modèle pour une suite d'observations de la série $y := (y_{-p}^T)$, pour un chemin réalisé $x := \{x_t,\ t = 1, ..., T\}$ s'écrit :

$$L_\theta(y, x) = \prod_{t=1}^{T}\prod_{i=1}^{N}\left[\Phi_i(y_t - F_{W_i}(y_{t-p}^{t-1}))\right]^{1_{\{e_i\}}(x_t)} \times \prod_{t=1}^{T}\prod_{i,j=1}^{N} a_{ij}^{1_{\{e_j,e_i\}}(x_t,x_{t+1})} \times \pi_0(X_1).$$

La vraisemblance globale des observations est donc :

$L_\theta(y) = \sum_x L_\theta(y, x)$, où $\sum_x$ représente la somme sur tous les chemins possibles de la chaîne de Markov cachée.

5.2.2.2 Maximisation de la vraisemblance

On montre facilement que le calcul exact de la log-vraisemblance par cette méthode est de complexité $O(N^T)$, mais l'algorithme E.M. (Demster et al. [21]) permet de trouver un suite de paramètres qui augmente la vraisemblance à chaque itération et converge vers un maximum local de la vraisemblance pour une très grande classe de modèles dont le modèle (5.2).

Rappelons la définition de l'algorithme E.M.

L'algorithme E.M. (Expectation/Maximisation)

1. Initialisation : Poser $k = 0$ et choisir θ_0
2. E-Step : Poser $\theta^* = \theta_k$ et calculer $Q(., \theta^*)$ avec

$$Q(\theta, \theta^*) = E_{\theta^*}\left[\ln\left(\frac{L_\theta(Y,X)}{L_{\theta^*}(Y,X)}\right)|y\right]$$

3. M-Step : Trouver

$$\hat{\theta} = \arg\max Q(\theta, \theta^*)$$

4. Poser θ_{k+1} par $\hat{\theta}$, et recommencer à l'étape 2) jusqu'à ce qu'un critère d'arrêt soit satisfait.

La suite créée $(\theta_k, k \geq 0)$ donne des valeurs croissantes de la fonction de vraisemblance $L_\theta(y)$, puisque d'après l'inégalité de Jensen :

$$\begin{array}{ll} \ln\left(L_{\widehat{\theta}^*}(y)\right) - \ln\left(L_{\theta^*}(y)\right) & = \ln\left(\int_x L_{\widehat{\theta}^*}(x,y)d\mu(x)/L_{\theta^*}(y)\right) \\ & = \ln\left(\int_x \frac{L_{\theta^*}(x,y)}{L_{\theta^*}(y)}\frac{L_{\widehat{\theta}^*}(x,y)}{L_{\theta^*}(x,y)}d\mu(x)\right) \\ (Jensen) & \geq \int_x \ln\left(\frac{L_{\widehat{\theta}^*}(x,y)}{L_{\theta^*}(x,y)}\right)\frac{L_{\theta^*}(x,y)}{L_{\theta^*}(y)}d\mu(x) \\ & = L_{\theta^*}(y)^{-1}Q(\widehat{\theta}^*, \theta^*) \geq 0 \end{array}$$

La suite (θ_k) converge alors vers un maximum local de la vraisemblance. Dans la suite, on appellera la fonction $Q(\theta, \theta^*)$ la pseudo-log-vraisemblance, elle se calcule facilement grâce à l'algorithme de Baum et Welch.

Calcul de la pseudo-log-vraisemblance (E-Step) pour θ^* fixé on a :

$$E_{\theta^*}\left[\ln L_\theta(Y,X) - \ln L_{\theta^*}(Y,X)\,|y\right]$$

$$= E_{\theta^*}\left[\left(\sum_{t=1}^{T}\sum_{i,j=1}^{N} 1_{\{e_j,e_i\}}(X_{t-1},X_t)\ln a_{ij} + \sum_{t=1}^{T}\sum_{i=1}^{N} 1_{\{e_i\}}(X_t)\left[\ln\Phi_i(Y_t - F_{W_i}(Y_{t-p}^{t-1}))\right]\right)|y\right] + Cte.$$

Ainsi, en notant :

$$\omega_t(e_i) = P_{\theta^*}(X_t = e_i \,|y)$$

et

$$\omega_t(e_j, e_i) = P_{\theta^*}(X_{t+1} = e_j,\, X_t = e_i \,|y),$$

on aura :

$$E_{\theta^*}\left[(\ln L_\theta(Y, X) - \ln L_{\theta^*}(Y, X))\,|y\right]$$

$$= \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \omega_t(e_j, e_i) \ln a_{ij} + \sum_{t=1}^{T} \sum_{i=1}^{N} \omega_t(e_i) \left[\ln \Phi_i(y(t) - F_{W_i}(y_{t-p}^{t-1}))\right] + Cte.$$

La pseudo-log-vraisemblance s'exprime donc comme la somme de deux termes :

$$U_\theta = \sum_{t=1}^{T-1} \sum_{i,j=1}^{N} \omega_t(e_j, e_i) \ln a_{ij}$$

et

$$V_\theta = \sum_{t=1}^{T} \sum_{i=1}^{N} \omega_t(e_i) \left[\ln \Phi_i(y(t) - F_{W_i}(y_{t-p}^{t-1}))\right]$$

où U_θ qui de ne dépend que des (a_{ij}) , et V_θ qui ne dépend que des w_i et des σ_i .

Pour exprimer U_θ et V_θ, on calcule $\omega_t(e_i)$ et $\omega_t(e_j, e_i)$ grâce à l'algorithme forward-backward de Baum et Welch (Rabiner [54]).

Dynamique forward Si on note : $\alpha_t(e_i) = L_{\theta^*}(X_t = e_i, y_1^t)$ la vraisemblance de l'état e_i à l'instant t et des observations y_1^t, on a la récurrence forward :

$$\alpha_{t+1}(e_i) = \left(\sum_{j=1}^{N} \alpha_t(e_j) \times a_{ij}^*\right) \times \Phi_i^*(y_{t+1} - F_{W_i^*}(y_{t-p+1}^t)).$$

Dynamique backward Si on note : $\beta_t(e_j) = L(y_{t+1}^T \,|X_t = e_j)$ on a la récurrence backward :

$$\beta_t(e_j) = \sum_{i=1}^{N} \Phi_i^*(y_{t+1} - F_{W_i^*}(y_{t-p+1}^t))\beta_{t+1}(e_i) \times a_{ij}^*.$$

On aura alors les résultats :

$$\omega_t(e_i) = \frac{\alpha_t(e_i)\beta_t(e_i)}{\sum_{i=1}^{N} \alpha_t(e_i)\beta_t(e_i)}$$

ainsi que :

$$\omega_t(e_j, e_i) = \frac{\alpha_t(e_j)a^*_{ij}\Phi^*_i(y_{t+1} - F_{W_i^*}(y^t_{t-p+1}))\beta_{t+1}(e_i)}{\sum_{i,j=1}^{N} \alpha_t(e_j)a^*_{ij}\Phi^*_i(y_{t+1} - F_{W_i^*}(y^t_{t-p+1}))\beta_{t+1}(e_i)}.$$

Maximisation de la pseudo-log-vraisemblance (M-step) Pour maximiser la pseudo-log-vraisemblance, il suffit donc de maximiser séparément U_θ et V_θ .

Maximum de U_θ : On trouve, en maximisant U_θ (cf Rabiner [54])

$$\widehat{a}_{ij} = \frac{E(nb\ transitions\ e_j \to e_i\,|y)}{E(temps\ d'occupation\ de\ e_j\,|y)}$$

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \omega_t(e_i, e_j)}{\sum_{t=1}^{T-1} \omega_t(e_j)}.$$

Maximum de V_θ : Puisque l'on a supposé l'innovation gaussienne, il faut maximiser :

$$V_\theta = \sum_{t=1}^{T}\sum_{i=1}^{N} \omega_t(e_i)\left[\frac{(y_t - F_{W_i}(y^{t-1}_{t-p}))^2}{2\sigma_i^2} + \ln\left(\sqrt{2\pi}\sigma_i\right)\right].$$

Ceci se résout aisément grâce à un apprentissage, pour chaque expert F_{W_i}, sur la fonction de coût pondérée par la probabilité à tout instant de l'état e_i :

$$\hat{W}_i = \arg\min \sum_{t=1}^{T} \omega_t(e_i)\left[(y_t - F_{W_i}(y^{t-1}_{t-p}))^2\right].$$

On en déduit les variances $\hat{\sigma}_i^2$ de chaque modèle, en posant :

$$\hat{\sigma}_i^2 = \frac{1}{\sum_{t=1}^{T} \omega_t(e_i)} \sum_{t=1}^{T} \omega_t(e_i)\left[(y_t - F_{\hat{W}_i}(y^{t-1}_{t-p}))^2\right].$$

5.2.3 Application à la série laser

On utilise la série laser complète de "Santa Fe time series prediction and analysis competition" (cf [62]). Le niveau de bruit est ici très bas, la principal source de bruit vient des erreurs de mesures. La longueur totale de la série est de 12500, on garde 11500 données pour l'apprentissage et les 1000 dernières données sont gardées pour valider le modèle appris.

5.2.3.1 Procédure d'estimation

L'estimation de ce genre de modèle soulève plusieurs questions :

1. Le nombre d'experts à utiliser : On ne connaît pas a priori le nombre de régimes différents de la série. On fera donc une estimation avec 2 experts pour commencer, puis on rajoutera un expert jusqu'à ce que l'erreur de prédiction du modèle augmente.
2. L'initialisation : La matrice de transition sera toujours initialisée avec tous ses éléments égaux pour n'avantager aucun expert au début. Dans le même souci de ne pas avantager un expert dès le début, les poids des experts sont initialisés très proches de zéro, ici on utilise une loi uniforme entre -0.01 et 0.01.
3. L'apprentissage : La phase de maximisation de la pseudo-log-vraisemblance est délicate, puisque qu'on peut difficilement savoir quand les experts ont atteint un minimum local de leur fonction de coût pondérée. Pour que l'algorithme garde un temps de calcul raisonnable on considérera qu'un bonne approximation du M-step est obtenue avec un apprentissage comprenant 10 itérations de l'algorithme de Levenberg-Marquart, car cet algorithme converge très rapidement. On voit empiriquement qu'un minimum local semble atteint à chaque itération de l'algorithme E.M., sauf peut-être pour les 3 premières itérations de l'E.M.
4. Le problème des mauvais minima locaux : Wu [63] a montré que même si on peut trouver un maximum global à la pseudo-log-vraisemblance à chaque itération de l'algorithme E.M., celui-ci ne converge que vers un maximum local de la log-vraisemblance. Ici les choses sont pires puisque qu'on ne trouve qu'un maximum local de la pseudo-log-vraisemblance. Il est donc important de recommencer plusieurs fois l'apprentissage avec des initialisations différentes. Ici 10 apprentissages différents ont suffit à trouver des résultats intéressants.

5.2.3.2 Estimation avec deux experts

On choisit d'utiliser 2 experts avec 10 entrées, 5 unités cachées, et une sortie linéaire, les fonctions d'activation sont des tangentes hyperboliques. On suppose donc que la chaîne de Markov cachée a deux états e_1 et e_2. La matrice de transition initiale vaut :

$$A_0 = \begin{pmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{pmatrix}$$

On procède à 200 itérations de l'algorithme E.M. Le but de l'apprentissage est de découvrir les éventuels régimes de la série et de pouvoir prévoir les effondrements, ainsi que la prévision de la valeur suivante de la série.

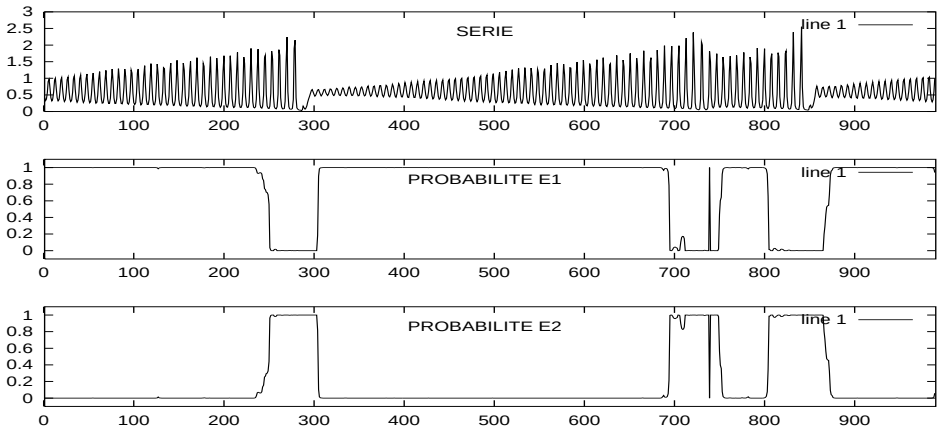
Estimation de la probabilité des modèles : Après apprentissage, la matrice de transition estimée est :

$$\hat{A} = \begin{pmatrix} 0.994 & 0.025 \\ 0.006 & 0.975 \end{pmatrix}$$

La figure 5.2 représente la série et en dessous, les probabilités des états conditionnellement aux observations. Le modèle fait apparaître très clairement deux états sur la série de validation.

Le premier état correspond au régime normal de la série, alors que le deuxième correspond au régime d'effondrement de la série.

FIG. 5.2 – Espérance conditionnelle des états sur la série de validation



La figure 5.3 représente la prévision des états à chaque instant c'est-à-dire :

$$\hat{Q}_{t+1} = A(Q_t)$$

où Q_t est l'estimation forward de l'état, c'est-à-dire (en reprenant les notations de la section 5.2.2.2) :

$$Q_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)}.$$

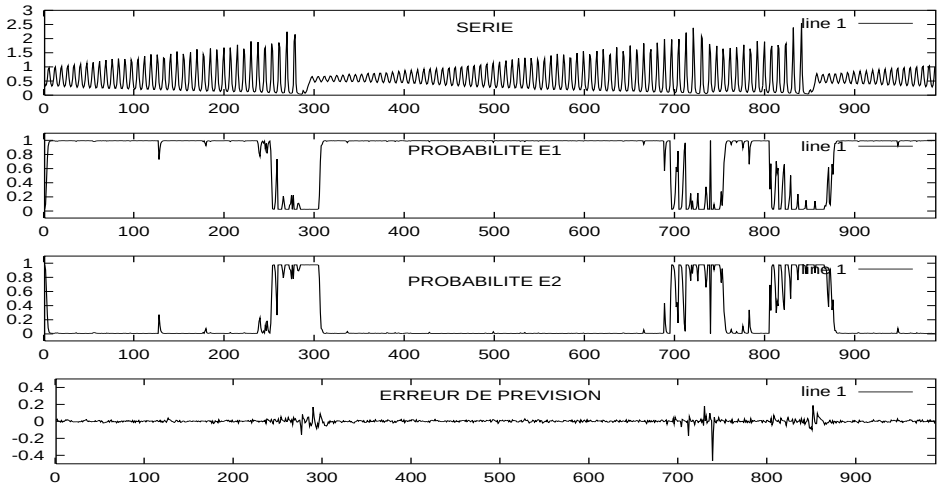
Cette prévision diffère de l'espérance conditionnelle car on n'utilise que les données disponibles à l'instant t pour prévoir l'état à l'instant $t+1$. La prévision par le modèle de la valeur à l'instant $t+1$ de la série, sachant les valeurs jusqu'à l'instant t, s'obtient aisément par :

$$\hat{Y}_{t+1} = \sum_{i=1}^{N} \hat{Q}_{t+1}(i) F_i(y_{t-p+1}^{t})$$

La prévision naturellement est moins bonne que l'espérance conditionnelle, mais il reste clair que quand le modèle prévoit que l'on passe à l'état 2, un effondrement est imminent.

L'erreur quadratique moyenne de prévision, normalisé par la variance de la série (E.N.M.S) est de 0.0033 sur la série de validation .

FIG. 5.3 – Estimation forward des états sur la série de validation



5.2.3.3 Estimation avec 3 experts

L'architecture des experts reste la même, on obtient alors sur la série de validation les résultats visualisés sur les figures 5.4 et 5.5.

On y voit clairement apparaître une segmentation en trois états : un état de régime normal, un état de régime pré-effondrement, un état de régime d'effondrement. La matrice estimée est ici :

$$\hat{A} = \begin{pmatrix} 0.9548 & 0.0002 & 0.0204 \\ 0.0356 & 0.9955 & 0.0000 \\ 0.0096 & 0.0043 & 0.9796 \end{pmatrix}$$

L'E.N.M.S. du modèle sur la base de validation est de 0.0046, elle est un peu plus grande que celle avec deux experts, il ne semble donc pas utile d'estimer un modèle avec plus d'experts.

FIG. 5.4 – Espérance conditionnelle des états sur la série de validation

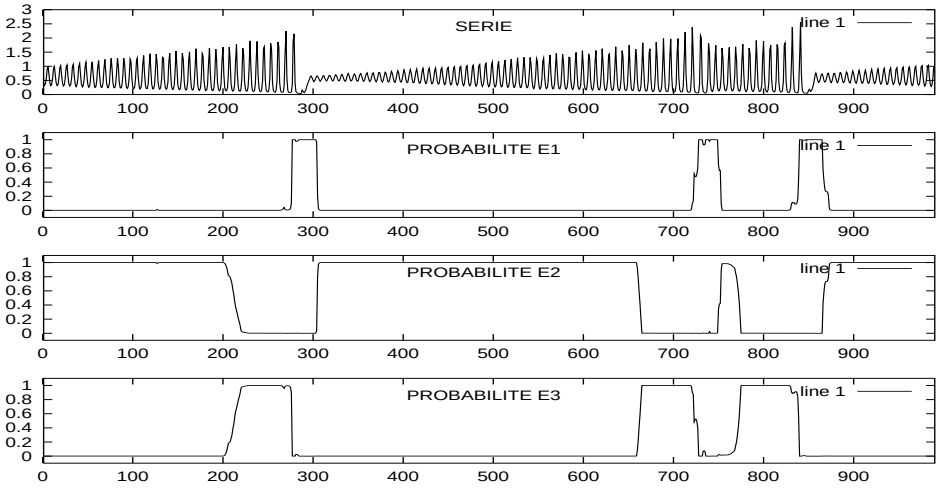
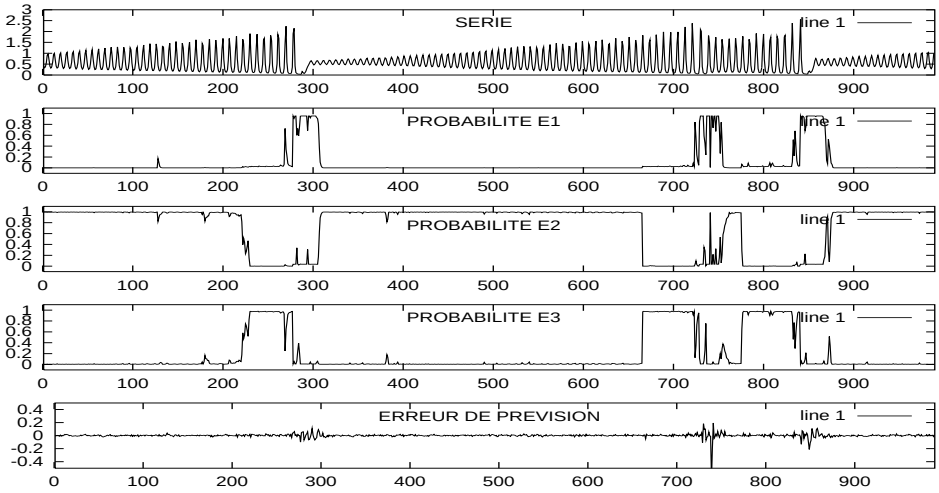


FIG. 5.5 – Estimation forward des états sur la série de validation



5.2.4 Conclusion

Si on compare avec les résultats obtenus avec le MLP "porte" de Weigend et al.[61] on obtient une E.N.M.S. comparable à leur modèle, mais avec beaucoup moins de paramètres. En effet la meilleure E.N.M.S. sur la base de validation est de 0.0035 dans leur cas avec 6 experts (de même architecture que pour notre modèle) et de 0.0033 avec 2 experts dans notre cas. De plus notre modèle fournit une bien meilleure segmentation de la série, car la segmentation obtenue avec un MLP "porte" oscille sans cesse d'un état à un autre (cf [61]), mais est très claire avec notre modèle.

On remarquera que la meilleure erreur de prévision est obtenue par le modèle avec deux experts, cependant le modèle avec trois experts donne quand même une segmentation intéressante avec l'apparition d'un état de pré-effondrement.

Enfin ce modèle semble prometteur non seulement pour la prévision classique de séries temporelles mais aussi pour prévoir des renversements de tendances (Krach financier, avalanche,...) car on peut en prédisant la probabilité des états à venir, avoir un aperçu du comportement futur de la série.

Chapitre 6

L'algorithme E.M. revisité

Pour trouver les paramètres qui maximisent la vraisemblance, on a recours à un algorithme E.M. Or pour calculer la pseudo-log-vraisemblance (E-step), on utilise généralement l'algorithme de Baum et Welch dit Forward-Backward. Cet algorithme a l'inconvénient de requérir la mémorisation de toutes les vraisemblances des observations, des estimateurs "Forward", et des estimateurs "Backward", ce qui demande beaucoup de ressources si la série étudiée est longue. De plus le passage Backward est contradictoire avec une implémentation en ligne de l'algorithme E.M. et les tentatives pour en implémenter une approximation doivent résoudre ce problème en faisant du "hors-ligne" local (voir par exemple Krishnamurthy [41] dans le cas de chaînes de Markov cachées dont l'espace des observations est continu, ce qui est un cas particulier des modèles étudiés dans cette thèse).

Cependant, Elliott [27] a montré dans le cas des chaînes de Markov cachées à observations continues que l'on pouvait exprimer les paramètres qui maximisent la pseudo-log-vraisemblance grâce à des estimateurs forward uniquement. Nous montrons comment on peut étendre les travaux d'Elliott aux modèles autorégressifs avec une innovation gaussienne, ce qui nous permet, dans le cas linéaire, de calculer sans passage Backward, ni mémorisation, les estimateurs qui maximisent la pseudo-log-vraisemblance. De plus, ces estimateurs étant récursifs, on peut alors proposer un schéma en ligne original pour l'algorithme E.M. appliqué à ce type de modèle.

Cette section s'organise en deux parties, après un rappel des hypothèses sur le modèle, on introduit la méthode utilisée pour calculer les estimateurs de façon récursive. Il s'agit en fait d'appliquer une version discrète du théorème de Girsanov pour pouvoir exprimer l'espérance conditionnelle des statistiques exhaustives utilisées pour le M-step de l'algorithme E.M. Ensuite, dans la deuxième partie, on proposera une version en ligne de cet algorithme et on en étudiera le comportement sur une simulation.

6.1 Introduction

Notre but est toujours de maximiser la vraisemblance grâce à l'algorithme E.M.. Cependant, on montre ici que l'on peut calculer directement par un algorithme forward les paramètres qui maximisent la pseudo-log-vraisemblance ce qui permet d'éviter le passage backward de l'algorithme de Baum et Welch. Ces estimateurs sont largement inspirés du chapitre 3 du livre d'Elliott [27].

Dans cette partie, pour que les estimateurs s'écrivent plus facilement on décalera les temps de 1 pour la chaîne de Markov cachée, cela ne change pas le modèle, il ne s'agit ici que d'un jeu d'écriture.

Le modèle considéré est donc le suivant :

$$\begin{cases} X_{t+1} = AX_t + V_{t+1} \\ Y_{t+1} = F_{X_t}(Y_{t-p+1}^t) + \sigma_{X_t}\varepsilon_{t+1} \end{cases}$$

où $(\varepsilon_t)_{t\in\mathbb{Z}}$ est une suite i.i.d gaussienne centrée réduite et $\sigma_{e_i} > 0$ pour tout $e_i \in \mathbb{E}$.

Le principal outil pour calculer ces nouveaux estimateurs est une version discrète du théorème de Girsanov. Ainsi on peut construire explicitement une probabilité $\bar{P}$ sous laquelle toutes les composantes du processus observé sont i.i.d. $\mathcal{N}(0,1)$.

En travaillant sous cette nouvelle mesure, les estimateurs récursifs du temps d'occupation d'un état de la chaîne de Markov cachée, du nombre de transitions d'un état vers l'autre, et des covariances de chaque modèle de régression sont facilement calculables. Il suffira alors de trouver une méthode pour revenir à la probabilité réelle pour avoir les estimateurs qui maximisent la pseudo-log-vraisemblance.

L'espace d'état de la chaîne (X_t) est toujours $\mathbb{E} = \{e_1, ..., e_N\}$, avec $e_i = (0, ..., 0, 1, 0, ..., 0)$ où seule la coordonnée i vaut 1. On suppose que X_0 est donné, ou que sa distribution ou moyenne $E[X_0]$ est connue.

Notation 11 *On notera* $\{\mathcal{F}_t\},\ t \in \mathbb{N}$ *la filtration engendrée par* $(X_l)_{1\leq l\leq t}$ *;* $\{\mathcal{Y}_t\},\ t \in \mathbb{N}$ *la filtration engendrée par* $(Y_l)_{1\leq l\leq t}$ *;* $\{\mathcal{G}_t\},\ t \in \mathbb{N}$ *la filtration engendrée par* $(Y_l, X_l)_{1\leq l\leq t}$

6.2 Changement de probabilité

Notons $\sigma_{e_i} = \sigma_i$, $\phi_i(x) = (2\pi\sigma_i^2)^{-1/2} exp\,(-x^2/2\sigma_i^2)$ pour la densité $\mathcal{N}(0,\sigma_i^2)$, $\phi(.)$ pour la densité de $\mathcal{N}(0,1)$ et $\langle .,. \rangle$ pour le produit scalaire dans $\mathbb{R}^N$. On rappelle d'abord le théorème de Bayes conditionnel dont on trouvera une démonstration dans le livre d'Elliott [27].

Théorème 12 *Soit* $(\Omega, \mathcal{F}, P)$ *un espace de probabilité, et* $\mathcal{G} \subset \mathcal{F}$ *une sous tribu. Soit* $\bar{P}$ *une autre mesure de probabilité absolument continue par rapport à* P*, la dérivée de*

Radon-Nikodym de $\bar{P}$ par rapport à P est $d\bar{P}/dP = \Lambda$. Alors si ϕ est une variable aléatoire $\bar{P}$ intégrable :

$$\bar{E}\left[\phi\,|\mathcal{G}\right] = \psi \;\; où \;\; \psi = \frac{E[\Lambda\phi|\mathcal{G}]}{E[\Lambda|\mathcal{G}]} \;\; si \;\; [\Lambda\,|\mathcal{G}] > 0$$
$$et \;\; \psi = 0 \;\; sinon$$

Maintenant, le processus observé $\{Y_t\},\ t \in \mathbb{N}$, est de la forme

$$Y_{t+1} = F_{X_t}(Y_{t-p+1}^t) + \sigma_{X_t} \times \varepsilon_{t+1}$$

où ε_t est i.i.d. $\mathcal{N}(0,1)$
on notera $\sigma_{X_t} = \langle \sigma, X_t \rangle$, avec σ le vecteur $(\sigma_{e_1}, ..., \sigma_{e_N})$.
Soit :

$$\lambda_l = \frac{\langle \sigma, X_{l-1}\rangle \phi(y_l)}{\phi(\varepsilon_l)}, \quad l \in \mathbb{N}^*,$$
$$\Lambda_0 = 1,$$

et

$$\Lambda_t = \prod_{l=1}^t \lambda_l, \quad t \geq 1.$$

Définissons une nouvelle mesure de probabilité $\bar{P}$ en établissant que sa dérivée de Radon-Nykodim par rapport à P restreinte à $\mathcal{G}_t$ est égale à Λ_t. L'existence d'une telle probabilité $\bar{P}$ est assuré par le théorème d'extension de Kolmogorov (cf [27]). On a alors le lemme suivant :

Lemme 5 *Sous $\bar{P}$, les (Y_t) sont des variables aléatoires indépendantes identiquement distribuées $\mathcal{N}(0,1)$*

Preuve $\bar{P}\left(Y_{t+1} \leq \tau\,|\mathcal{G}_t\right) = \bar{E}\left[1_{\{Y_{t+1}\leq\tau\}}\,|\mathcal{G}_t\right]$ grâce au théorème de Bayes conditionnel on a :

$$\bar{E}\left[1_{\{Y_{t+1}\leq\tau\}}\,|\mathcal{G}_t\right]$$

$$= \frac{E\left[\Lambda_{t+1} 1_{\{Y_{t+1}\leq\tau\}}\,|\mathcal{G}_t\right]}{E\left[\Lambda_{t+1}\,|\mathcal{G}_t\right]}$$

$$= \frac{\Lambda_t}{\Lambda_t} \times \frac{E\left[\lambda_{t+1} 1_{\{Y_{t+1}\leq\tau\}}\,|\mathcal{G}_t\right]}{E\left[\lambda_{t+1}\,|\mathcal{G}_t\right]}.$$

Maintenant

$$E\left[\lambda_{t+1}\,|\mathcal{G}_t\right] = \int_{-\infty}^{\infty} \frac{\langle \sigma, X_t\rangle\, \phi(Y_{t+1})}{\phi(\varepsilon_{t+1})} \times \phi(\varepsilon_{t+1}) d\varepsilon_{t+1}$$

$$= \int_{-\infty}^{\infty} \langle \sigma, X_t\rangle\, \phi(F_{X_t}(Y_{t-p+1}^t) + \langle \sigma, X_t\rangle \times \varepsilon_{t+1}) d\varepsilon_{t+1} = 1$$

d'où, en remarquant que $\varepsilon_{t+1} = \frac{Y_{t+1} - F_{X_t}\left(Y_{t-p+1}^t\right)}{\langle \sigma, X_t \rangle}$:

$$\begin{aligned} \bar{P}\left(Y_{t+1} \leq \tau \,|\mathcal{G}_t\right) &= E\left[\lambda_{t+1} 1_{\{Y_{t+1} \leq \tau\}} \,|\mathcal{G}_t\right] \\ &= \int_{-\infty}^{\infty} \frac{\langle \sigma, X_t \rangle \phi(Y_{t+1})}{\phi(\varepsilon_{t+1})} \times 1_{\{Y_{t+1} \leq \tau\}} \times \phi(\varepsilon_{t+1}) d\varepsilon_{t+1} \\ &= \int_{-\infty}^{\tau} \phi(Y_{t+1}) dy_{t+1} = \bar{P}\left(Y_{t+1} \leq \tau\right) \end{aligned}$$

■

Remarque 13 *On a*

$$\begin{aligned} &\bar{E}\left[X_{t+1} \,|\mathcal{G}_t\right] = E\left[\Lambda_{t+1} X_{t+1} \,|\mathcal{G}_t\right] / E\left[\Lambda_{t+1} \,|\mathcal{G}_t\right] \\ &= E\left[\lambda_{t+1} X_{t+1} \,|\mathcal{G}_t\right] = E\left[\lambda_{t+1}\left(AX_t + V_{t+1}\right) |\mathcal{G}_t\right] \\ &= AX_t E\left[\lambda_{t+1} \,|\mathcal{G}_t\right] + E\left[\lambda_{t+1} \,|\mathcal{G}_t\right] E\left[V_{t+1} \,|\mathcal{G}_t\right] = AX_t \end{aligned}$$

car λ_{t+1} et V_{t+1} sont indépendants et V_{t+1} est indépendant de $\mathcal{G}_t$.

Maintenant supposons que l'on débute sous une probabilité $\bar{P}$ sur $(\Omega, \mathcal{F})$. Cette probabilité est telle que sous $\bar{P}$:

1. $\{X_t\},\ t \in \mathbb{N}$, est une chaîne de Markov de matrice de transition A, ainsi

$$X_{t+1} = AX_t + V_{t+1}$$

   où $\bar{E}\left[V_{t+1} \,|\mathcal{F}_t\right] = 0$ pour tout $k \in \mathbb{N}$
2. $\{Y_t\},\ t \in \mathbb{N}$, est une suite i.i.d. $\mathcal{N}(0,1)$, en particulier indépendante de X_t

On souhaite alors construire une probabilité P telle que sous P :

$$\varepsilon_{t+1} = \frac{Y_{t+1} - F_{X_t}(Y_{t-p+1}^t)}{\langle \sigma, X_t \rangle},\ t \in \mathbb{N}$$

est une suite i.i.d. $\mathcal{N}(0,1)$, et donc $Y_{t+1} = F_{X_t}(Y_{t-1}^t) + \sigma_{X_t} \varepsilon_{t+1}$. Pour construire P à partir de $\bar{P}$ on introduit les inverses de λ_t et Λ_t.

Ecrivons :

$$\begin{aligned} &\bar{\lambda}_l = \lambda_l^{-1} = \tfrac{\phi(\varepsilon_l)}{\langle \sigma, X_{l-1} \rangle \phi(Y_l)}, \\ &\bar{\Lambda}_0 = 1 \end{aligned}$$

et

$$\bar{\Lambda}_t = \prod_{l=1}^{t} \bar{\lambda}_l,\ t \in \mathbb{N}^*.$$

Définissons P par $\left(dP/d\bar{P}\right)|_{\mathcal{G}_t} = \bar{\Lambda}_t$. Cette probabilité est bien définie, car on a supposé $\sigma_i > 0,\ i \in \{1, ..., N\}$, on a alors le lemme :

Lemme 6 *Sous P, $(\varepsilon_t)_{t \in \mathbb{N}}$ est une suite i.i.d. $\mathcal{N}(0,1)$.*

Preuve C'est la même preuve que le lemme précédent en remplaçant dans la démonstration $y_{t+1}, \varepsilon_{t+1}, \lambda_{t+1}, \Lambda_{t+1}$ par $\varepsilon_{t+1}, y_{t+1}, \bar{\lambda}_{t+1}, \bar{\Lambda}_{t+1}$.

6.2.1 Retour à la probabilité réelle

Notation 12 *Si* $(H_t),\ t \in \mathbb{N}$, *est une suite quelconque, adaptée à* $(\mathcal{G}_t)$, *on écrira :*

$$\gamma_t(H_t) = \bar{E}\left[\bar{\Lambda}_t H_t \,|\mathcal{Y}_t\right]$$

Ainsi $\gamma_t(H_t)$ est l'espérance conditionnelle, non normalisée, de H_t suivant $\mathcal{Y}_t$. En utilisant le théorème de Bayes conditionnel on a :

$$\hat{H}_t := E\left[H_t \,|\mathcal{Y}_t\right] = \frac{\bar{E}\left[\bar{\Lambda}_t H_t \,|\mathcal{Y}_t\right]}{\bar{E}\left[\bar{\Lambda}_t \,|\mathcal{Y}_t\right]} = \frac{\gamma_t(H_t)}{\gamma_t(1)}$$

Supposons maintenant que $(H_t),\ t \in \mathbb{N}$, est une suite scalaire, on a :

$$\Delta H_{t+1} = H_{t+1} - H_t,\ \ H_{t+1} = H_t + \Delta H_{t+1}$$

d'où

$$\gamma_{t+1}(H_{t+1}) = \bar{E}\left[\Lambda_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] + \bar{E}\left[\Lambda_{t+1}\Delta H_{t+1} \,|\mathcal{Y}_{t+1}\right].$$

Analysons le premier terme de droite :

$$\begin{aligned}
\bar{E}\left[\bar{\Lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] &= \bar{E}\left[\bar{\Lambda}_t \bar{\lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] \\
&= \bar{E}\left[\bar{\Lambda}_t H_t \frac{\phi\left(\frac{Y_{t+1} - F_{X_t}(Y_{t-p+1}^t)}{\langle \sigma, X_t\rangle}\right)}{\langle \sigma, X_t\rangle \phi(Y_{t+1})} \,|\mathcal{Y}_{t+1}\right]
\end{aligned}$$

Notation 13 *Ecrivons*

$$\Gamma^i\left(Y_{t+1}\right) = \frac{\phi\left(\frac{Y_{t+1} - F_{X_t}(Y_{t-p+1}^t)}{\langle \sigma, e_i\rangle}\right)}{\langle \sigma, e_i\rangle\, \phi\left(Y_{t+1}\right)}$$

Par définition $\sum_{i=1}^N \langle X_t, e_i\rangle = 1$, alors

$$\bar{E}\left[\bar{\Lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] = \sum_{i=1}^N \bar{E}\left[\bar{\Lambda}_t \left\langle X_t, \Gamma^i\left(Y_{t+1}\right)\right\rangle H_t \,|\mathcal{Y}_{t+1}\right]$$

$$= \sum_{i=1}^N \left\langle \bar{E}\left[\bar{\Lambda}_t X_t H_t \,|\mathcal{Y}_{t+1}\right], \Gamma^i\left(Y_{t+1}\right)\right\rangle$$

et comme sous $\bar{P}$ les Y_t sont i.i.d.

$$\bar{E}\left[\bar{\Lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] = \sum_{i=1}^N \left\langle \bar{E}\left[\bar{\Lambda}_t X_t H_t \,|\mathcal{Y}_t\right], \Gamma^i\left(Y_{t+1}\right)\right\rangle.$$

D'où le lemme :

Lemme 7 *Avec les notations précédentes*

$$\bar{E}\left[\bar{\Lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1}\right] = \sum_{i=1}^{N} \left\langle \gamma_t\left(H_t X_t\right), \Gamma^i\left(Y_{t+1}\right)\right\rangle .$$

De plus il découle de la linéarité du produit scalaire et de l'espérance conditionnelle que :

$$\gamma_t\left(H_t X_t X_t^T\right) = \sum_{i=1}^{N} \left\langle \gamma_t\left(H_t X_t\right), e_i\right\rangle e_i e_i^T.$$

Ainsi, l'estimation de $\gamma_{t+1}(H_{t+1})$ implique le calcul de $\gamma_t(H_t X_t)$. On va donc calculer récursivement $\gamma_t(H_t X_t)$. Ensuite pour exprimer $\gamma_t(H_t)$, il suffit de remarquer que :

$$\langle X_t, \bar{1}\rangle = \sum_{i=1}^{N} \langle X_t, e_i\rangle = 1$$

avec $\bar{1}$ le vecteur $(1, ..., 1)'$ de $\mathbb{R}^N$, on a alors :

$$\begin{aligned} \langle \gamma_t(H_t X_t), \bar{1}\rangle &= \gamma_t(\langle H_t X_t, \bar{1}\rangle) \\ &= \gamma_t(H_t \langle X_t, \bar{1}\rangle) = \gamma_t(H_t). \end{aligned}$$

Finalement, lorsque l'estimation non-normalisée $\gamma_t(H_t X_t)$ est connue, l'estimateur $\gamma_t(H_t)$ est obtenu en sommant ses composantes.

De plus en prenant $H_t = 1$, on obtient :

$$\begin{aligned} \gamma_t(1) &= \gamma_t(\langle X_t, \bar{1}\rangle) = \langle \gamma_t(X_t), \bar{1}\rangle \\ &= \bar{E}\left[\bar{\Lambda}_t \,|\mathcal{Y}_t\right]. \end{aligned}$$

Et par le théorème de Bayes conditionnel, on aura :

$$E\left[H_t \,|\mathcal{Y}_t\right] = \frac{\gamma_t(H_t)}{\gamma_t(1)}.$$

Maintenant, on écrit le théorème central de ce chapitre, qui a été obtenu par Elliott dans le cas des chaînes de Markov cachées avec un espace d'observation continue et qu'on généralise à notre cadre.

Théorème 13 *Soit H_t un processus scalaire $\mathcal{G}$-adapté de la forme : H_0 est $\mathcal{F}_0$ mesurable, $H_{t+1} = H_t + \alpha_{t+1} + \langle \beta_{t+1}, V_{t+1}\rangle + \delta_{t+1} f(Y_{t+1})$, $k \geq 0$, où $V_{t+1} = X_{t+1} - AX_t$, f est une fonction réelle, α, β, δ sont des processus $\mathcal{G}$ prévisibles (β est N-dimensionnel). Alors :*

$$\begin{aligned} \gamma_{t+1}(H_{t+1} X_{t+1}) &:= \gamma_{t+1,t+1}(H_{t+1}) \\ &= \sum_{i=1}^{N} \Big\{ \left\langle \gamma_t(H_t X_t), \Gamma^i(Y_{t+1})\right\rangle a_i \\ &\quad + \gamma_t\left(\alpha_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right) a_i \\ &\quad + \gamma_t\left(\delta_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right) f(y_{t+1}) a_i \\ &\quad + \left(diag(a_i) - a_i a_i^T\right) \gamma_t\left(\beta_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right) \Big\} \end{aligned}$$

avec $a_i := Ae_i$ et $diag(a_i)$ la matrice ayant pour diagonale le vecteur a_i.

Preuve du théorème On prouve d'abord deux résultats préliminaires.

Résultat 1

$$\begin{aligned}\bar{E}\left[V_{t+1}\left|\mathcal{Y}_{t+1}\right.\right] &= \bar{E}\left[\bar{E}\left[V_{t+1}\left|\mathcal{G}_t,\mathcal{Y}_{t+1}\right.\right]\left|\mathcal{Y}_{t+1}\right.\right] \\ &= \bar{E}\left[\bar{E}\left[V_{t+1}\left|\mathcal{G}_t\right.\right]\left|\mathcal{Y}_{t+1}\right.\right] = 0.\end{aligned} \tag{6.1}$$

Résultat 2 En notant $diag(z)$ la matrice diagonale ayant le vecteur z pour diagonale, on a :

$$X_{t+1}X_{t+1}^T = AX_t(AX_t)^T + AX_tV_{t+1}^T + V_{t+1}(AX_t)^T + V_{t+1}V_{t+1}^T.$$

Comme X_t est de la forme $(0,\cdots,0,1,0,\cdots,0)$ on a

$$X_{t+1}X_{t+1}^T = diag(X_{t+1}) = diag(AX_t) + diag(V_{t+1})$$

donc

$$V_{t+1}V_{t+1}^T = diag(AX_t) + diag(V_{t+1}) - A\ diag(X_t)\ A^T - AX_tV_{t+1}^T - V_{t+1}(AX_t)^T.$$

Finalement, on obtient le résultat

$$\begin{aligned}\langle V_{t+1}\rangle &:= E[V_{t+1}V_{t+1}^T\left|\mathcal{F}_t\right.] \\ &= E[V_{t+1}V_{t+1}^T\left|X_t\right.] \\ &= diag(AX_t) - A\ diag(X_t)\ A^T.\end{aligned} \tag{6.2}$$

Preuve principale On a

$$\begin{aligned}&\gamma_{t+1,t+1}(H_{t+1}) = \bar{E}\left[\bar{\Lambda}_{t+1}H_{t+1}X_{t+1}\left|\mathcal{Y}_{t+1}\right.\right] \\ &= \bar{E}\left[(AX_t+V_{t+1})\left(H_t+\alpha_{t+1}+<\beta_{t+1},V_{t+1}>+\delta_{t+1}f(y_{t+1})\right)\times\bar{\Lambda}_{t+1}\left|\mathcal{Y}_{t+1}\right.\right]\end{aligned}$$

donc grâce à l'équation (6.1),

$$\gamma_{t+1,t+1}(H_{t+1}) = \bar{E}\left[\left(\left(H_t+\alpha_{t+1}+\delta_{t+1}f(y_{t+1})\right)AX_t+<\beta_{t+1},V_{t+1}>\right)\times\bar{\Lambda}_{t+1}\left|\mathcal{Y}_{t+1}\right.\right].$$

Par conséquent :

$$\begin{aligned}&\gamma_{t+1,t+1}(H_{t+1}) = \sum_{j=1}^{N}\left\{\bar{E}\left[\left(\left(H_t+\alpha_{t+1}+\delta_{t+1}f(y_{t+1})\right)<AX_t,e_j>e_j\right)\bar{\Lambda}_{t+1}\left|\mathcal{Y}_{t+1}\right.\right]\right\} \\ &+\bar{E}\left[<\beta_{t+1},V_{t+1}>\times\bar{\Lambda}_{t+1}\left|\mathcal{Y}_{t+1}\right.\right],\end{aligned}$$

d'où :

$$\begin{aligned}&\gamma_{t+1,t+1}(H_{t+1}) = \sum_{j=1}^{N}\sum_{i=1}^{N}\left\{\bar{E}\left[\left(\left(H_t+\alpha_{t+1}+\delta_{t+1}f(y_{t+1})\right)<X_t,e_i>\right)\bar{\Lambda}_{t+1}a_{ji}e_j\left|\mathcal{Y}_{t+1}\right.\right]\right\} \\ &+\bar{E}\left[<\beta_{t+1},V_{t+1}>\times\bar{\Lambda}_{t+1}\left|\mathcal{Y}_{t+1}\right.\right].\end{aligned}$$

On a noté $a_i = Ae_i$, donc

$$\gamma_{t+1,t+1}(H_{t+1}) = \sum_{i=1}^{N} \left\{ \bar{E}\left[((H_t + \alpha_{t+1} + \delta_{t+1} f(y_{t+1})) < X_t, e_i >) \bar{\Lambda}_{t+1} a_i \,|\mathcal{Y}_{t+1} \right] \right\}$$
$$+ \bar{E}\left[< \beta_{t+1}, V_{t+1} > \times \bar{\Lambda}_{t+1} \,|\mathcal{Y}_{t+1} \right].$$

On a vu (lemme 7), que pour H_t une suite adaptée à $\mathcal{G}_t$

$$\bar{E}\left[\bar{\Lambda}_{t+1} H_t \,|\mathcal{Y}_{t+1} \right] = \sum_{i=1}^{N} \left\langle \gamma_t(H_t X_t), \Gamma^i(y_{t+1}) \right\rangle$$

d'où, pour tout $e_r \in \mathbb{E}$

$$\begin{array}{ll} \bar{E}\left[\bar{\Lambda}_{t+1} H_t < X_t, e_r > |\mathcal{Y}_{t+1} \right] & = \sum_{i=1}^{N} \left\langle \gamma_t(H_t X_t < X_t, e_r >), \Gamma^i(y_{t+1}) \right\rangle \\ & = \sum_{i=1}^{N} \left\langle \gamma_t(H_t X_t X_t^T e_r), \Gamma^i(y_{t+1}) \right\rangle \end{array}$$

et comme on a aussi la formule (lemme 7) :

$$\gamma_t(H_t X_t X_t^T) = \sum_{i=1}^{N} \left\langle \gamma_t(H_t X_t), e_i \right\rangle e_i e_i^T,$$

on aura :

$$\bar{E}\left[\bar{\Lambda}_{t+1} H_t < X_t, er > |\mathcal{Y}_{t+1} \right] = \sum_{i=1}^{N} \left\langle \gamma_t(H_t X_t X_t^T e_r), \Gamma^i(y_{t+1}) \right\rangle = \left\langle \gamma_t(H_t X_t), \Gamma^r(y_{t+1}) \right\rangle.$$

Comme α, β, δ sont $\mathcal{G}$ prévisibles, l'espérance conditionnelle linéaire et $f(y_{t+1})$ mesurable par rapport à $\mathcal{Y}_{t+1}$, le résultat préliminaire (6.2) nous permet de déduire le résultat annoncé ■

Ce résultat permet, en considérant des processus $(H_t)_{t\in\mathbb{N}}$ particuliers, d'obtenir récursivement des estimateurs pour un large type de processus. On pourra notamment calculer l'espérance conditionnelle du temps d'occupation d'un état, du nombre de passages d'un état à l'autre et des covariances des différents modèles de régression.

6.3 Application de ces estimateurs

Nous allons montrer comme appliquer ces équations aux modèles autorégressifs linéaires à changements de régime markoviens. Nous avons besoin pour cela du calcul de plusieurs quantités.

6.3.1 Estimateur de l'état

Dans le théorème 13 prenons $H_t = H_0 = 1$, $\alpha_t = 0$, $\beta_t = 0$, $\delta_t = 0$, alors :

$$\gamma_{t+1}(X_{t+1}) = \sum_{i=1}^{N} \left\langle \gamma_t(X_t), \Gamma^i(Y_{t+1}) \right\rangle a_i. \tag{6.3}$$

On obtient donc le vecteur $E\left[X_t \,|\mathcal{Y}_t\right] = \gamma_t(X_t)/\gamma_t(1)$

6.3.2 Estimateur du nombre de sauts de l'état r à l'état s

Le nombre de sauts de l'état e_r à l'état e_s au temps t est donné par :

$$\mathcal{J}_t^{rs} = \sum_{l=1}^{t} \left\langle X_{l-1}, e_r \right\rangle \left\langle X_l, e_s \right\rangle.$$

En utilisant $X_l = AX_{l-1} + V_l$ on a :

$$\begin{array}{rcl} \mathcal{J}_{t+1}^{rs} & = & \sum_{l=1}^{t+1} \left\langle X_{l-1}, e_r \right\rangle \left\langle X_l, e_s \right\rangle \\ & = & \mathcal{J}_t^{rs} + \left\langle X_t, e_r \right\rangle \left\langle X_{t+1}, e_s \right\rangle \\ & = & \mathcal{J}_t^{rs} + \left\langle X_t, e_r \right\rangle \left(\left\langle AX_t, e_s \right\rangle + \left\langle V_{t+1}, e_s \right\rangle \right) \\ & = & \mathcal{J}_t^{rs} + \left\langle X_t, e_r \right\rangle a_{sr} + \left\langle X_t, e_r \right\rangle \left\langle V_{t+1}, e_s \right\rangle. \end{array}$$

Ainsi, en appliquant le théorème 13, avec $H_{t+1} = \mathcal{J}_{t+1}^{rs}$, $H_0 = 0$, $\alpha_{t+1} = \langle X_t, e_r \rangle$, $\beta_{t+1} = \langle X_t, e_r \rangle e_s^T$ on obtient

$$\begin{array}{rcl} \gamma_{t+1,t+1}\left(\mathcal{J}_{t+1}^{rs}\right) = & \sum_{i=1}^{N} & \left\langle \gamma_{t,t}\left(\mathcal{J}_t^{rs}\right), \Gamma^i\left(Y_{t+1}\right) \right\rangle a_i \\ & + & \left\langle \gamma_t\left(X_t\right), \Gamma^r(Y_{t+1}) \right\rangle a_{sr} e_s. \end{array} \tag{6.4}$$

6.3.3 Estimateur du temps d'occupation

Ecrivons $\mathcal{O}_t^r$ pour le temps passé par X à l'état e_r, jusqu'au temps t, alors :

$$\begin{array}{rcl} \mathcal{O}_{t+1}^r & = & \sum_{n=1}^{t+1} \left\langle X_n, e_r \right\rangle \\ & = & \mathcal{O}_t^r + \left\langle X_t, e_r \right\rangle. \end{array}$$

En appliquant le théorème 13, avec $H_{t+1} = \mathcal{O}_{t+1}^r$, $H_0 = 0$, $\alpha_{t+1} = \langle X_t, e_r \rangle$, $\beta_{t+1} = 0$, et $\delta_{t+1} = 0$, alors :

$$\begin{array}{rcl} \gamma_{t+1,t+1}\left(\mathcal{O}_{t+1}^r\right) = & \sum_{i=1}^{N} & \left\langle \gamma_{t,t}\left(\mathcal{O}_t^r\right), \Gamma^i\left(Y_{t+1}\right) \right\rangle a_i \\ & + & \left\langle \gamma_t\left(X_t\right), \Gamma^r(Y_{t+1}) \right\rangle a_r. \end{array} \tag{6.5}$$

Remarque 14 *On peut maintenant, en accord avec l'algorithme E.M., trouver les nouveaux coefficients de la matrice de transition de la chaîne de Markov cachée*

$$\hat{a}_{sr} = \frac{\gamma_T\left(\mathcal{J}_T^{sr}\right)}{\gamma_T\left(\mathcal{O}_T^r\right)}.$$

6.3.4 Estimation des fonctions de régression

Dans le cas linéaire, ces estimateurs permettent de trouver les paramètres $\hat{\theta}_{e_i}$ qui maximisent la pseudo-log-vraisemblance de l'algorithme E.M. directement, sans avoir recours à un calcul backward comme dans l'algorithme de Baum et Welch. En effet si on note les modèles de régression :

$$Y_t = a_1^i Y_{t-1} + ... + a_p^i Y_{t-p} + a_0^i + \sigma_i \varepsilon_t,\ 1 \leq i \leq N.$$

Si on observe une série $(y_{-p+1}, \cdots, y_n)$, en notant $\psi^T(t) = (1, Y_{t-1}, ..., Y_{t-p})$, $\theta_{e_i} = (a_0^i, ..., a_p^i)$ on a l'estimateur classique des moindres carrés qui s'écrit :

$$\hat{\theta}_{e_i}(n) = \left[\sum_{t=1}^{n} \psi(t)\psi^T(t)\right]^{-1} \sum_{t=1}^{n} \psi(t) Y_t. \tag{6.6}$$

On a donc besoin d'estimer des processus de la forme :

1.
$$\mathcal{TA}_{t+1}^r(j) = \sum_{l=1}^{t+1} \langle X_l, e_r \rangle Y_{l-j} Y_{l+1}$$
pour tous les j allant de -1 à p et $1 \leq r \leq N$.
On prend $j = -1$ pour calculer le moment d'ordre 2 du processus.

2.
$$\mathcal{TB}_{t+1}^r(i,j) = \sum_{l=1}^{t+1} \langle X_l, e_r \rangle Y_{l-j} Y_{l-i}$$
pour tous les j, i allant de 0 à p et $1 \leq r \leq N$.

3.
$$\mathcal{TC}_{t+1}^r = \sum_{l=1}^{t+1} \langle X_l, e_r \rangle Y_{l+1}.$$

4.
$$\mathcal{TD}_{t+1}^r(j) = \sum_{l=1}^{t+1} \langle X_l, e_r \rangle Y_{l-j}$$
pour tous les j allant de 0 à p et $1 \leq r \leq N$.

En appliquant le théorème 13 avec $H_{t+1}(j) = \mathcal{TA}_{t+1}^r(j)$, $H_0 = 0$, $\alpha_{t+1} = 0$, $\beta_{t+1} = 0$, $\delta_{t+1} = \langle X_t, e_r \rangle Y_{t-j}$ et $f(Y_{t+1}) = Y_{t+1}$,si $j \neq -1$ ou $\delta_{t+1} = \langle X_t, e_r \rangle$ et$f(Y_{t+1}) = Y_{t+1}^2$ si $j = -1$, on aura

$$\begin{array}{rcl} \gamma_{t+1,t+1}\left(\mathcal{TA}_{t+1}^r(j)\right) = & \sum_{i=1}^{N} & \langle \gamma_{t,t}\left(\mathcal{TA}_t^r(j)\right), \Gamma^i(Y_{t+1}) \rangle a_i \\ & + & \langle \gamma_t(X_t), \Gamma^r(Y_{t+1}) \rangle Y_{t-j} Y_{t+1} a_r, \end{array} \tag{6.7}$$

où a_r est la r-ième colonne de A.

Ensuite avec

$H_{t+1}(j) = \mathcal{TB}^r_{t+1}(i,j)$, $H_0 = 0$, $\alpha_{t+1} = 0$, $\beta_{t+1} = 0$ et $\delta_{t+1} = \langle X_t, e_r\rangle Y_{t-j}Y_{t-i}$ et $f(Y_{t+1}) = 1$ on a :

$$\begin{array}{rcl}\gamma_{t+1,t+1}\left(\mathcal{TB}^r_{t+1}(i,j)\right) = & \sum_{i=1}^N & \langle\gamma_{t,t}\left(\mathcal{TB}^r_t(j)\right), \Gamma^i(Y_{t+1})\rangle\, a_i \\ & + & \langle\gamma_t(X_t), \Gamma^r(Y_{t+1})\rangle\, Y_{t-j}Y_{t-i}a_r.\end{array} \tag{6.8}$$

Puis, avec

$H_{t+1} = \mathcal{TC}^r_{t+1}$, $H_0 = 0$, $\alpha_{t+1} = 0$, $\beta_{t+1} = 0$ et $\delta_{t+1} = \langle X_t, e_r\rangle$ et $f(Y_{t+1}) = Y_{t+1}$ on a :

$$\begin{array}{rcl}\gamma_{t+1,t+1}\left(\mathcal{TC}^r_{t+1}\right) = & \sum_{i=1}^N & \langle\gamma_{t,t}\left(\mathcal{TC}^r_t(j)\right), \Gamma^i(Y_{t+1})\rangle\, a_i \\ & + & \langle\gamma_t(X_t), \Gamma^r(Y_{t+1})\rangle\, Y_{t+1}a_r.\end{array} \tag{6.9}$$

Enfin, avec

$H_{t+1}(j) = \mathcal{TD}^r_{t+1}(j)$, $H_0 = 0$, $\alpha_{t+1} = 0$, $\beta_{t+1} = 0$ et $\delta_{t+1} = \langle X_t, e_r\rangle Y_{t-j}$ et $f(Y_{t+1}) = 1$ on aura :

$$\begin{array}{rcl}\gamma_{t+1,t+1}\left(\mathcal{TD}^r_{t+1}(j)\right) = & \sum_{i=1}^N & \langle\gamma_{t,t}\left(\mathcal{TD}^r_t(j)\right), \Gamma^i(Y_{t+1})\rangle\, a_i \\ & + & \langle\gamma_t(X_t), \Gamma^r(Y_{t+1})\rangle\, Y_{t-j}a_r.\end{array} \tag{6.10}$$

On peut maintenant, déterminer les nouveaux paramètres des modèles $F_{\theta_{e_r}}$, $r \in \{1, ..., N\}$ qui maximisent la pseudo-log-vraisemblance de l'algorithme E.M. En effet, on aura l'estimateur $\gamma_n\left(\mathcal{TA}^r_n(j)\right)$ (resp. $\gamma_n\left(\mathcal{TB}^r_n(j)\right)$, $\gamma_n\left(\mathcal{TC}^r_n(j)\right)$ et $\gamma_n\left(\mathcal{TD}^r_n(j)\right)$) en sommant toutes les composantes de $\gamma_{n,n}\left(\mathcal{TA}^r_n(j)\right)$ (resp. $\gamma_{n,n}\left(\mathcal{TB}^r_n(j)\right)$, $\gamma_{n,n}\left(\mathcal{TC}^r_n(j)\right)$ et $\gamma_{n,n}\left(\mathcal{TD}^r_n(j)\right)$), puis par le théorème de Bayes conditionnel, on calcule l'estimation

$$\hat{\mathcal{T}}\mathcal{A}^r_n(j) = \gamma_n\left(\mathcal{TA}^r_n(j)\right)/\gamma_n(1).$$

Enfin, pour $1 \leq r \leq N$, en notant R^r la matrice symétrique ayant pour élément R^r_{ij}, avec :

$$R^r_{11} = 1, R^r_{1j} = R^r_{j1} = \hat{\mathcal{T}}\mathcal{D}^r(j),\ R_{ij} = \hat{\mathcal{T}}\mathcal{B}^r(i-1, j-1)$$

et

$$C^r = (\hat{\mathcal{T}}\mathcal{C}^r, (\hat{\mathcal{T}}\mathcal{A}^r(i))_{0\leq i\leq p})$$

on aura :

$$\hat{\theta}_r = {R^r}^{-1}C^r$$

et cela **sans passage backward**, contrairement à l'algorithme de Baum et Welch.

6.3.5 Calcul de la variance résiduelle des modèles

Les statistiques calculées précédemment sont exhaustives, elles vont nous permettre de déterminer directement les variances résiduelles $\hat{\sigma}_1, ..., \hat{\sigma}_N$ qui maximisent la pseudo-log-vraisemblance. En effet, pour $1 \leq r \leq N$, on a l'erreur quadratique totale du modèle $r := \chi^2(r)$ qui vaut pour θ_r fixé :

$$\chi^2(r) = \sum_1^n \left(y_t - \theta_r^T \psi(t)\right)^2 \times \langle X_{t-1}, e_r \rangle$$

d'où :

$$\chi^2(r) = \sum_1^n \langle X_{t-1}, e_r \rangle \left(y_t^2 + \theta_r^T \psi(t)\psi^T(t)\theta_r - 2y_t\theta_r^T\psi(t)\right).$$

Par conséquent :

$$\chi^2(r) = \sum_1^n \langle X_{t-1}, e_r \rangle y_t^2 + \sum_1^n \langle X_{t-1}, e_r \rangle \theta_r^T \psi(t)\psi^T(t)\theta_r - \sum_1^n \langle X_{t-1}, e_r \rangle 2y_t\theta_r^T\psi(t),$$

finalement par linéarité des produits matriciels et scalaires :

$$\chi^2(r) = \mathcal{TA}^r(-1) + \theta_r^T R^r \theta_r - 2\theta_r^T C^r.$$

Ainsi on aura l'estimateur de la variance résiduelle du modèle r :

$$\hat{\sigma}_r^2 = \frac{1}{\mathcal{O}_r}\left(\hat{\mathcal{T}}\mathcal{A}^r(-1) + \hat{\theta}_r^T R^r \hat{\theta}_r - 2\hat{\theta}_r^T C^r\right). \tag{6.11}$$

Calcul des probabilités conditionnelles des états Si on veut connaître la probabilité des états à chaque instant, on peut calculer les $\omega_m(e_i) = E\left[\langle X_m, e_i \rangle \,|\mathcal{Y}_n\right]$ pour $m \in \{1, ..., n\}$ et $i \in \{1, ..., N\}$. Pour cela on peut de la même manière que dans l'algorithme de Baum et Welch calculer des estimateurs backward.

Notons :

$$\bar{\Lambda}_m = \prod_{l=m}^n \bar{\gamma}_l$$

où

$$\bar{\gamma}_l = \frac{\phi(\varepsilon_l)}{\langle \sigma, X_{l-1} \rangle \, \phi(y_l)}$$

et

$$\beta_m(e_r) = \bar{E}\left[\bar{\Lambda}_{m+2} \,| X_m = e_r, \mathcal{Y}_n\right].$$

On montre (en adaptant l'exercice 3 de la section 2.11 de Elliott [27] à notre modèle) que β_m satisfait la récursion backward suivante :

$$\beta_m(e_r) = \sum_{l=1}^N \Gamma^i(Y_{m+2})\beta_{m+1}(e_l)a_{rl}$$

avec $\beta_n = \beta_{n-1} = 1$ et finalement :

$$\omega_m(e_r) = \frac{\langle \gamma_m(X_m), e_r \rangle \, \beta_m(e_r) \Gamma^r(Y_{m+1})}{\sum_{r=1}^{N} \langle \gamma_m(X_m), e_r \rangle \, \beta_m(e_r) \Gamma^r(Y_{m+1})}.$$

6.4 Algorithme E.M. en ligne

Puisque l'on a montré que l'on pouvait calculer les paramètres qui optimisent la pseudo-log-vraisemblance directement, il semble naturel d'essayer d'implémenter cet algorithme en-ligne, c'est-à-dire de remettre à jour les paramètres pour chaque observation. Il n'existe cependant pas de preuve de la convergence d'un tel algorithme. On étudiera donc empiriquement sur des simulations son comportement, ainsi qu'une technique possible pour l'accélérer, on comparera enfin son comportement avec l'algorithme E.M. hors-ligne.

6.4.1 Estimateurs récursifs en ligne

On suppose toujours que l'on part de la mesure de probabilité telle que la chaîne de Markov cachée soit homogène de matrice de transition A et que les observations y_t soient i.i.d. $\mathcal{N}(0,1)$. Supposons que l'on remette à jour les paramètres après chaque nouvelle observation. Ces paramètres seront alors une fonction du temps.

Notation 14 *On notera $\Theta_t, t \in \mathbb{N}$, la tribu engendrée par la suite de paramètres $\theta_t, 0 \leq t \leq t$. $\{\mathcal{G}_t\}$, $t \in \mathbb{N}$ est maintenant la filtration engendrée par $(Y_t, X_t, \theta_t)_{1 \leq t \leq t}$. De même, comme maintenant les paramètres dépendent du temps, on note*

$$A^t = \left(a_{ij}^t\right), \, F_{e_i}^t, \, \sigma_t$$

la matrice de transition, les fonctions de régression et les variances au temps t

En généralisant la méthode avec des paramètres constants on pose :

$$\begin{aligned} \bar{\lambda}_l &= \frac{\phi(Y_l - F_{X_{l-1}}^{l-1}(Y_{l-p}^{l-1}))}{\langle \sigma_{l-1}, X_{l-1} \rangle \phi(Y_l)}, \\ \bar{\Lambda}_0 &= 1 \end{aligned}$$

et

$$\bar{\Lambda}_t = \prod_{l=1}^{t} \bar{\lambda}_l, \;\; k \in \mathbb{N}^*.$$

Remarquons que ces formules dépendent maintenant de la valeur des paramètres au temps l.

Définissons P par $\left(dP/d\bar{P}\right)|_{\mathcal{G}} = \bar{\Lambda}_t$, avec la nouvelle tribu $\mathcal{G}$ qui comprend aussi la suite de paramètres.

On aura alors sous P, $(\varepsilon_t)_{t\in\mathbb{N}}$ qui est une suite i.i.d. $\mathcal{N}(0,1)$.

Notation 15 *Si (H_t), $t \in \mathbb{N}$, est une suite quelconque, adaptée à la nouvelle filtration $\mathcal{G}$, on écrit :*

$$\gamma_{t,t}(H_t) = \bar{E}\left[\bar{\Lambda}_t H_t X_t \,|\mathcal{Y}_t\right]$$

et

$$\Gamma_t^i\left(Y_{t+1}\right) = \frac{\phi\left(\frac{Y_{t+1}-F_{X_t}^t(Y_{t-p+1}^t)}{\langle\sigma_t,e_i\rangle}\right)}{\langle\sigma_t,e_i\rangle\,\phi\left(Y_{t+1}\right)}$$

Théorème 14 *Soit H_t un processus scalaire $\mathcal{G}$-adapté de la forme : H_0 est $\mathcal{F}_0$ mesurable, $H_{t+1} = H_t + \alpha_{t+1} + \langle\beta_{t+1}, V_{t+1}\rangle + \delta_{t+1} f(Y_{t+1})$, $k \geq 0$, où $V_{t+1} = X_{t+1} - A_t X_t$, f est une fonction réelle, α, β, δ sont des processus $\mathcal{G}$ prévisibles (β est N-dimensionnel). Alors :*

$$\begin{aligned}
\gamma_{t+1}(H_{t+1}X_{t+1}) &:= \gamma_{t+1,t+1}(H_{t+1}) \\
&= \sum_{i=1}^{N}\Big\{\left\langle\gamma_t(H_tX_t),\Gamma_t^i(Y_{t+1})\right\rangle a_i^t \\
&\quad +\gamma_t\left(\alpha_{t+1}\left\langle X_t,\Gamma_t^i(Y_{t+1})\right\rangle\right)a_i^t \\
&\quad +\gamma_t\left(\delta_{t+1}\left\langle X_t,\Gamma_t^i(Y_{t+1})\right\rangle\right)f_{\theta_t}(y_{t+1})a_i^t \\
&\quad +\left(diag(a_i^t)-a_i^t a_i^{tT}\right)\gamma_t\left(\beta_{t+1}\left\langle X_t,\Gamma_t^i(Y_{t+1})\right\rangle\right)\Big\}
\end{aligned}$$

avec $a_i^t = A^t e_i$ et $diag(a_i^t)$ la matrice ayant pour diagonale le vecteur a_i^t.

Ce théorème se démontre exactement comme le théorème 13.

On peut alors calculer de nouveaux estimateurs pour le temps d'occupation, les transitions, les covariances empiriques. Cependant les paramètres sont remis à jour à chaque observation, c'est-à-dire que, comme à l'instant "t" on a un estimateur $\gamma_{t,t}(H_t)$, on remet à jour les paramètres en posant :

$$\hat{a}_{sr}^t = \frac{\gamma_t\left(\mathcal{J}_t^{sr}\right)}{\gamma_t\left(\mathcal{O}_t^r\right)}$$

et, avec les mêmes notations que précédemment :

$$\hat{\theta}_r^t = R_t^{r\,-1} C_t^r.$$

Remarque 15 *Ce théorème traduit seulement le fait que l'on peut toujours construire une formule de retour à la probabilité réelle pour les processus (H_t). Cependant, il est important de noter que les paramètres sont maintenant une fonction du temps, on ne calcule donc plus les statistiques exhaustives de la pseudo-log-vraisemblance et on ne peut donc plus assurer à chaque instant la croissance de la vraisemblance. Comme cette croissance est déterminante dans la preuve de la convergence de l'algorithme E.M., la preuve de la convergence d'un tel algorithme E.M. en ligne reste un problème ouvert.*

L'estimateur de $(\sigma_i)_{1\leq i\leq N}$ est différent de celui du cas hors-ligne (formule (6.11)), car les paramètres ne sont plus constants. Néanmoins il est facile à estimer en considérant le processus :

$$\mathcal{TE}^r_{t+1}(j) = \sum_{l=1}^{k+1} \langle X_l, e_r\rangle \, (Y_{t+1} - \hat{Y}^r_{t+1})^2$$

qui vérifie les hypothèses du théorème 14.

6.4.2 Etude empirique de cet algorithme

Nous allons étudier, sur des simulations, le comportement de cet estimateur en ligne. On va, dans un premier temps, introduire le coefficient d'oubli exponentiel qui est utilisé dans [41] pour accélérer la convergence de l'algorithme E.M. en ligne.

6.4.2.1 Coefficient d'oubli exponentiel

On introduit un coefficient d'oubli exponentiel, car cette modification a deux avantages. D'abord on augmente ainsi l'importance relative de la mise à jour du paramètre au vu de la nouvelle observation. Ensuite les erreurs provenant des mises à jour alors que les paramètres étaient loin des vrais paramètres, sont gommées petit à petit.

Pour mettre en pratique cet oubli, il suffit de remarquer que toutes les formules se divisent en 2 parties :

1. Partie sur le passé du processus. C'est celle qui pour un processus $\gamma_{t,t}(H_t)$ s'écrit :

$$\sum_{i=1}^{N} \left\{ \left\langle \gamma_t(H_t X_t), \Gamma^i(Y_{t+1}) \right\rangle a_i \, .$$

2. Partie sur la nouvelle observation

$$\begin{array}{l} \gamma_t \left(\alpha_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right) a_i \\ +\gamma_t \left(\delta_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right) f_{\theta_t}(y_{t+1}) a_i \\ + \left(diag(a_i) - a_i a_i^{'}\right) \gamma_t \left(\beta_{t+1} \left\langle X_t, \Gamma^i(Y_{t+1})\right\rangle\right)\Big) \Big\} \, . \end{array}$$

L'idée est donc de multiplier la première partie des estimateurs par un coefficient d'oubli $0 < \rho < 1$. Pour que l'estimateur garde quand même une mémoire longue, on choisira en pratique $0.999 \leq \rho \leq 0.9999$.

Remarque 16 *On considère donc ici des processus $\bar{H}_t$ tels que :*

$$\bar{H}_{t+1} = \rho_t \times \bar{H}_t + \alpha_{t+1} + \langle \beta_{t+1}, V_{t+1} \rangle + \delta_{t+1} f(Y_{t+1})$$

où ρ_t est une valeur déterministe . Grâce à la linéarité de l'espérance conditionnelle, le théorème 14 s'applique aussi à ce genre de processus.

6.4.2.2 La simulation

On simule une série avec trois modèles de régression :

$$\begin{aligned} AR_1 &: y_{t+1} = -1 - 0.6y_t - 0.2y_{t-1} + \varepsilon_{t+1} \\ AR_2 &: y_{t+1} = 1 + 0.5y_t + 0.2y_{t-1} + 2\varepsilon_{t+1} \\ AR_3 &: y_{t+1} = 0.2y_t - 0.3y_{t-1} + \varepsilon_{t+1} \end{aligned}$$

avec ε_t, une loi $\mathcal{N}(0, 1)$.
La matrice de transition A pour la chaîne de Markov cachée est :

$$\begin{pmatrix} 0.8 & 0.05 & 0.05 \\ 0.1 & 0.9 & 0.1 \\ 0.1 & 0.05 & 0.85 \end{pmatrix}$$

On simule une série de longueur 100000, les 1000 premières observations de la série ainsi générée sont représentées figure 6.1 :

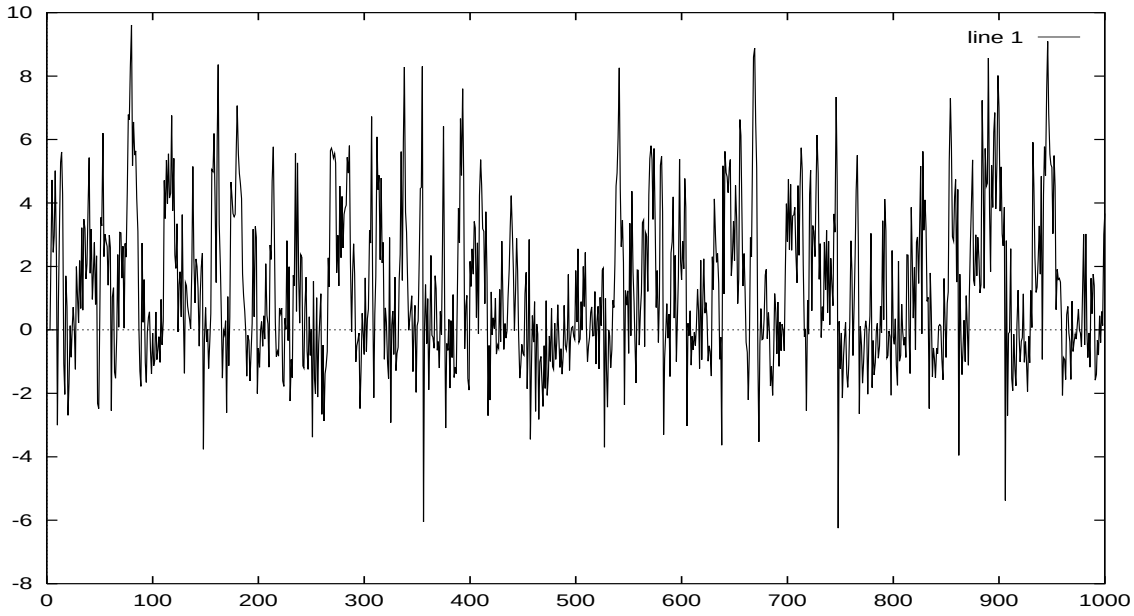
6.4.2.3 L'estimation suivant le coefficient d'oubli

Les paramètres de régression initiaux sont choisis au hasard, mais toujours les mêmes pour chaque estimation (c'est-à-dire pour chaque ρ). La matrice initiale A_0 a tous ses éléments égaux à 0.33, les variances initiales sont de 1.0 pour les trois modèles.

On montre l'évolution des termes diagonaux a_{ii} de la matrice de transition estimée ainsi que les paramètres des fonctions de régression des 3 modèles au cours de l'apprentissage. Les paramètres sont affichés toutes les 100 itérations. On étudie ainsi le comportement de l'estimateur E.M. récursifs suivant le coefficient d'oubli.

Estimation récursive sans oubli ($\rho = 1$) On peut remarquer que les paramètres sont mal estimés, pour l'estimateur sans oubli (figure 6.2), car leurs valeurs ont du mal à s'éloigner de la valeur d'initialisation.

FIG. 6.1 – 1000 premières observations de la série simulée



Estimation récursive avec un oubli fort ($\rho = 0.999$) L'algorithme converge vers les bons paramètres. Cependant, celui-ci est trop petit et les estimations des paramètres sont très dispersées. (cf figure 6.3)

Estimation récursive avec un oubli faible ($\rho = 0.9999$) On peut remarquer que les paramètres sont encore mal estimés, l'oubli n'est pas assez fort (cf figure 6.4). Il a un peu les mêmes défauts que l'estimateur sans oubli.

Estimation récursive avec un oubli moyen ($\rho = 0.9995$) Ici aussi l'algorithme converge vers les bons paramètres. Les estimateurs sont un peu dispersés (cf figure 6.5), mais le coefficient $\rho = 0.9995$ semble un bon compromis.

Estimation récursive avec un oubli évanescent On commence l'estimation avec un oubli important ($\rho = 0.999$), puis au cours de l'estimation on diminue l'oubli, c'est-à-dire qu'on augmente ρ jusqu'à 1 (plus d'oubli). La règle choisie est (It étant le nombre d'itérations) :

$$\rho = \inf(0.999 + 10^{-8} \times It., 1)$$

L'algorithme converge vers les bons paramètres, et les paramètres sont stables en fin d'estimation (cf figure 6.6). C'est le meilleur algorithme, mais il risque de ne pas être adapté pour un autre problème d'estimation

Algorithme E.M. hors-ligne A titre de comparaison, on montre la trajectoire des paramètres pour l'algorithme E.M. classique (cf figure 6.7). Il est à noter que les algorithmes récursifs convergent en un seul passage, alors qu'il faut plus de 30 évaluations de l'algorithme E.M. hors-ligne pour converger.

FIG. 6.2 – Estimateur sans oubli, coefficients des AR, et termes diagonaux de la matrice de transition.

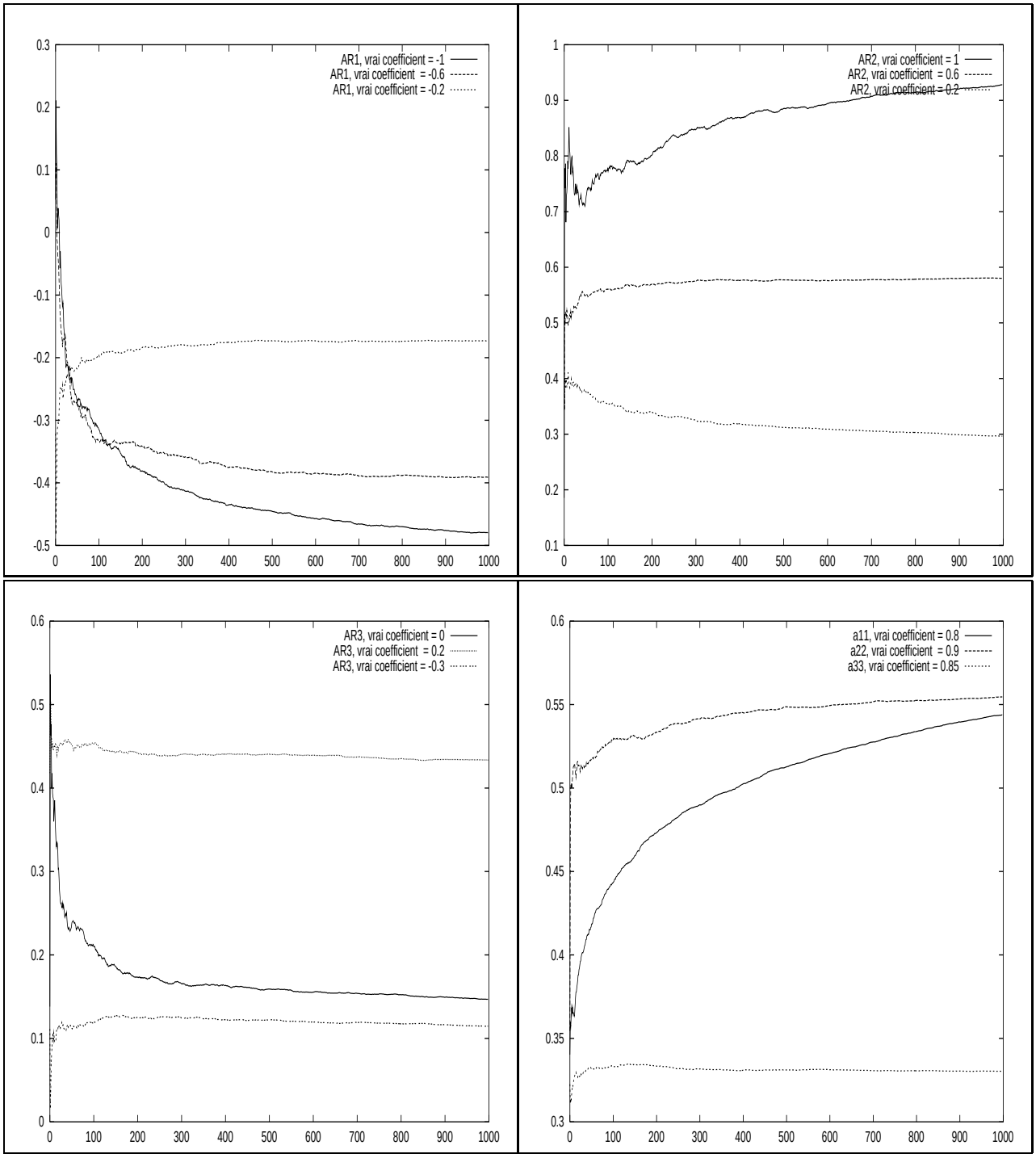


FIG. 6.3 – Estimateurs avec oubli ($\rho = 0.999$), coefficients des AR, et termes diagonaux de la matrice de transition.

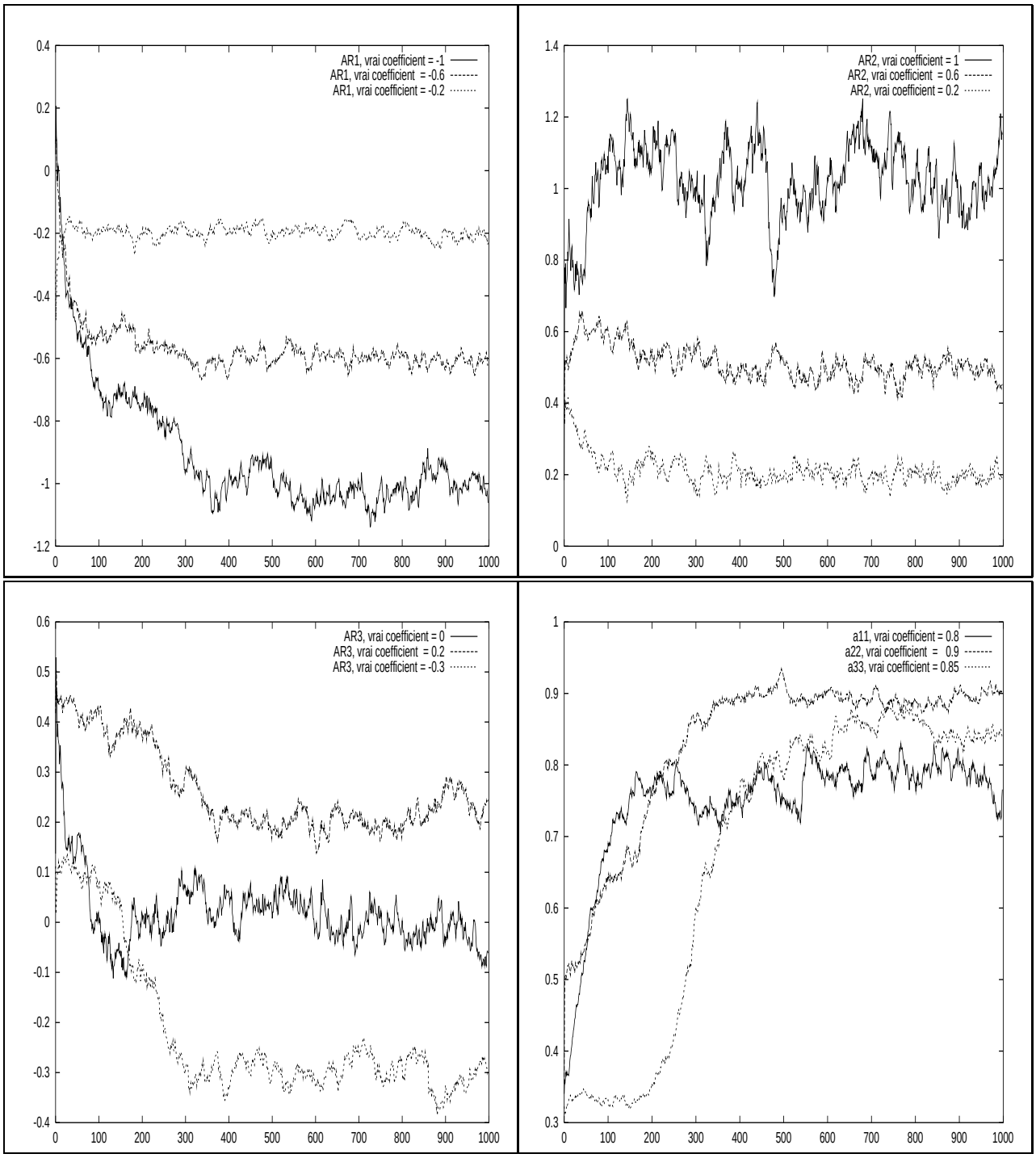


FIG. 6.4 – Estimateurs avec oubli ($\rho = 0.9999$), coefficients des AR, et termes diagonaux de la matrice de transition.

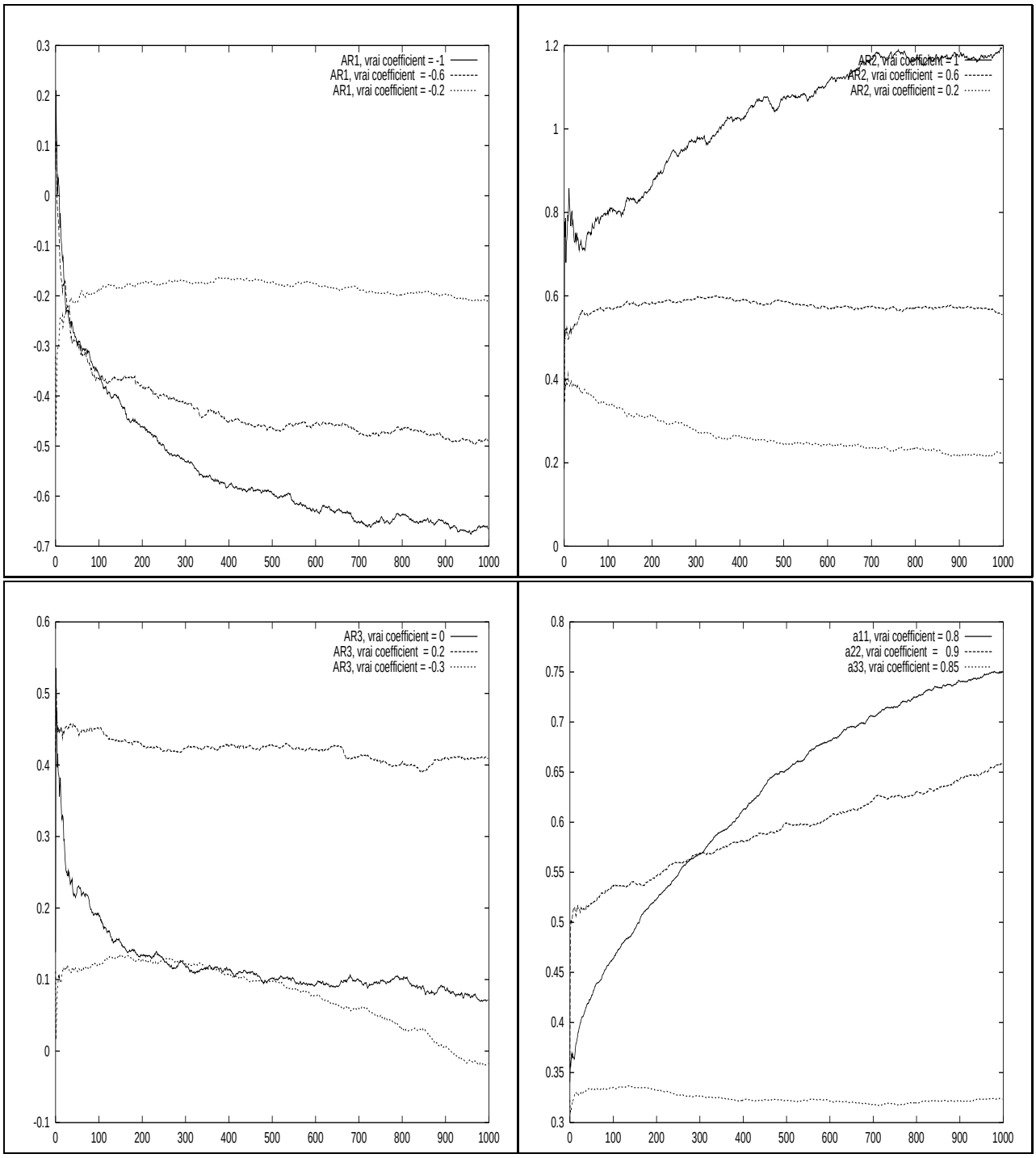


FIG. 6.5 – Estimateurs avec oubli ($\rho = 0.9995$), coefficients des AR, et termes diagonaux de la matrice de transition.

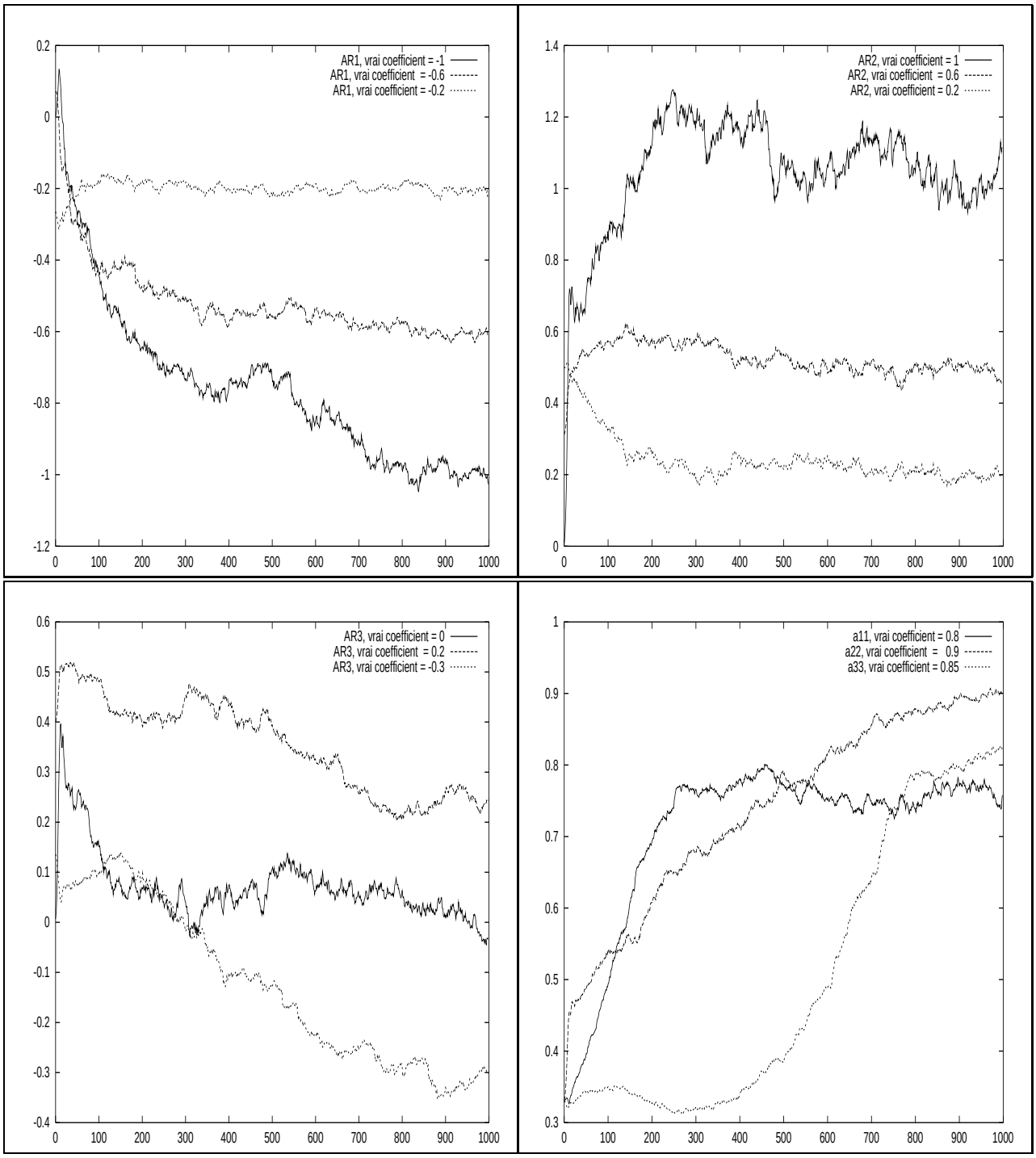


FIG. 6.6 – Estimateurs avec oubli lentement décroissant, coefficients des AR, et termes diagonaux de la matrice de transition.

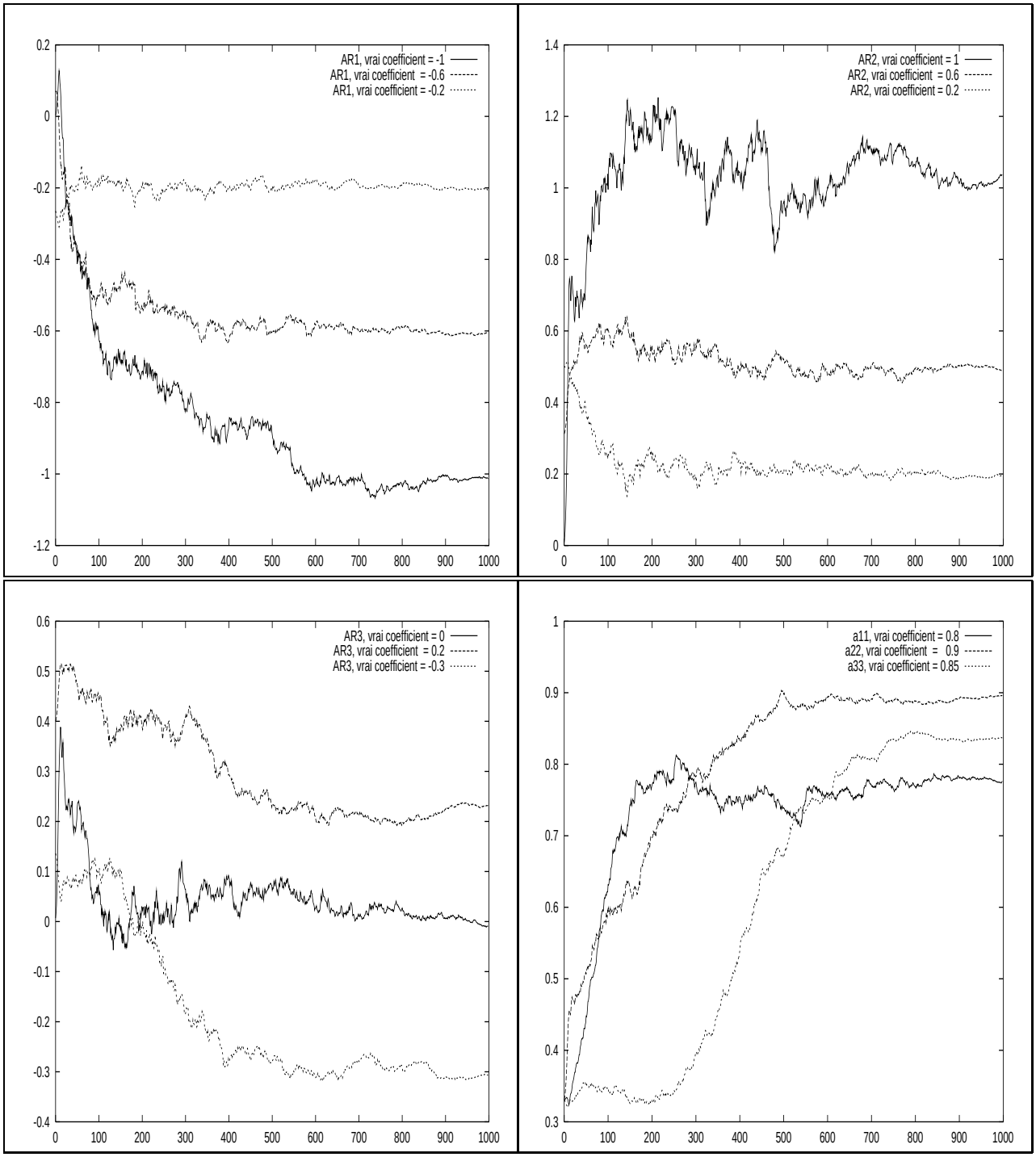
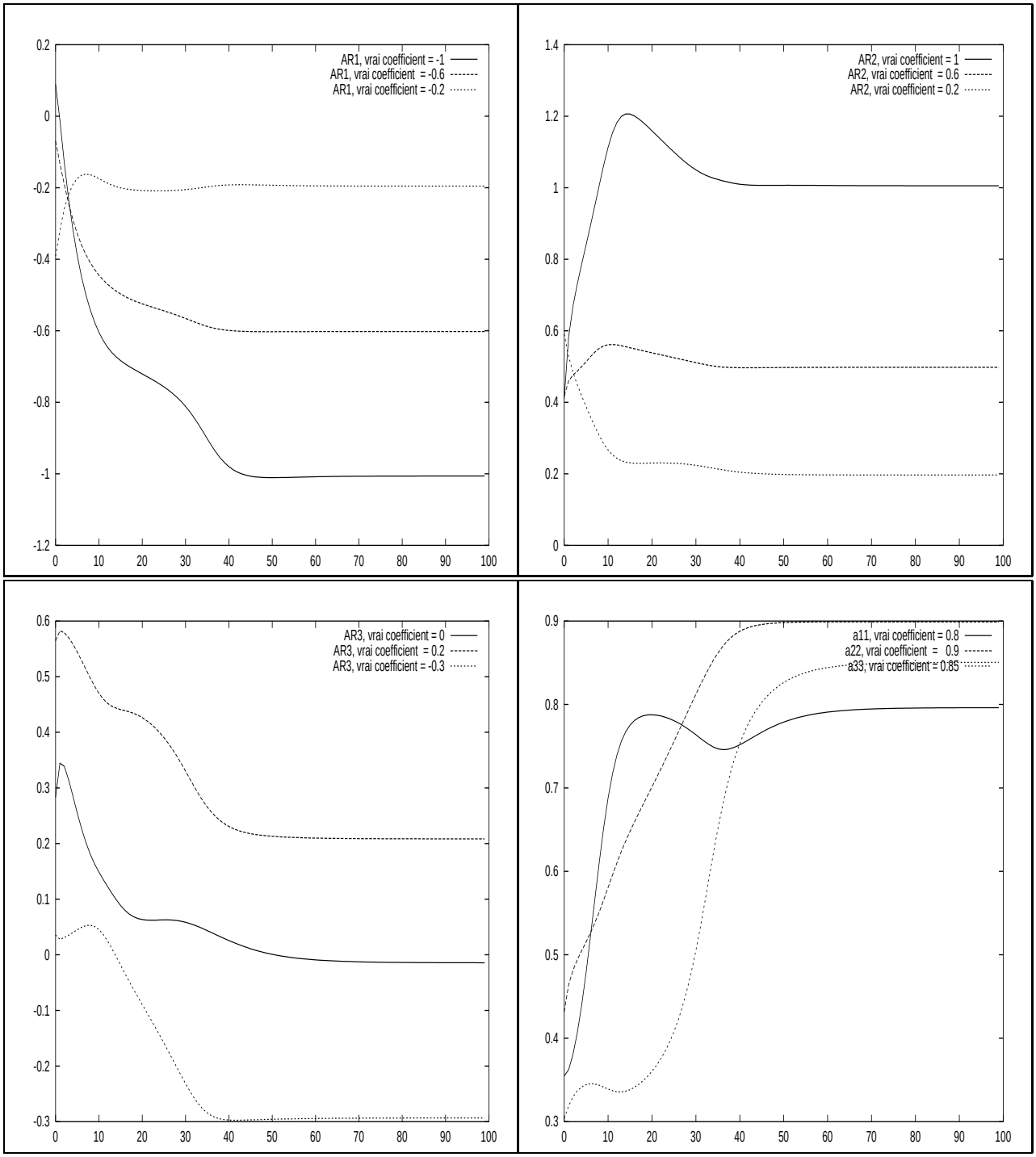


FIG. 6.7 – Estimateurs de l'E.M. hors ligne, coefficients des AR, et termes diagonaux de la matrice de transition.



6.5 Conclusion

Pour le type de modèles étudiés, on a montré que l'on pouvait obtenir des estimateurs forward du maximum de la pseudo-log-vraisemblance sans avoir recours à l'algorithme forward-backward de Baum et Welch. Ces estimateurs, bien que plus coûteux en temps de calcul, permettent d'éviter de mémoriser toutes les probabilités conditionnelles des états. Il peuvent de plus être utilisés pour d'autre quantités que les statistiques exhaustives du modèle, par exemple les probabilités conditionnelles des états dans une fenêtre de temps.

De plus, ces estimateurs ont permis de construire un schéma d'algorithme E.M. en ligne pour ces modèles. Cet algorithme, si on lui rajoute un coefficient d'oubli adéquat, semble avoir un comportement acceptable. Bien sûr, on ne peut pas être sûr que le coefficient d'oubli trouvé ici convienne à tous les problèmes, tout comme il n'existe pas de "pas" universel pour un gradient stochastique à pas constant. Cela donne quand même une idée de ce qu'il faudra utiliser dans la pratique. Le principal avantage de cette méthode récursive est le temps de calcul. En effet sur l'exemple étudié, celui-ci était réduit d'un facteur 10. Nous verrons cependant que nous pouvons encore améliorer le calcul de l'estimateur du maximum de vraisemblance au chapitre suivant.

Chapitre 7

Estimation directe des modèles autorégressifs à changements de régime markoviens

7.1 Introduction

Il s'agit d'étudier une méthode totalement différente de l'algorithme E.M., puisqu'on développe ici des estimateurs maximisant directement la log-vraisemblance par optimisation différentielle. On traite ici le cas de l'innovation gaussienne, mais cette étude est adaptable à d'autres densités pour le bruit. Nous nous placerons dans un cadre multidimensionnel qui est, par exemple, très utilisé en reconnaissance de la parole.

Après avoir introduit une nouvelle paramétrisation du modèle, nous calculerons la log-vraisemblance et sa dérivée. On obtiendra une forme additive de ces deux quantités et l'on en déduira une méthode récursive d'estimation du modèle. Nous montrerons enfin sur des simulations que dans certains cas, cet algorithme est bien plus performant que ceux du chapitre précédent.

7.2 Paramétrisation du modèle

7.2.1 Rappel des équations du modèle

Soit $(X_t)_{t\in\mathbb{N}}$ une chaîne de Markov à valeurs dans un espace d'état fini $E = \{e_1, ..., e_N\}$. Soit $(Y_t)_{t\in\mathbb{N}} \in \mathbb{R}^d$ la série des observations. on note toujours Y_{t-p+1}^t le vecteur $(Y_{t-p+1}, \cdots, Y_t)$. On considère le modèle suivant à un instant t fixé : $Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + M_{X_{t+1}}\varepsilon_{t+1}$ avec

1. $F_{e_i} \in \{F_{e_1}, ..., F_{e_N}\}$ une fonction paramétrique continûment différentiable de $\mathbb{R}^{d\times p} \longrightarrow \mathbb{R}^d$.

2. $M_{e_i} \in \{M_{e_1}, ..., M_{e_N}\}$ une matrice telle que $\Sigma_{e_i} = M_{e_i} M_{e_i}^T \in \mathbb{R}^{d \times d}$ soit définie positive.
3. (ε_t), $1 \leq t \leq T$ une suite i.i.d. gaussienne $\mathcal{N}(0, I_d)$ où I_d est la matrice identité de $\mathbb{R}^{d \times d}$
4. Sans perte de généralité, on identifie toujours l'espace d'état $E = \{e_1, \cdots, e_N\}$ avec le simplexe de $\mathbb{R}^N$ où e_i est un vecteur unité de $\mathbb{R}^N$ avec 1 sur la i-ème composante zéro partout ailleurs.

La chaîne X_t est caractérisée par sa matrice de transition $A = (a_{ij})$ qui est telle que :

$$P(X_{t+1} = e_i / X_t = e_j) = a_{ij}$$

En définissant : $V_{k+1} := X_{t+1} - AX_t$, on obtient les équations générales du modèle :

$$\begin{cases} X_{t+1} = AX_t + V_{t+1} \\ Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + M_{X_{t+1}} \varepsilon_{t+1} \end{cases}$$

A priori les paramètres du modèle sont donc

- Les coefficients (a_{ij}) de la matrice de transition A
- Les matrices de covariance Σ_{e_i}
- Les paramètres W_{e_i} des fonctions de régression F_{e_i}.

Cependant cette paramétrisation ne facilite pas les calculs, et on utilisera plutôt la paramétrisation décrite ci-après.

7.2.2 Paramétrisation de la matrice A

La matrice A est stochastique, c'est-à-dire que la somme d'une colonne quelconque de A est 1, on a donc $N - 1$ paramètres libres par colonne. Pour traduire cette contrainte on pose $\upsilon_{ij} = \ln \frac{a_{ij}}{a_{Nj}}$. On remarque alors que $\upsilon_{Nj} = 0$, et $(\upsilon_{1j}, \cdots, \upsilon_{N-1,j}) \in \mathbb{R}^{N-1}$, l'avantage de cette paramétrisation est de pouvoir optimiser la matrice A sans contrainte.

Expression de A en fonction de (υ_{ij}), $1 \leq i \leq N-1$, $1 \leq j \leq N$. Notons A_j la colonne j de A, alors on a :

$$A_j = \left(\frac{e^{\upsilon_{ij}}}{1 + e^{\upsilon_{1j}} + \cdots + e^{\upsilon_{N-1j}}} \right)_{1 \leq i \leq N}.$$

On en déduit une forme agréable pour dériver A par rapport aux paramètres (υ_{ij}) :

$$\frac{\partial a_{ij}}{\partial \upsilon_{ij}} = \frac{\partial}{\partial \upsilon_{ij}} \frac{e^{\upsilon_{ij}}}{1 + e^{\upsilon_{1j}} + \cdots + e^{\upsilon_{N-1j}}} = \frac{e^{\upsilon_{ij}}}{1 + e^{\upsilon_{1j}} + ... + e^{\upsilon_{N-1j}}} \left(1 - \frac{e^{\upsilon_{ij}}}{1 + e^{\upsilon_{1j}} + \cdots + e^{\upsilon_{N-1j}}}\right)$$

d'où

$$\frac{\partial a_{ij}}{\partial \upsilon_{ij}} = a_{ij}(1 - a_{ij}) \tag{7.1}$$

et pour $l \neq i$

$$\frac{\partial a_{ij}}{\partial \upsilon_{lj}} = \frac{e^{\upsilon_{ij}}}{1 + \cdots + e^{\upsilon_{N-1j}}} \times \left(-\frac{e^{\upsilon_{lj}}}{1 + \cdots + e^{\upsilon_{N-1j}}}\right) == -a_{ij}a_{lj}. \tag{7.2}$$

On a, par ailleurs, si $k \neq j$:

$$\frac{\partial a_{ij}}{\partial \upsilon_{lk}} = 0.$$

7.2.3 Paramétrisation des matrices de covariance du bruit

Puisque Σ_{e_i} est supposée définie positive, on choisit alors de prendre pour paramètres les termes de son inverse $\Sigma_{e_i}^{-1}$. De plus cette matrice étant symétrique, on ne prendra que les éléments sous la diagonale (diagonale incluse). On verra par la suite que cela simplifie l'expression de la dérivée de la vraisemblance par rapport aux paramètres.

7.2.4 Paramétrisation des fonctions de régression

On supposera seulement que la fonction F_{e_i} est continûment dérivable par rapport à chaque composante de son vecteur paramètre W_{e_i} .

7.2.5 Notation du vecteur paramètre du modèle

Le vecteur paramètre θ est donc (en le notant en ligne)

$$\theta = (W_{e_i}^T, \cdots, W_{e_N}^T, \upsilon_{11}, \cdots, \upsilon_{(N-1)N}, (\Sigma_{e_1}^{-1})_{11}, \cdots, (\Sigma_{e_1}^{-1})_{dd}, \cdots, (\Sigma_{e_N}^{-1})_{dd})$$

où $(\Sigma_{e_i}^{-1})_{lk}$ est le coefficient de la ligne l et de la colonne k $(k \leq l)$ de la matrice $\Sigma_{e_i}^{-1}$et $\omega_{e_i}^T$ représente les paramètres de la fonction F_{e_i} écrit en ligne.

7.3 Calcul de la log-vraisemblance et de sa dérivée

Afin de simplifier les notations, on notera $L(y_1, \cdots, y_n)$ la vraisemblance des observations $(y_1, \cdots, y_n)$ bien que celle-ci dépende du paramètre θ. De même la dépendance aux paramètres sera implicite dans l'expression des probabilités et des densités conditionnelles. Il serait possible de calculer la vraisemblance grâce à l'algorithme forward de Baum et Welch, néanmoins cette méthode requiert une stratégie de normalisation pour ne pas dépasser les capacités numériques de l'ordinateur, c'est pourquoi on utilise ici une forme plus agréable qui permet de calculer récursivement le *logarithme* de la vraisemblance par une méthode que l'on retrouve dans la thèse de Mevel [50] pour le cas de chaînes de Markov cachées à observations indépendantes.

7.3.1 La log-vraisemblance

On suppose les observations (y_{-p+1}, y_0) connues, la fonction de vraisemblance s'écrit :

$$L(y_1, ..., y_n) = \prod_{t=1}^{n} L(y_t \,|y_{-p+1}, \cdots, y_{t-1}) \; = L(y_n \,|y_{-p+1}, \cdots, y_{n-1}) \times \prod_{t=1}^{n-1} L(y_t \,|y_{-p+1}, \cdots, y_{t-1})$$

$$= \sum_{i=1}^{N} L(y_n \,|X_n = e_i, y_{-p+1}, \cdots, y_{n-1}) P(X_n = e_i \,|y_{-p+1}, \cdots, y_{n-1}) \times \prod_{t=1}^{n-1} L(y_t \,|y_{-p+1}, \cdots, y_{t-1}).$$

Si on note

- p_n le vecteur dont la i-ème composante est : $p_n(i) = P(X_n = e_i \,|y_{-p+1}, \cdots, y_{n-1})$
- b_n le vecteur dont la i-ème composante est :$b_n(i) = L(y_n \,|X_n = e_i, y_{-p+1}, \cdots, y_{n-1})$, c'est-à-dire la densité conditionnelle de y_n sachant $X_n = e_i$ et $(y_{-p+1}, \cdots, y_{n-1})$.
- $B_n = diag\,(b_n)$ la matrice diagonale ayant pour diagonale le vecteur b_n,

on aura :

$$L(y_1, \cdots, y_n) = b_n^T p_n \times \prod_{t=1}^{n-1} L(y_t \,|y_{-p+1}, \cdots, y_{t-1}) = \prod_{t=1}^{n} b_t^T p_t.$$

On en déduit une forme pratique de la log-vraisemblance :

$$\ln(L(y_1, \cdots, y_n)) = \sum_{t=1}^{n} \ln(b_t^T p_t). \tag{7.3}$$

Il suffit donc de calculer p_t pour $t = 1, \cdots, n$, pour pouvoir calculer la log-vraisemblance, car :

$$b_t(i) = L(y_t \,|X_t = e_i, y_{t-1}, \cdots, y_{-p+1}) := \Phi_{e_i}(y_t - F_{e_i}(y_{t-p+1}^{t-1}))$$

où

$$\Phi_{e_i}(y_t - F_{e_i}(y_{t-1})) = \frac{1}{\left(\sqrt{2\pi}\right)^d \sqrt{\det\left(\Sigma_{e_i}\right)}} \exp\left(-\frac{1}{2}\left((y_t - F_{e_i}(y_{t-p+1}^{t-1}))^T \Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p+1}^{t-1}))\right)\right)$$

est la densité conditionnelle de y_t sachant $X_t = e_i$.

Calcul de p_t Soit $p_t(i)$ la i-ème coordonnée ($1 \leq i \leq N$) du vecteur p_t . On a $p_t(i) = P(X_t = e_i \,|y_{-p+1}, \cdots, y_{t-1})$. Calculons $p_{t+1}(i)$:

$$p_{t+1}(i) = P(X_{t+1} = e_i \,|y_{-p+1}, \cdots, y_t) = \sum_{j=1}^{N} P(X_{t+1} = e_i, X_t = e_j \,|y_{-p+1}, \cdots, y_t)$$

$$= \sum_{j=1}^{N} P(X_{t+1} = e_i \,|X_t = e_j, y_{-p+1}, \cdots, y_t) \times P(X_t = e_j \,|y_{-p+1}, \cdots, y_t).$$

Mais comme (X_t) est une chaîne de Markov homogène :

$$P(X_{t+1} = e_i \,|X_t = e_j, y_{-p+1}, \cdots, y_t) = P(X_{t+1} = e_i \,|X_t = e_j) = a_{ij}$$

d'où

$$p_{t+1}(i) = \sum_{j=1}^{N} a_{ij} P(X_t = e_j \,|y_{-p+1}, \cdots, y_t).$$

De plus par la définition des densités conditionnelles :

$$P(X_t = e_j \,|y_{-p+1}, \cdots, y_t) = \frac{L(e_j, y_t \,|y_{-p+1}, \cdots, y_{t-1})}{L(y_t \,|y_{-p+1}, \cdots, y_{t-1})}$$

soit

$$P(X_t = e_j \,|y_{-p+1}, \cdots, y_t) = \frac{L(y_t \,|X_t = e_j, y_{-p+1}, \cdots, y_{t-1}) \times P(X_t = e_j \,|y_{-p+1}, \cdots, y_{t-1})}{L(y_t \,|y_{-p+1}, \cdots, y_{t-1})}$$

$$P(X_t = e_j \,|y_{-p+1}, \cdots, y_t) = \frac{b_t(j) \times p_t(j)}{b_t^T p_t}$$

donc

$$p_{t+1}(i) = \frac{\sum a_{ij} b_t(j) \times p_t(j)}{b_t^T p_t}$$

et finalement on en déduit :

$$p_{t+1} = \frac{A B_t p_t}{b_t^T p_t}. \tag{7.4}$$

On supposera que p_1 suit la distribution uniforme sur $\{1, \cdots, N\}$ et on pourra ainsi calculer p_t, $t = 1, \cdots n$ par récurrence. Nous verrons au chapitre suivant que le choix de la distribution initiale pour p_1 a une importance négligeable sur le calcul de la log-vraisemblance (grâce à la propriété d'oubli exponentiel de la distribution initiale).

7.3.2 Dérivée de la log-vraisemblance

On rappelle que l'on a

$$\ln(L(y_1, \cdots, y_n)) = \sum_{t=1}^{n} \ln(b_t^T p_t)$$

donc, si on note θ_j le j-ème paramètre du modèle, on a

$$\frac{\partial \ln(L(y_1, \cdots, y_n))}{\partial \theta_j} = \sum_{t=1}^{n} \frac{\frac{\partial b_t^T p_t}{\partial \theta_j}}{b_t^T p_t}.$$

Il suffit donc de calculer $\frac{\partial b_t^T p_t}{\partial \theta_j}$ pour pouvoir calculer la dérivée de la log-vraisemblance, en remarquant que :

$$\frac{\partial b_t^T p_t}{\partial \theta_j} = \frac{\partial b_t}{\partial \theta_j}^T p_t + b_t^T \frac{\partial p_t}{\partial \theta_j}. \tag{7.5}$$

7.3.2.1 Calcul de $\frac{\partial b_t}{\partial \theta_j}$ suivant θ_j

Comme $b_t(i)$ ne s'annule jamais, on peut utiliser la formule

$$\frac{\partial b_t(i)}{\partial \theta_j} = b_t(i) \times \frac{\partial \ln(b_t(i))}{\partial \theta_j},$$

la dérivée du logarithme de $b_t(i)$ étant plus simple à exprimer.

Si θ_j est un coefficient ν_{ij}, paramétrisant la matrice A : b_t étant indépendant de A on a

$$\frac{\partial b_t}{\partial \theta_j} = 0.$$

Si θ_j est un coefficient de la matrice de covariance inverse $\Sigma_{e_i}^{-1}$ Supposons que $\theta_j = (\Sigma_{e_i}^{-1})_{kl}$, alors :

$$\frac{\partial b_t}{\partial \theta_j} = u_i$$

où u_i est le vecteur de $\mathbb{R}^N$ dont tous les éléments sont nuls sauf la i-ème coordonnée qui vaut :

$$b_t(i) \times \frac{\partial[-\frac{1}{2}(d\ln(2\pi) - \ln(\det(\Sigma_{e_i}^{-1})) + Tr((y_t - F_{e_i}(y_{t-p}^{t-1}))^T \Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p}^{t-1}))]}{\partial \theta_j}.$$

Pour pouvoir calculer cette dérivée, on rappelle que l'on a les formules suivantes :

- Si A (de coefficients a_{ij}) est une matrice constante et X une matrice de coefficients x_{ij} :
$$\frac{\partial}{\partial x_{ij}} Tr(AX) = a_{ji}. \tag{7.6}$$
- En supposant maintenant X inversible et en notant x_{ij}^{-1} les coefficients de X^{-1} on a :
$$\frac{\partial}{\partial x_{ij}} \ln(\det(X)) = x_{ji}^{-1}. \tag{7.7}$$
- Si A, B, C sont trois matrices de tailles convenables, la trace de leur produit est invariante par permutation circulaire :
$$Tr(ABC) = Tr(BCA) = Tr(CAB). \tag{7.8}$$

On a, grâce aux formules (7.8) et (7.6) :

$$\frac{\partial(Tr((y_t - F_{e_i}(y_{t-p}^{t-1}))^T \Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p}^{t-1}))))}{\partial\theta_j} = \left((y_t - F_{e_i}(y_{t-p}^{t-1}))(y_t - F_{e_i}(y_{t-p}^{t-1}))^T\right)_{kl}$$

et grâce à la formule (7.7)

$$\frac{\partial \ln(\det(\Sigma_{e_i}^{-1}))}{\partial\theta_j} = (\Sigma_{e_i})_{kl}.$$

En résumé, le i-ème élément de u_i vaut (en utilisant la symétrie de Σ_{e_i}) :

$$\begin{array}{l} b_t(i) \times \left((\Sigma_{e_i})_{kl} - \left((y_t - F_{e_i}(y_{t-p}^{t-1}))(y_t - F_{e_i}(y_{t-p}^{t-1}))^T\right)_{kl}\right), \; si\; k \neq l \\ b_t(i) \times \frac{1}{2}\left((\Sigma_{e_i})_{kl} - \left((y_t - F_{e_i}(y_{t-p}^{t-1}))(y_t - F_{e_i}(y_{t-p}^{t-1}))^T\right)_{kl}\right), \; si\; k = l. \end{array} \tag{7.9}$$

Si θ_j est un des paramètres de la fonction de régression F_{e_i} Supposons que θ_j correspond à la composante l du vecteur paramètre W_i, on a :

$$\frac{\partial b_t}{\partial\theta_j} = (0, \cdots, \frac{\partial b_t(i)}{\partial\theta_j}, \cdots, 0)^T$$

où $\frac{\partial b_t(i)}{\partial\theta_j}$ vaut :

$$b_t(i) \times \frac{\partial - \frac{1}{2}[(d\ln(2\pi) - \ln(\det(\Sigma_{e_i}^{-1})) + Tr((y_t - F_{e_i}(y_{t-p}^{t-1}))^T \Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p}^{t-1}))]}{\partial\theta_j}$$

$$= -\frac{1}{2}\frac{\partial[Tr((y_t - F_{e_i}(y_{t-p}^{t-1}))^T \Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p}^{t-1}))]}{\partial\theta_j} \times b_t(i)$$

et en utilisant la formule (7.8)

$$\frac{\partial b_t(i)}{\partial \theta_j} = -\frac{1}{2}\frac{\partial [Tr(\Sigma_{e_i}^{-1}(y_t - F_{e_i}(y_{t-p}^{t-1})(y_t - F_{e_i}(y_{t-p}^{t-1}))^T]}{\partial \theta_j} \times b_t(i).$$

On note $\left(y_t - F_{e_i}(y_{t-p}^{t-1})\right)(l)$ (resp. $\left(F_{e_i}(y_{t-p}^{t-1})\right)(l)$) la l-ième composante du vecteur $\left(y_t - F_{e_i}(y_{t-p}^{t-1})\right)$ (resp. $\left(F_{e_i}(y_{t-p}^{t-1})\right)$). En utilisant la formule de dérivées des composées de fonctions et la formule (7.6), on a finalement :

$$\frac{\partial b_t(i)}{\partial \theta_j} =$$

$$\frac{1}{2}b_t(i) \times \sum_{1 \leq m,l \leq d} (\Sigma_{e_i}^{-1})_{lm} \left(\left(F_{e_i}(y_{t-p}^{t-1}) - y_t\right)(l) \frac{\partial F_{e_i}(y_{t-p}^{t-1})(m)}{\partial \theta_j} + \left(F_{e_i}(y_{t-p}^{t-1}) - y_t\right)(m) \frac{\partial F_{e_i}(y_{t-p}^{t-1})(l)}{\partial \theta_j} \right).$$

7.3.2.2 Calcul de $\frac{\partial p_t}{\partial \theta_j}$ suivant θ_j

On sait que p_t vérifie la récurrence (7.4) :

$$p_{t+1} = \frac{AB_t p_t}{b_t^T p_t}$$

En dérivant cette expression par rapport au paramètre θ_j, on a :

$$\frac{\partial p_{t+1}}{\partial \theta_j} = \frac{\partial AB_t p_t}{\partial \theta_j} \times \frac{1}{b_t^T p_t} + AB_t p_t \times \frac{\partial b_t^T p_t}{\partial \theta_j} \times \left(-\frac{1}{(b_t^T p_t)^2}\right)$$

soit

$$\frac{\partial p_{t+1}}{\partial \theta_j} = \left(\frac{\partial AB_t}{\partial \theta_j} p_t + AB_t \frac{\partial p_t}{\partial \theta_j}\right) \times \frac{1}{b_t^T p_t} + AB_t p_t \times \left(\frac{\partial b_t^T}{\partial \theta_j} p_t + b_t^T \frac{\partial p_t}{\partial \theta_j}\right) \times \left(-\frac{1}{(b_t^T p_t)^2}\right).$$

On a alors :

$$\frac{\partial p_{t+1}}{\partial \theta_j} = \frac{AB_t}{b_t^T p_t}\left[I - \frac{p_t b_t^T}{b_t^T p_t}\right]\frac{\partial p_t}{\partial \theta_j} + \left(\frac{\partial AB_t}{\partial \theta_j}\right)\frac{p_t}{b_t^T p_t} - \frac{AB_t p_t}{(b_t^T p_t)^2}\left(\frac{\partial b_t^T}{\partial \theta_j} p_t\right)$$

d'où :

$$\frac{\partial p_{t+1}}{\partial \theta_j} = \frac{AB_t}{b_t^T p_t}\left[I - \frac{p_t b_t^T}{b_t^T p_t}\right]\frac{\partial p_t}{\partial \theta_j} + \left(\frac{\partial A}{\partial \theta_j} B_t + A\frac{\partial B_t}{\partial \theta_j}\right)\frac{p_t}{b_t^T p_t} - \frac{AB_t p_t}{(b_t^T p_t)^2}\left(\frac{\partial b_t^T}{\partial \theta_j} p_t\right) \quad (7.10)$$

avec, si p_1 est la distribution initiale : $\frac{\partial p_1}{\partial \theta_j} = 0$ pour tout j.

7.3.2.3 Calcul de $\frac{\partial A}{\partial \theta_j}$

Si θ_j est un coefficient paramétrisant la matrice A, avec $\theta_j = \upsilon_{lm}$, on a

$$\frac{\partial A}{\partial \theta_j} = C(\upsilon_{lm})$$

où $C(\upsilon_{lm})$ est une matrice dont tous les coefficients sont nuls, sauf la colonne m : C_m qui est telle que :

$$\begin{cases} C_m(i) = -a_{im}a_{lm} \text{ si } i \neq l \\ C_m(i) = a_{lm}(1 - a_{lm}) \text{ si } i = l \end{cases} . \tag{7.11}$$

7.3.2.4 Calcul de $\frac{\partial B_t}{\partial \theta_j}$

- Si θ_j est un coefficient de la matrice de la matrice $\Sigma_{e_i}^{-1}$ ou de la fonction de régression F_{e_i}, on déduit facilement $\frac{\partial B_t}{\partial \theta_j}$ de $\frac{\partial b_t(i)}{\partial \theta_j}$, puisque c'est la matrice :

$$diag(0, ..., \frac{\partial b_t(i)}{\partial \theta_j}, ..., 0).$$

- Sinon $\frac{\partial B_t}{\partial \theta_j}$ est la matrice nulle.

7.3.2.5 Résumé

Avec les notations précédentes, pour estimer les paramètres de l'estimateur du maximum de vraisemblance du modèle, on doit maximiser :

$$\ln(L(y_1, \cdots, y_n)) = \sum_{t=1}^{n} \ln(b_t^T p_t)$$

Ce qui s'obtient par une méthode d'optimisation différentielle classique, car le gradient se calcule ainsi :

$$\frac{\partial \ln(L(y_1, \cdots, y_n))}{\partial \theta_j} = \sum_{t=1}^{n} \frac{\frac{\partial b_t^T p_t}{\partial \theta_j}}{b_t^T p_t}$$

et $\frac{\partial b_t^T p_t}{\partial \theta_j}$, se calcule récursivement grâce aux formules :

$$\frac{\partial b_t^T p_t}{\partial \theta_j} = \frac{\partial b_t}{\partial \theta_j}^T p_t + b_t^T \frac{\partial p_t}{\partial \theta_j}$$

avec

$$\begin{cases} \frac{\partial b_t}{\partial \theta_j} = 0 \text{ si } \theta_j = \nu_{ij},\ 1 \leq i \leq N-1,\ 1 \leq j \leq N \\ \frac{\partial b_t}{\partial \theta_j} = b_t(i) \times \frac{1}{2}\left((\Sigma_{e_i})_{ll} - \left((y_t - F_{e_i}(y_{t-p}^{t-1}))(y_t - F_{e_i}(y_{t-p}^{t-1}))^T\right)_{ll}\right) \text{ si } \theta_j = \left(\Sigma_{e_i}^{-1}\right)_{ll} \\ \frac{\partial b_t}{\partial \theta_j} = b_t(i) \times \left((\Sigma_{e_i})_{lm} - \left((y_t - F_{e_i}(y_{t-p}^{t-1}))(y_t - F_{e_i}(y_{t-p}^{t-1}))^T\right)_{lm}\right) \text{ si } \theta_j = \left(\Sigma_{e_i}^{-1}\right)_{l \neq m} \\ \frac{\partial b_t}{\partial \theta_j} = \frac{1}{2} b_t(i) \times \sum_{m,l} (\Sigma_{e_i}^{-1})_{lm} \left(\left(F_{e_i}(y_{t-p}^{t-1}) - y_t\right)(l) \frac{\partial F_{e_i}(y_{t-p}^{t-1})(m)}{\partial \theta_j} \right. \\ \left. + \left(F_{e_i}(y_{t-p}^{t-1}) - y_t\right)(m) \frac{\partial F_{e_i}(y_{t-p}^{t-1})(l)}{\partial \theta_j} \right) \text{ si } \theta_j \text{ est un coefficient de } W_i \end{cases}$$

et $\frac{\partial p_t}{\partial \theta_j}$ qui vérifie la récurrence :

$$\frac{\partial p_{t+1}}{\partial \theta_j} = \frac{AB_t}{b_t^T p_t}\left[I - \frac{p_t b_t^T}{b_t^T p_t}\right]\frac{\partial p_t}{\partial \theta_j} + \left(\frac{\partial A}{\partial \theta_j}B_t + A\frac{\partial B_t}{\partial \theta_j}\right)\frac{p_t}{b_t^T p_t} - \frac{AB_t p_t}{(b_t^T p_t)^2}\left(\frac{\partial b_t^T}{\partial \theta_j}p_t\right).$$

De plus si p_1 est la distribution initiale : $\frac{\partial p_1}{\partial \theta_j} = 0$ pour tout j .

et

$$\begin{cases} \frac{\partial B_t}{\partial \theta_j} &= diag(0, \cdots, \frac{\partial b_t(i)}{\partial \theta_j}, \cdots, 0) & \text{si } \theta_j \in F_{e_i} \text{ ou } \Sigma_{e_i}^{-1} \\ \frac{\partial B_t}{\partial \theta_j} &= 0 & \text{sinon} \end{cases}$$

$$\begin{cases} \frac{\partial A}{\partial \theta_j} &= C(\upsilon_{lm}) & \text{si } \theta_j \in A,\ \theta_j = \upsilon_{lm} \\ \frac{\partial A}{\partial \theta_j} &= O_{N\times N} & \text{sinon} \end{cases},$$

enfin $C(\upsilon_{lm})$ définie par (7.11).

7.4 Application : Estimation récursive

7.4.1 Estimation récursive du maximum de vraisemblance

Un estimateur récursif θ_{n+1} du vecteur paramètre θ basé sur les $n+1$ premières observations de $(y_t)_{t\in\mathbb{N}^*}$ est de la forme :

$$\theta_{n+1} = \theta_n + \gamma_n H_n h(y_{n+1}, \theta_n)$$

où $h(y,\theta)$ est la fonction score, H_n une matrice adaptative et γ_n est une suite de gains satisfaisant

$$\gamma_n \leq 0,\ \sum_{n=1}^{\infty}\gamma_n = \infty,\ \sum_{n=1}^{\infty}\gamma_n^2 < \infty. \tag{7.12}$$

Pour des observations indépendantes avec une densité $f(y,\theta)$, la fonction score est

$$h(y,\theta) = \left\{ \frac{\partial \ln\left(f(y,\theta)\right)}{\partial \theta_i},\ 1 \leq i \leq dimension\, de\, \theta \right\}$$

et le choix optimal pour H_n est l'inverse de la matrice d'information , i.e. $H_n^{-1} = I(\theta_n)$, où

$$I\left(\theta\right) = E\left[h(y,\theta)h(y,\theta)^T\right].$$

Le calcul de cette matrice d' information requiert une intégration numérique, ce qui est très coûteux en temps de calcul. On utilisera donc à la place une estimation de cette matrice, i. e.

$$H_n^{-1} = \frac{1}{n}\sum_{t=1}^{n} h(y_t,\theta_{t-1})h(y_t,\theta_{t-1})^T.$$

La matrice H_n peut être estimée récursivement, grâce au lemme d'inversion de matrice de Ricatti (en notant $h_n = h(y_n,\theta_{n-1})$) :

$$H_n = \frac{1}{1-\gamma_n}\left(H_n - \frac{\gamma_n H_{n-1} h_n h_n^T H_{n-1}}{(1-\gamma_n) + \gamma_n h_n^T H_n h_n}\right).$$

7.4.2 Estimation récursive du maximum de vraisemblance pour un modèle autorégressif à changements de régime markoviens

Dans notre cas, les observations ne sont pas indépendantes identiquement distribuées, cependant la log-vraisemblance (7.3) a une forme additive comme pour le cas i.i.d. En suivant une démarche similaire[1] à Holst et al. [35], on déduit en l'algorithme récursif pour notre cas.

Notons

$$\theta_n = (\omega_{e_i}^{n^T}, \cdots, \omega_{e_N}^{n^T}, \upsilon_{11}^n, \cdots, \upsilon_{(N-1)N}^n, (\Sigma_{e_1}^{n^{-1}})_{11}, \cdots, (\Sigma_{e_1}^{n^{-1}})_{dd}, \cdots, (\Sigma_{e_N}^{n^{-1}})_{dd})^T$$

le paramètre au temps n, et A_n la matrice associée avec $\left(\upsilon_{11}^n, \cdots, \upsilon_{(N-1)N}^n\right)$, on a

$$\begin{cases} \theta_{n+1} = \theta_n + \gamma_n H_n h_{n+1} \\ H_n = \frac{1}{1-\gamma_n}\left(H_n - \frac{\gamma_n H_{n-1} h_n h_n^T H_{n-1}}{(1-\gamma_n)+\gamma_n h_n^T H_n h_n}\right) \end{cases}$$

où h_n est le vecteur gradient tel que la j-ème coordonnées soit :

$$h_n(j) = \frac{\partial {b_n}^T}{\partial \theta_n^j}\, p_n + b_n^T \frac{\partial p_n}{\partial \theta_n^j}$$

[1] Bien que les algorithmes semblent proches, nous cherchons à maximiser la log-vraisemblance, alors que Holst et al. cherchent à maximiser la pseudo-log-vraisemblance définie section 5.2.2.2.

où

$$p_{n+1} = \frac{A_n B_n p_n}{b_n^T p_n}$$

et

$$\frac{\partial p_{n+1}}{\partial \theta_n^j} = \frac{A_n B_n}{b_n^T p_n}\left[I - \frac{p_n b_n^T}{b_n^T p_n}\right]\frac{\partial p_n}{\partial \theta_n^j} + \left(\frac{\partial A_n}{\partial \theta_n^j}B_n + A_n\frac{\partial B_n}{\partial \theta_n^j}\right)\frac{p_n}{b_n^T p_n} - \frac{A_n B_n p_n}{(b_n^T p_n)^2}\left(\frac{\partial b_n^T}{\partial \theta_n^j}p_n\right).$$

Les conditions pour la consistance et la normalité asymptotique de ces procédures sont en général des questions ouvertes même dans le cas i.i.d. Dans cet article le modèle est très général et il n'existe pas de résultats théoriques. Nous verrons cependant que cet estimateur semble très bien se comporter sur des données simulées. Dans la suite, la valeur initiale du pas est $\gamma_0 = 0.08$, il décroît à la vitesse $\frac{1}{n^{1/2+1e-16}}$.

7.5 Performance des algorithmes sur des simulations

Nous montrons ici le bon comportement des algorithmes étudiés, que les fonctions de régression soient non-linéaires (MLP) ou linéaires. Nous indiquons de plus les temps de calculs obtenus sur un PC de bureau (400Mhz).

7.5.1 Simulation avec deux MLP pour fonctions de régression

7.5.1.1 Le modèle

On simule une série avec deux MLP qui ont 2 entrées, une couche cachée de 3 unités (avec des tangentes hyperboliques pour fonctions d'activation) et deux sorties. En adoptant les notations du chapitre 2 :

- Les vecteurs poids $(W_{11}, \cdots, W_{13}, \cdots, W_{33}, A_{11}, \cdots, A_{14}, \cdots, A_{24})$ des deux experts sont les suivants

$$MLP_1 : (0.38, 0.86, 0.88, 0.86, 0.08, -0.64, 0.54, 0.21, 0.23, 0.69, -0.77, -0.42, -0.05, 0.42, -0.52, -0.92, 0.26)$$

$$MLP_2 : (-0.36, 0.76, 0.70, -0.25, 0.94, 0.18, -0.24, 0.84, 0.16, -0.55, -0.94, -0.73, -0.48, 0.98, 0.40, -0.61, 0.74).$$

- La matrice de transition de la chaîne de Markov cachée est

$$A = \begin{pmatrix} 0.95 & 0.1 \\ 0.05 & 0.9 \end{pmatrix}.$$

- Les matrices de covariance des bruits sont

$$\Sigma_1 = \begin{pmatrix} 0.29 & 0.34 \\ 0.34 & 1.48 \end{pmatrix} \text{ et } \Sigma_2 = \begin{pmatrix} 1.09 & 0.33 \\ 0.33 & 0.1 \end{pmatrix}.$$

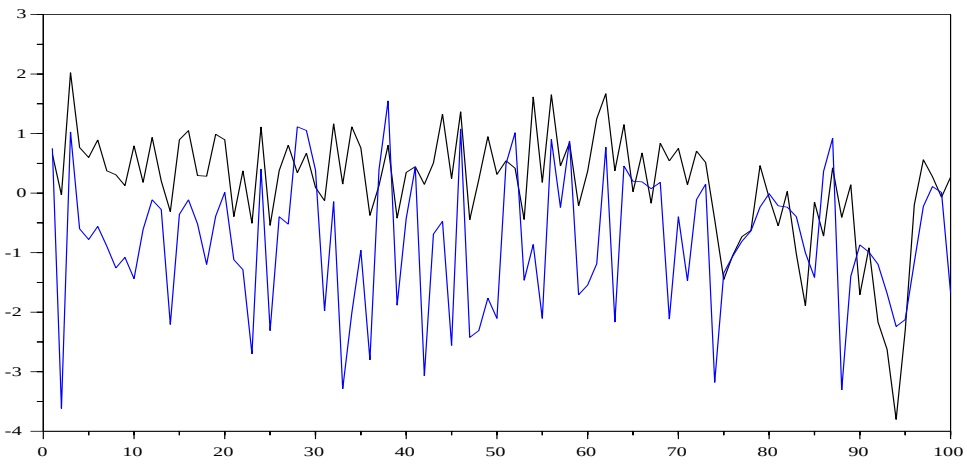
On simule une série longue de 30000 données.

Nous procédons à deux estimations :

- La première sur les 1000 premières données pour comparer les algorithmes E.M, différentiel, et stochastique
- La seconde sur toute les données pour voir les performances de l'algorithme stochastique qui est particulièrement adapté aux séries très longues.

Les 100 premières observations de la série bidimensionnelle sont représentés figure 7.1. On peut notamment remarquer que l'on ne peut pas voir "à l'oeil nu" l'existence de différents régimes.

FIG. 7.1 – La série simulée



7.5.1.2 Estimation à l'aide des 1000 premières observations

Dans toute la suite, les temps CPU donnés à titre indicatif sont obtenus sur un PC (400 Mhz).

On estime les paramètres à partir de 10 initialisations aléatoires du vecteur paramètre.

- Algorithme E.M : On fait 50 itérations de l'algorithme E.M. (cf section 5.2.2.2), avec 10 itérations de BFGS pour chaque expert afin de calculer le M-Step.
- Optimisation Différentielle : On fait 100 itérations de BFGS (cf section 2.3.1.2).
- Estimation récursive (cf section 7.4.1) : Comme l'algorithme n'a pas le temps de converger sur les 1000 premières valeurs de la série, on fait 30 passages.

On remarque sur cet exemple (cf tableau 7.1) que l'estimation différentielle directe et l'algorithme E.M. donnent des résultats à peu près équivalents, l'estimation directe étant quand même deux fois plus rapide.

L'algorithme stochastique est quant à lui plus robuste aux minima locaux et plus rapide.

Les matrices de covariances estimées pour le modèle avec la meilleure log-vraisemblance (-1.24691) sont

$$\widehat{\Sigma}_1 = \begin{pmatrix} 0.27 & 0.32 \\ 0.32 & 1.49 \end{pmatrix} \text{ et } \widehat{\Sigma}_2 = \begin{pmatrix} 1.10 & 0.33 \\ 0.33 & 0.10 \end{pmatrix}$$

et la matrice de transition estimée est

$$\widehat{A} = \begin{pmatrix} 0.96 & 0.1 \\ 0.04 & 0.9 \end{pmatrix}.$$

TAB. 7.1 – Log-vraisemblance des estimations après calcul, log-vraisemblance pour les vrais paramètres : -1.20

algorithme E.M.	optimisation différentielle	optimisation récursive
-1.36805	-2.13761	-1.30815
-1.40708	-1.69924	-1.25308
-2.4903	-1.40469	-1.33211
-1.72062	-1.53105	-1.34769
-1.35515	-1.69533	-1.33211
-1.76341	-1.67276	-1.35613
-1.48806	-1.56657	-1.2944
-1.56468	-1.75466	-1.24691
-1.5589	-1.53692	-1.3545
-1.53337	-1.56613	-1.37256
CPU : 1365 s.	CPU : 664.91 s.	CPU : 222 s.

7.5.1.3 Estimation récursive des paramètres avec toutes les données (30000)

Ici, on fait un seul passage sur toutes les données, nous n'estimons le modèle que par l'algorithme récursif, car les autres algorithmes sont très lents pour un tel nombre de données. Nous pouvons voir sur le tableau 7.2 que la log-vraisemblance des estimations est très proche de celle du vrai modèle.

Les matrices de covariances estimées pour le modèle avec la meilleure log-vraisemblance sont

$$\widehat{\Sigma}_1 = \begin{pmatrix} 0.29 & 0.35 \\ 0.35 & 1.51 \end{pmatrix} \text{ et } \widehat{\Sigma}_2 = \begin{pmatrix} 1.10 & 0.33 \\ 0.33 & 0.10 \end{pmatrix}$$

et la matrice de transition estimée est

$$\widehat{A} = \begin{pmatrix} 0.95 & 0.1 \\ 0.05 & 0.9 \end{pmatrix}.$$

L'algorithme récursif est donc très efficace, la principale difficulté pour que cette algorithme converge est de bien choisir le pas initial (γ_0). Mais ce problème est bien contrôlé par une méthode telle que la projection sur des compacts de plus en plus grands (cf Chen et al. [15]).

TAB. 7.2 – Log-vraisemblance de l'estimation récursive après 1 passage , log-vraisemblance pour les vrais paramètres : -1.18

Log-vraisemblance finale
-1.20364
-1.19179
-1.20097
-1.22429
-1.20561
-1.18873
-1.22802
-1.29867
-1.2385
-1.18573
CPU : 248.59 s.

7.5.2 Simulation avec 4 fonctions de régressions linéaires.

On augmente ici le nombre de régimes, on prend des fonctions de régression linéaires pour garder un nombre de paramètres raisonnable. Le processus reste bidimensionnel.

7.5.2.1 Le modèle

Les paramètres des fonctions autorégressives sont choisis de telle sorte que le modèle soit stable.

– Les modèles AR ont pour équations :

$$AR_1 : \begin{array}{l} Y^1(t) \\ Y^2(t) \end{array} = \begin{array}{l} 0.42Y^1(t-1) - 0.37Y^2(t-1) + 0.19 \\ -0.39Y^1(t-1) - 0.40Y^2(t-1) - 0.16 \end{array}$$

$$AR_2 : \begin{array}{l} Y^1(t) \\ Y^2(t) \end{array} = \begin{array}{l} -0.57Y^1(t-1) - 0.19Y^2(t-1) + 0.28 \\ -0.30Y^1(t-1) - 0.37Y^2(t-1) - 0.01 \end{array}$$

$$AR_3 : \begin{array}{c} Y^1(t) \\ Y^2(t) \end{array} = \begin{array}{l} 0.13Y^1(t-1) - 0.17Y^2(t-1) + -0.8 \\ -0.40Y^1(t-1) + 0.47Y^2(t-1) - 0.43 \end{array}$$

$$AR_4 : \begin{array}{c} Y^1(t) \\ Y^2(t) \end{array} = \begin{array}{l} -0.46Y^1(t-1) - 0.50Y^2(t-1) - 0.04 \\ -0.44Y^1(t-1) - 0.48Y^2(t-1) - 0.65 \end{array}.$$

- La matrice de transition est

$$A = \begin{pmatrix} 0.9 & 0.1 & 0.02 & 0.02 \\ 0.03 & 0.8 & 0.03 & 0.02 \\ 0.04 & 0.05 & 0.92 & 0.01 \\ 0.03 & 0.05 & 0.03 & 0.95 \end{pmatrix}.$$

- Les matrices de covariance des bruits sont

$$\Sigma_1 = \begin{pmatrix} 0.29 & 0.34 \\ 0.34 & 1.48 \end{pmatrix}, \Sigma_2 = \begin{pmatrix} 1.09 & 0.33 \\ 0.33 & 0.1 \end{pmatrix}$$

$$\Sigma_3 = \begin{pmatrix} 0.05 & 0.04 \\ 0.04 & 0.05 \end{pmatrix}, \Sigma_4 = \begin{pmatrix} 0.73 & 0.39 \\ 0.39 & 0.34 \end{pmatrix}.$$

7.5.2.2 Estimation à l'aide des 1000 premières observations

On estime les paramètres sur 10 initialisations aléatoires du vecteur paramètre.

- Algorithme E.M. : On fait 100 itérations de l'algorithme E.M., le calcul du M-Step est direct car le modèle est linéaire.
- Optimisation Différentielle : On fait 200 itérations de BFGS.
- Estimation récursive : Comme l'algorithme n'a pas le temps de converger sur les 1000 premières valeurs de la série, on fait 50 passages.

On remarque sur cet exemple (tableau 7.3) que l'estimation différentielle directe est plus lente et un peu moins bonne que l'algorithme E.M.

On doit cependant remarquer que puisque les fonctions de régression sont linéaires, le M-Step est immédiat car on dispose de statistiques exhaustives (cf chapitre 6).

Cela explique les bon résultats de l'algorithme E.M. dans cet exemple.

L'algorithme stochastique reste toujours plus robuste aux minima locaux et plus rapide.

Les matrices de covariances estimées pour le modèle avec la meilleure log-vraisemblance (-1.42711) sont

$$\widehat{\Sigma}_1 = \begin{pmatrix} 0.28 & 0.29 \\ 0.29 & 1.46 \end{pmatrix} \text{ et } \widehat{\Sigma}_2 = \begin{pmatrix} 0.90 & 0.27 \\ 0.27 & 0.08 \end{pmatrix}$$

$$\widehat{\Sigma}_3 = \begin{pmatrix} 0.05 & 0.04 \\ 0.04 & 0.05 \end{pmatrix} \text{ et } \widehat{\Sigma}_4 = \begin{pmatrix} 0.73 & 0.38 \\ 0.38 & 0.35 \end{pmatrix}$$

et la matrice de transition estimée est

$$\widehat{A} = \begin{pmatrix} 0.91 & 0.18 & 0.02 & 0.03 \\ 0.04 & 0.74 & 0.03 & 0.01 \\ 0.04 & 0.04 & 0.92 & 0.01 \\ 0.01 & 0.04 & 0.03 & 0.95 \end{pmatrix}.$$

TAB. 7.3 – Log-vraisemblance des estimations après calculs, log-vraisemblance pour les vrais paramètres : -1.44

algorithme E.M.	optimisation diff.	optimisation récursive
-1.55407	-1.82305	-1.43204
-1.46167	-1.59602	-1.86068
-1.91019	-1.61028	-1.43376
-1.8313	-1.82791	-1.42987
-1.55495	-1.8152	-1.42753
-1.46917	-1.6255	-1.43096
-1.49617	-1.60715	-1.42711
-1.55667	-1.63081	-1.44114
-1.5559	-1.71152	-1.44114
-1.94858	-1.81164	-1.42774
CPU :828 s.(100 it.)	CPU :1129 s.(200 it.)	CPU :550 s. (50 p.)

7.5.2.3 Estimation récursive des paramètres avec toutes les données (30000).

On fait 2 passages sur toutes les données, car le nombre de paramètres est plus grand que dans l'expérience précédente. Cela revient à trouver une initialisation proche du vrai paramètre pour le deuxième passage.

On voit sur le tableau 7.4 que la log-vraisemblance des estimations est de nouveau très proche du vrai modèle.

Les matrices de covariances estimées pour le modèle avec la meilleure log-vraisemblance (-1.21053) sont

$$\widehat{\Sigma}_1 = \begin{pmatrix} 0.30 & 0.38 \\ 0.38 & 1.58 \end{pmatrix} \text{ et } \widehat{\Sigma}_2 = \begin{pmatrix} 1.15 & 0.27 \\ 0.35 & 0.10 \end{pmatrix}$$

$$\widehat{\Sigma}_3 = \begin{pmatrix} 0.05 & 0.04 \\ 0.04 & 0.05 \end{pmatrix} \text{ et } \widehat{\Sigma}_4 = \begin{pmatrix} 0.73 & 0.38 \\ 0.38 & 0.33 \end{pmatrix}$$

et la matrice de transition estimée est

$$\widehat{A} = \begin{pmatrix} 0.90 & 0.10 & 0.02 & 0.02 \\ 0.03 & 0.79 & 0.03 & 0.02 \\ 0.04 & 0.06 & 0.92 & 0.01 \\ 0.03 & 0.05 & 0.03 & 0.95 \end{pmatrix}.$$

La log-vraisemblance estimée est ici pratiquement égale à la log-vraisemblance théorique.

TAB. 7.4 – Log-vraisemblance de l'estimation récursive après 2 passages , log-vraisemblance pour les vrais paramètres : -1.21

Log-vraisemblance finale
-1.21196
-1.21053
-1.2113
-1.21094
-1.24045
-1.23448
-1.21135
-1.21122
-1.21096
-1.21145
CPU : 776 s. (2 passages)

7.6 Conclusion

La méthode décrite permet de calculer la log-vraisemblance et sa dérivée de manière relativement simple. Cela fournit une alternative à l'algorithme d'estimation classique (algorithme E.M.) pour ce genre de modèle. La forme additive de la log-vraisemblance, permet en plus d'estimer récursivement les paramètres, ce qui est réputé robuste vis-à-vis des minima locaux (les simulations le confirment) et permet aussi d'estimer les paramètres sur des séries extrêmement longues.

Chapitre 8

Etude statistique de l'estimateur du maximum de vraisemblance

8.1 Consistance de l'estimateur du maximum de vraisemblance

8.1.1 Introduction

Dans cette partie, on montre la consistance forte de l'estimateur du maximum de vraisemblance, pour des modèles autorégressifs à changements de régime markoviens. L'étude dans le cas où ces modèles sont lipschitziens et d'ordre de régression 1, a été faite par Francq et Roussignol [28]. De plus, on trouve aussi une démonstration de la consistance de l'estimateur du maximum de vraisemblance de ces modèles dans Krishnamurthy et Rydén [40], qui adaptent la démonstration de Leroux [46] (faite dans le cas des chaînes de Markov cachées "classiques", voir section 5.1.3) aux modèles autorégressifs à changements de régime markoviens. Cependant Krishnamurthy et Rydén n'étudient pas les conditions d'ergodicité du modèle. Nous étudions ici le cas sous-linéaire (ce qui inclut le cas lipschitzien) pour un ordre de régression quelconque. Nous donnons de plus des critères simples pour la consistance de cet estimateur dans le cas gaussien. Nous reprenons les notations utilisées dans les chapitres précédents.

Soit (X_t), $t \in \mathbb{Z}$ une chaîne de Markov homogène à valeurs dans un espace d'état fini $\mathbb{E} = \{e_1, ..., e_N\}$. Soit $(Y_t) \in \mathbb{R}^d$, $t \in \mathbb{Z}$ la série des observations et pour $t \in \mathbb{Z}$, $p \in \mathbb{N}^*$, Y_{t-p}^t le vecteur $(Y_{t-p}, \cdots, Y_t)^T$. On considère le modèle suivant :

à un instant t fixé : $Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + \varepsilon_{t+1}(X_{t+1})$ où

1. Pour tout $e_i \in \mathbb{E}$, $F_{e_i} \in \{F_{e_1}, ..., F_{e_N}\}$ est une fonction borélienne de $\left(\mathbb{R}^d\right)^p \mapsto \mathbb{R}^d$.
2. Pour tout $e_i \in \mathbb{E}$, $(\varepsilon_t(e_i))_{t\in\mathbb{Z}}$ est une suite i.i.d. centrée de $\mathbb{R}^d$, les suites $(\varepsilon_t(e_i))_{1\leq i\leq N}$ sont indépendantes.

Comme dans les chapitres précédents, on identifie l'espace d'état $\mathbb{E} = \{e_1, \cdots, e_N\}$ avec le simplexe de $\mathbb{R}^N$ où e_i est un vecteur de $\mathbb{R}^N$ avec 1 sur la i-ème composante zéro et partout ailleurs.

La chaîne X_t est caractérisée par sa matrice de transition $A = (a_{ij})$ que nous définissons par :

$P(X_{t+1} = e_i / X_t = e_j) = a_{ij}$.

Ainsi, en définissant : $V_{t+1} := X_{t+1} - AX_t$, les équations générales du modèle sont :

$$\left\{ \begin{array}{c} X_{t+1} = AX_t + V_{t+1} \\ Y_{t+1} = F_{X_{t+1}}(Y_{t-p+1}^t) + \varepsilon_{t+1}(X_{t+1}) \end{array} \right. . \tag{8.1}$$

Notation 16 *Dans toute la suite, on notera :*

- *Pour tout $i, j \in \{1, \cdots, N\}$,*
  - *W_i le vecteur paramètre de la fonction F_{e_i}*
  - *β_i le vecteur paramètre de la densité ϕ_{e_i} du bruit $\varepsilon(e_i)$*
- *θ_A le vecteur paramètre qui détermine la matrice A*
- *θ le vecteur paramètre du modèle tel que*

$$\theta = (W_1, \cdots, W_N, \beta_1, \cdots \beta_N, \theta_A)$$

- *D la dimension du vecteur paramètre θ et $\Theta \subset \mathbb{R}^D$ l'ensemble des paramètres possibles*
- *$\widetilde{F}_{e_i}^{\theta}$ la fonction : $y_1^{p+1} \mapsto (y_1, \cdots, y_p, F_{W_i}(y_1^p))$.*

Remarque 17 *On identifiera Θ avec l'ensemble quotient de Θ par rapport au permutations sur $\{1, \cdots, N\}$ pour que W_i, β_i et θ_A soient déterminés à une permutation près.*

Remarque 18 *La chaîne vectorisée $\left(Y_{t-p}^t\right)_{t>\in\mathbb{Z}}$ vérifie l'équation :*

$$Y_{t-p}^t = \begin{pmatrix} Y_t \\ Y_{t-1} \\ \vdots \\ Y_{t-p} \end{pmatrix} = \begin{pmatrix} F_{X_t}^{\theta}(Y_{t-1}, \cdots, Y_{t-p}) \\ Y_{t-1} \\ \vdots \\ Y_{t-p} \end{pmatrix} + \begin{pmatrix} \varepsilon_t^{\theta}(X_t) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{8.2}$$

Le processus complet $\begin{pmatrix} X_t \\ Y_{t-p}^t \end{pmatrix}_{t\in\mathbb{Z}}$ est donc une chaîne de Markov.

8.1.2 Le contraste associé à la log-vraisemblance

8.1.2.1 La log-vraisemblance

On suppose les observations (y_{-p+1}, y_0) connues, la dépendance de la vraisemblance par rapport à (y_{-p+1}, y_0) sera par la suite implicite. On rappelle que la fonction de vraisemblance s'écrit :

$$L_\theta(y_1,...,y_n) = \prod_{t=1}^{n} L_\theta(y_t \left| y_{-p+1}^{t-1}\right.) = L_\theta(y_n \left| y_{-p+1}^{n-1}\right.) \times \prod_{t=1}^{n-1} L_\theta(y_t \left| y_{-p+1}^{t-1}\right.)$$

$$= \sum_{i=1}^{N} L_\theta(y_n \left| X_n = e_i, y_{-p+1}^{n-1}\right.) P_\theta(X_n = e_i \left| y_{-p+1}^{n-1}\right.) \times \prod_{t=1}^{n-1} L_\theta(y_t \left| y_{-p+1}^{t-1}\right.).$$

Un contraste usuel est l'opposé de la valeur moyenne de la log-vraisemblance des observations. On supposera dans cette section que pour $\theta \in \Theta$ et tout $n \in \mathbb{N}^*$, $L_\theta(y_1, \cdots, y_n) > 0$.

Notation 17 *Dans toute la suite, on notera, pour $t \in \mathbb{Z}$, $t > -p+1$*

- *p_t^θ le vecteur de $\mathbb{R}^N$ dont la i-ième composante est :*

$$p_t^\theta(i) = P_\theta(X_t = e_i \left| y_{-p+1}^{t-1}\right.) \tag{8.3}$$

- *b_t^θ le vecteur de $\mathbb{R}^N$ dont la i-ième composante est :*

$$b_t^\theta(i) = L_\theta(y_t \left| X_t = e_i, y_{-p+1}^{t-1}\right.) = \Phi_{e_i}^\theta(y_t - F_{e_i}^\theta(y_{t-p}^{t-1})) \tag{8.4}$$

- *$B_t^\theta = diag\left(b_t^\theta\right)$ la matrice diagonale ayant pour diagonale le vecteur b_t^θ.*

La vraisemblance s'écrira alors :

$$L_\theta(y_1, \cdots, y_n) = \left(b_n^\theta\right)^T p_n^\theta \times \prod_{t=1}^{n-1} L_\theta(y_t \left| y_{-p+1}^{t-1}\right.) = \prod_{t=1}^{n} \left(b_t^\theta\right)^T p_t^\theta$$

soit encore :

$$\ln(L_\theta(y_1, \cdots, y_n)) = \sum_{t=1}^{n} \ln\left(\left(b_t^\theta\right)^T p_t^\theta\right). \tag{8.5}$$

8.1.2.2 Rappel de la récurrence pour le calcul de p_t^θ

On a prouvé au chapitre 7 que la quantité

$$p_{t+1}^\theta(i) = P_\theta(X_{t+1} = e_i \left| y_{-p+1}, \cdots, y_t\right.)$$

se calculait récursivement par l'équation (7.4)

$$p_{t+1}^\theta = \frac{A_\theta B_t^\theta p_t^\theta}{\left(b_t^\theta\right)^T p_t^\theta}.$$

On prend p_1 arbitraire, par exemple la distribution uniforme sur $\{1, \cdots, N\}$ (on verra que cela ne change pas, asymptotiquement, la valeur moyenne de la log-vraisemblance). L'équation (7.4) permet de calculer p_t^θ, $t = 2, \cdots n$ par récurrence. Par la suite, la dépendance de la log-vraisemblance en p_1 sera implicite.

8.1.2.3 Contraste associé à la log-vraisemblance

Le contraste associé à la log-vraisemblance est :

$$U_n(\theta) = -\frac{1}{n}\sum_{t=1}^{n} \ln\left(\left(b_t^\theta\right)^T p_t^\theta\right). \tag{8.6}$$

Pour étudier les propriétés statistiques de l'estimateur du minimum de contraste, on est conduit à considérer le processus étendu

$$Z_t^\theta = \begin{pmatrix} X_t \\ p_t^\theta \\ Y_{t-p}^t \end{pmatrix}_{t\in\mathbb{Z}},$$

processus qui est une chaîne de Markov.

Notation 18 *On notera* $\mathbb{W}$ *l'ensemble* $\mathbb{E}\times\mathbb{C}\times\left(\mathbb{R}^d\right)^{p+1}$ *où* $\mathbb{C}$ *est l'ensemble des vecteurs de* $\mathbb{R}^N$ *dont toutes les coordonnées sont positives ou nulles et de somme égale à* 1 .

On notera aussi dans toute la suite $\lambda(\phi)$ *l'intégrale d'une fonction* ϕ *par rapport à une mesure positive* λ *et* $(\Pi\phi)(x) := \int \Pi(x,dy)\phi(x)$, *où* $\Pi(x,dy)$ *est le noyau de transition d'une chaîne de Markov.*

8.1.3 Stabilité du processus étendu Z_t^θ

8.1.3.1 Ergodicité de la chaîne $\left(X_t, Y_{t-p}^t\right)_{t\in\mathbb{Z}}^T$

On commence par établir des conditions suffisantes assurant que la chaîne de Markov $\left(X_t, Y_{t-p}^t\right)_{t\in\mathbb{Z}}^T$ est géométriquement ergodique, et que sa mesure invariante admet un moment d'ordre $s \geq 1$. On se place dans le cadre des hypothèses suivantes.

Hypothèse (S) On suppose que pour le vrai modèle (i.e. $\theta = \theta_0$) :

1. Les fonctions de régression F_{e_i} sont continues et sous-linéaires, c'est-à-dire qu'il existe $(\alpha_{e_i}, \xi_{e_i})_{i\in I} \in \left(\mathbb{R}_+^2\right)^N$ et une norme de $\left(\mathbb{R}^d\right)^{p+1}$ tels que pour tout $y_1^{p+1} \in \left(\mathbb{R}^d\right)^{p+1}$:
$$\left\|\widetilde{F}_{e_i}^{\theta_0}(y_1^{p+1})\right\| \leq \alpha_{e_i}\left\|y_1^{p+1}\right\| + \xi_{e_i}$$
2. La matrice de transition A_{θ_0} est irréductible et apériodique, on note r_0 le plus petit entier tel que :
$$\forall i,j \in \{1,\cdots,N\} : P\left(X_{r_0} = e_i \,|X_1 = e_j\right) > 0.$$
r_0 est appelé : indice de primitivité de A_{θ_0}.

3. En notant $\rho(Q_s)$ le rayon spectral de la matrice Q_s définie par :

$$Q_s = \begin{pmatrix} (\alpha_{e_1})^s a_{11} & \cdots & (\alpha_{e_N})^s a_{N1} \\ \vdots & \ddots & \vdots \\ (\alpha_{e_1})^s a_{1N} & \cdots & (\alpha_{e_N})^s a_{NN} \end{pmatrix},$$

on suppose $\rho(Q_s) < 1$.

4. $\forall e_i \in \mathbb{E}$, $E\left\|\varepsilon^{\theta_0}(e_i)\right\|^s < \infty$
5. Pour tout $e_i \in \mathbb{E}$, les variables $\varepsilon^{\theta_0}(e_i)$ ont une densité strictement positive par rapport à la mesure de Lebesgue.

On a alors le théorème suivant

Théorème 15 *Supposons que pour le modèle (8.1) les hypothèses (**S**) soient vérifiées, alors*

- *la chaîne* $\begin{pmatrix} X_t \\ Y_{t-p}^t \end{pmatrix}_{t\in\mathbb{Z}}$ *est géométriquement ergodique*
- *Sa mesure invariante* σ *admet un moment d'ordre* s.

Preuve Il s'agit d'une légère adaptation de la preuve du théorème 2 de Yao et Attali [65]. Comme $\rho(Q_s) < 1$ et que Q_s est une matrice non-négative, le théorème de Perron-Frobenius dit qu'il existe un entier positif u tel que

$$(Q_s)^u < 1.$$

[65] montre qu'alors pour la fonction de Lyapounov

$$V\left(x, y_1^{p+1}\right) = \left\|y_1^{p+1}\right\|^s + 1$$

définie sur $E \times \left(\mathbb{R}^d\right)^p$, pour un $0 < \rho_u < 1$ on a l'inégalité de contraction

$$\Pi^u V\left(x, y_1^{p+1}\right) \leq \rho_u V\left(x, y_1^{p+1}\right) + \delta_u + 1 - \rho_u \tag{8.7}$$

avec

$$\delta_u := E_{(x,y)}\left[\xi_{X_u} + \left\|\varepsilon^{X_u}\right\| + \sum_{i=1}^{u-1} \alpha_{X_u} \cdots \alpha_{X_{i+1}} \left(\xi_{X_i} + \left\|\varepsilon^{X_i}\right\|\right)\right]^s < \infty$$

où $E_{(x,y)}$ désigne l'espérance conditionnellement à $(X_0, Y_0) = (x, y)$.

D'autre part, les hypothèses (**S**)-2 et (**S**)-5 assurent que pour tout $v = \left(x, y_1^{p+1}\right) \in \mathbb{E} \times \left(\mathbb{R}^d\right)^{p+1}$, $\Pi^{\sup(p+1,r_0)}(v, dv)$ a une densité strictement positive par rapport à la mesure $\delta \otimes \lambda$, où δ est la mesure de comptage sur E et λ celle de Lebesgue sur $\left(\mathbb{R}^d\right)^{p+1}$. Cela montre que la chaîne est $\delta \otimes \lambda$-irréductible. Comme l'hypothèse (**S**)-1 implique que Π est fellerienne, la proposition 6.2.8 et le théorème 16.1.2 de [51] montre que la chaîne est V-uniforme ergodique, donc en particulier géométriquement ergodique.

Notation 19 *Dans toute la suite* $d\nu_y(y) = \sum_{i=1}^N d\sigma(e_i, y)$ *désigne la mesure stationnaire du processus* Y.

8.1.3.2 "Oubli" du vecteur initial pour p_t^θ

Le calcul de la log-vraisemblance s'appuie sur un calcul récursif des $\left(p_t^\theta\right)_{t\in\mathbb{N}^*}$ et donc dépend de la probabilité initiale p_1 . Il est donc naturel de montrer que cette condition initial soit "oublié" asymptotiquement. Cet "oubli" assure que le processus $\left(Z_t^\theta\right)_{t\in\mathbb{Z}}$ admet une seule mesure invariante. On suppose maintenant que pour tout $\theta\in\Theta$, les hypothèses suivantes sont vérifiées :

Hypothèse (P)

1. Pour tout $\theta\in\Theta$, la matrice A est irréductible apériodique.
2. Pour tout $\theta\in\Theta$, et pour tout $e_i\in\mathbb{E}$, les fonctions $F_{e_i}^\theta$ sont continues.
3. Pour tout $\theta\in\Theta$, et tout $e_i\in\mathbb{E}$, les variables $\varepsilon^\theta(e_i)$ ont une densité continue et strictement positive par rapport à la mesure de Lebesgue.

Remarque 19 *Les matrices A_θ sont de dimension $N\times N$, donc leur indice de primitivité (cf (**S**)-2) est plus petit que N^2, on notera maintenant r l'indice de primitivité maximal de A_θ pour $\theta\in\Theta$.*

Notation 20 *Notons pour tout $n\in\mathbb{N}^*$*

$$\delta_t^\theta=\frac{\max_{i\in S}b_t^\theta(i)}{\min_{i\in S}b_t^\theta(i)}<\infty\ \ et\ \ \epsilon^\theta=\min_{i,j\in S,\,a_{ij}\neq 0}a_{ij}^\theta>0.$$

On notera $\|.\|_1$ la norme L_1 de $\mathbb{R}^N$.

Considérons $\left(p_t^\theta\right)_{t\in\mathbb{N}^*}$ et $\left(p_t^{\theta\prime}\right)_{t\in\mathbb{N}^*}$ les processus issus de p_1 et p_1'définis par la récurrence (7.4). On a la proposition suivante :

Proposition 5 *Sous les hypothèses (**S**), pour tout θ vérifiant les hypothèses (**P**), il existe un réel $0<\rho<1$ tel que pour tous vecteurs initiaux de probabilités p_1,p_1' de $\mathbb{R}^N$*

$$\limsup_t\left\|p_t^\theta-p_t^{\theta\prime}\right\|_1^{\frac{1}{t}}\overset{p.s.}{<}\rho$$

donc, en particulier

$$\left\|p_t^\theta-p_t^{\theta\prime}\right\|_1\xrightarrow{p.s.}0$$

et comme pour tout $t\in\mathbb{N}^*$, $\left\|p_t^\theta\right\|_1=1$

$$E\left[\left\|p_t^\theta-p_t^{\theta\prime}\right\|_1\right]\longrightarrow 0.$$

Preuve Mevel et Legland ([45],[44]) montrent que pour tous entiers t et r tels que $t \geq r$, on a la majoration :

$$\left\|p_t^\theta - p_t^{\theta\prime}\right\|_1 \leq 2\left(\epsilon^\theta\right)^{-r} \delta_1^\theta \cdots \delta_r^\theta \prod_{\kappa=0}^{\left[\frac{t}{r}\right]-1} \left(1 - \left(\epsilon^\theta\right)^r \left[\delta_{\kappa r+2}^\theta \cdots \delta_{(\kappa+1)r}^\theta\right]^{-1}\right) \left\|p_1 - p_1'\right\|_1$$

où $\left[\frac{t}{r}\right]$ représente la partie entière de $\frac{t}{r}$ et $\|.\|$ est la norme L_1 de $\mathbb{R}^N$. On en déduit que pour $t \geq 2r$

$$\left\|p_t^\theta - p_t^{\theta\prime}\right\|_1^{\frac{1}{\left[\frac{t}{r}\right]-1}}$$

$$\leq \left(2\left(\epsilon^\theta\right)^{-r} \delta_1^\theta \cdots \delta_r^\theta \left\|p_1 - p_1'\right\|_1\right)^{\frac{1}{\left[\frac{t}{r}\right]-1}} \times \left(\prod_{\kappa=0}^{\left[\frac{t}{r}\right]-1} \left(1 - \left(\epsilon^\theta\right)^r \left[\delta_{\kappa r+2}^\theta \cdots \delta_{(\kappa+1)r}^\theta\right]^{-1}\right)\right)^{\frac{1}{\left[\frac{t}{r}\right]-1}}$$

$$= \left(2\left(\epsilon^\theta\right)^{-r} \delta_1^\theta \cdots \delta_r^\theta \left\|p_1 - p_1'\right\|_1\right)^{\frac{1}{\left[\frac{t}{r}\right]-1}} \exp\left(\frac{\sum_{\kappa=0}^{\left[\frac{t}{r}\right]-1} \ln\left(1 - \left(\epsilon^\theta\right)^r \left[\delta_{\kappa r+2}^\theta \cdots \delta_{(\kappa+1)r}^\theta\right]^{-1}\right)}{\left[\frac{t}{r}\right] - 1}\right).$$

Comme δ_t^θ est supérieur ou égal à 1 , on a

$$0 < \Delta_\theta = \mathbb{E}\left[\left(\delta_{tr+2}^\theta \cdots \delta_{(t+1)r}^\theta\right)^{-1}\right] \leq 1$$

et par la loi des grands nombres et l'inégalité de Jensen

$$\limsup_t \left\|p_t^\theta - p_t^{\theta\prime}\right\|_1^{\frac{1}{\left[\frac{t}{r}\right]-1}} \overset{p.s.}{\leq} \exp\left(\ln\left(1 - \left(\epsilon^\theta\right)^r \Delta_\theta\right)\right) < 1$$

en prenant $0 < \rho = \left(\exp\left(\ln\left(1 - \left(\epsilon^\theta\right)^r \Delta_\theta\right)\right)\right)^{\frac{1}{r}} < 1$, on déduit la propriété annoncée.

8.1.3.3 Application au processus Z_t^θ

De l'ergodicité géométrique de $\left(X_t, Y_{t-p}^t\right)^T$ et de la proposition 5, on déduit le lemme suivant :

Lemme 8 *Soit $Z_t^\theta = \left(X_t^x, \left(Y_{t-p}^t\right)^y, p_t^\theta\right)$ et $Z_t^{\theta\prime} = \left(X_t^{x'}, \left(Y_{t-p}^t\right)^{y'}, p_t^{\theta\prime}\right)$ deux réalisations du processus Z_t^θ issues de deux valeurs initiales $z = (x, p_1, y), z' = (x', p_1', y') \in \mathbb{W}$, sous les hypothèses (**S**) et (**P**) on aura*

$$E\left[\left\|Z_t^\theta - Z_t^{\theta\prime}\right\|\right] \overset{t\to\infty}{\longrightarrow} 0 \tag{8.8}$$

Preuve Pour toutes normes $\|.\|_a$ de $\mathbb{W}$, $\|.\|_b$ de $\mathbb{E}\times\left(\mathbb{R}^d\right)^{p+1}$, il existe deux constantes c_1 et c_2 telles que pour tout $(x,p,y_1^{p+1})\in\mathbb{W}$,

$$\left\|x,p,y_1^{p+1}\right\|_a \leq c_1\left\|x,y_1^{p+1}\right\|_b + c_2\left\|p\right\|_1 .$$

On a donc

$$E\left[\left\|Z_t^z - Z_{t'}^{z'}\right\|_a\right] \leq c_1 E\left[\left\|\left(X_t^x,\left(Y_{t-p}^t\right)^y\right)^T - \left(X_t^{x'},\left(Y_{t-p}^t\right)^{y'}\right)^T\right\|_b\right] + c_2 E\left[\left\|p_t^\theta - p_t^{\theta\prime}\right\|_1\right].$$

Maintenant l'ergodicité géométrique de $\left(X_t,Y_{t-p}^t\right)$ implique

$$E\left[\left\|\left(X_t^x,\left(Y_{t-p}^t\right)^y\right)^T - \left(X_t^{x'},\left(Y_{t-p}^t\right)^{y'}\right)^T\right\|_b\right] \overset{k\to\infty}{\longrightarrow} 0$$

et la proposition 5 implique $E\left[\left\|p_t^\theta - p_t^{\theta\prime}\right\|_1\right] \overset{k\to\infty}{\longrightarrow} 0$, ce qui termine la preuve.

On peut alors énoncer le théorème :

Théorème 16 *Pour tout $\theta\in\Theta$, sous les hypothèses (**S**) et (**P**), si V est la fonction :*

$$z=(x,p,y)\in\mathbb{W}\mapsto V(z)=\|y\|^s+1$$

alors

- *le processus $\left(Z_t^\theta\right)_{t\in\mathbb{Z}}$ est stable de loi stationnaire notée ν^θ,*
- *pour toute fonction g ν^θ-p.s. continue telle que $\forall z\in\mathbb{W}, |g(z)|\leq const.\left(V^{1-\eta}(z)+1\right)$ pour une constante $\eta>0$*

$$\frac{1}{n}\sum_{t=1}^n g\left(Z_t^\theta\right) \overset{p.s.}{\longrightarrow} \nu^\theta(g).$$

Preuve Comme pour tout $t\in\mathbb{N}^*$, p_t^θ est un vecteur de probabilité, la démonstration du théorème 15 montre que pour la fonction de Lyapounov :

$$V(x,p,y_1^{p+1}) = \left\|y_1^{p+1}\right\|^s + 1$$

il existe un entier $u>0$ tels que

$$\Pi^u V(z) \leq \rho_u V(z) + \delta_u + 1 - \rho_u.$$

Le lemme 8 implique

$$\forall z,z'\in\mathbb{W}^2,\ Z_t^\theta - Z_t^{\theta\prime} \overset{Loi}{\longrightarrow} 0$$

ce qui est un critère d'unicité de la mesure invariante (cf Duflo [25]). Le théorème est alors une conséquence du théorème 5 de [65].

8.1.4 Consistance du maximum de vraisemblance

Nous allons d'abord donner un cadre général pour la consistance forte de l'estimateur du maximum de vraisemblance.

8.1.4.1 Hypothèses de base

Notation 21 *Pour $\theta \in \Theta$, notons $\widetilde{L}_\theta\left(Y_1^{p+1}\right)$, la vraisemblance de Y_1^{p+1} pour la mesure invariante dont l'existence et l'unicité sont assurées par le théorème 15.*

On se place dans le cadre des hypothèses suivantes.

Hypothèse (H)

1. Les hypothèses (**S**) et (**P**) sont vérifiées.
2. Θ est un ouvert de $\mathbb{R}^D$ qui contient le vrai paramètre θ_0.
3. Pour tout $\left(y_1^{p+1}\right) \in \left(\mathbb{R}^d\right)^{p+1}$, les fonctions $\theta \mapsto \Phi_{e_i}^\theta(y_{p+1} - F_{e_i}^\theta(y_1^p))$ et $\theta \mapsto A_\theta$ sont continues sur Θ.
4. Il existe $\delta > 0$ et $s > 1$ tels que pour tout $i \in \{1, \cdots, N\}$ et $\theta \in \Theta$

$$E_{\theta_0}\left[\sup_{\|\theta-\theta'\|<\delta}\left|\ln\left(\Phi_{e_i}^{\theta'}(Y_{p+1} - F_{e_i}^{\theta'}(Y_1^p))\right)\right|^s\right] < \infty. \tag{8.9}$$

5. Identifiabilité : Si $\theta \neq \theta'$, alors les vraisemblances $\widetilde{L}_\theta\left(Y_1^{p+1}\right)$ et $\widetilde{L}_{\theta'}\left(Y_1^{p+1}\right)$ sont presque sûrement différentes.
6. Les observations $(Y_t)_{t\in\mathbb{Z}}$ sont générées par la solution stationnaire de 8.1.

Remarque 20 *La condition (H)-4 est légèrement plus forte que celle de Krishnamurty et Rydén [40] (il suffit dans leur preuve que $s = 1$). Cette condition permet cependant de simplifier notablement la preuve de la consistance.*

8.1.4.2 Processus de contraste

La proposition suivante montre que la valeur moyenne de la log-vraisemblance est bien un processus de contraste définie comme dans [19].

Proposition 6 *Sous les hypothèses (**H**) :*

il existe une constante finie $K(\theta, \theta_0) \geq 0$, telle que, pour $n \longrightarrow \infty$

$$U_n(\theta) - U_n(\theta_0) \xrightarrow{p.s.} K(\theta, \theta_0)$$

avec $K(\theta, \theta_0) = 0$ si et seulement si $\theta = \theta_0$.

Preuve La condition (**H**)-4 et le théorème 16 assurent que pour tout $\theta \in \Theta$, lorsque $n \longrightarrow \infty$

$$U_n(\theta) \xrightarrow{p.s.} U(\theta) < \infty.$$

Soit λ^d la mesure de Lebesgue sur $\mathbb{R}^d$ et ν_y^p est la mesure invariante de Y_1^p sur $\left(\mathbb{R}^d\right)^p$. Pour $1 \leq i \leq N$, notons $\hat{p}_1^\theta(i) = E_\theta\left[X_1 = e_i \,|Y_0, \cdots\right]$, l'existence et l'unicité de $\hat{p}_1^\theta$ sont assurées par le théorème 16.

De la même façon que pour le lemme 4 de Krishnamurthy et Rydén [40], on a alors presque sûrement :

$$\lim_{n\to\infty} U_n(\theta) - U_n(\theta_0) =$$

$$-\int\int \ln\left(\frac{\sum_{i=1}^N \left(\Phi_{e_i}^\theta(Y_1 - F_{e_i}^\theta(Y_{-p+1}^0))\right) \times E_\theta\left(\hat{p}_1^\theta(i) \,\middle|Y_{-p+1}^0\right)}{\sum_{i=1}^N \left(\Phi_{e_i}^{\theta_0}(Y_1 - F_{e_i}^{\theta_0}(Y_{-p+1}^0))\right) \times E_{\theta_0}\left(\hat{p}_1^{\theta_0}(i) \,\middle|Y_{-p+1}^0\right)}\right) d\lambda^d(Y_1)\, d\nu_y^p(Y_{-p+1}^0)$$

$$= -E_{\theta_0}\left[\ln\left(\frac{\widetilde{L}_\theta\left(Y_{-p+1}^1\right)}{\widetilde{L}_{\theta_0}\left(Y_{-p+1}^1\right)}\right)\right].$$

L'inégalité de Jensen montre que $K(\theta, \theta_0) \geq 0$. Enfin si $\theta \neq \theta_0$, l'hypothèse d'identifiabilité (**H**)-5 donne $K(\theta, \theta_0) > 0$.

8.1.4.3 Consistance forte

Suivant la présentation de Francq et Roussignol [28], on établit d'abord le lemme :

Lemme 9 *Pour tout $\theta_1 \in \Theta$, $\theta_1 \neq \theta_0$, il existe un voisinage $V(\theta_1)$ de θ_1 tel que :*

$$\limsup_n \sup_{\theta \in V(\theta_1)} -\left(U_n(\theta) - U_n(\theta_0)\right) < 0$$

Preuve Soit $V_m(\theta_1)$ la boule ouverte de centre θ_1 et de rayon $1/m$. Puisque Θ est ouvert, on a, pour m assez grand, $V_m(\theta_1) \in \Theta$.

Soit

$$S_n^m(Y_1, \cdots, Y_n) = \sup_{\theta \in V_m(\theta_1)} \prod_{t=1}^n \left(b_t^\theta\right)^T p_t^\theta,$$

pour tous entiers positifs m, n, t , on a

$$\ln S_{n+k}^m(Y_1, \cdots, Y_{n+k}) \leq \ln S_n^m(Y_1, \cdots, Y_n) + \ln S_t^m(Y_{t+1}, \cdots, Y_{n+k}).$$

La condition (**H**)-4 et le théorème ergodique pour des processus sous-additifs de Kingman [39] donne p.s.

$$\lim_{n\to\infty} \frac{1}{n} \ln S_n^m(Y_1, \cdots, Y_n) = \kappa_m(\theta_1) := \inf_{n\geq 1} \frac{1}{n} E_{\theta_0}\left[\ln S_n^m(Y_1, \cdots, Y_n)\right].$$

Maintenant, en remarquant que la log-vraisemblance (8.5) est additive, donc sous-additive, on a aussi

$$\lim_{n\to\infty} \frac{1}{n} \ln L_\theta (Y_1, \cdots, Y_n) := \inf_{n\geq 1} \frac{1}{n} E_{\theta_0} [\ln L_\theta (Y_1, \cdots, Y_n)] .$$

Donc la proposition 6 assure que pour tout $\epsilon > 0$ et $\theta \in V_m(\theta_1)$, il existe n_ϵ tel que

$$\frac{1}{n_\epsilon} E_{\theta_0} [\ln L_\theta (Y_1, \cdots, Y_n)] < \epsilon.$$

Les hypothèses (**H**)-3, (**H**)-4 et le théorème de convergence dominée assurent que pour m assez grand, on a

$$\left| \frac{1}{n_\epsilon} E_{\theta_0} [\ln L_\theta (Y_1, \cdots, Y_n)] - \frac{1}{n_\epsilon} E_{\theta_0} [\ln S_n^m (Y_1, \cdots, Y_n)] \right| < \frac{\epsilon}{2}.$$

On en déduit

$$\kappa_m (\theta_1) \leq \frac{1}{n_\epsilon} E_{\theta_0} [\ln S_n^m (Y_1, \cdots, Y_n)] < \frac{\epsilon}{2}.$$

■

Le Lemme 9 prouve la consistance forte de l'estimateur du minimum de contraste sur tout sous ensemble compact contenant θ_0 .

Théorème 17 *Soit $\Theta^* \subset \Theta$ un compact contenant le vrai paramètre θ_0. Sous les hypothèses (**H**), l'estimateur du minimum de contraste*

$$\hat{\theta}_n = \inf_{\theta\in\Theta^*} U_n (\theta)$$

converge presque sûrement vers θ_0 quand $n \to \infty$.

8.1.5 Exemple

Nous allons montrer comment on peut vérifier les hypothèses (**H**), dans un cas particulier, mais important.

Considérons le modèle (8.1), avec un bruit gaussien :

Hypothèses (G) On supposera que les hypothèses (**S**) sont vérifiées avec un moment $s > 2$, que les hypothèses (**P**) et (**H**) sont vérifiées, sauf les conditions (**H**)-4 et (**H**)-5 que nous remplaçons par

1. Il existe $0 < \lambda^i_{\max} < \infty$ et $0 < \lambda^i_{\min} < \infty$, tels que pour tout $\theta \in \Theta$, pour tout $e_i \in E$, ε^{e_i} est normal $\mathcal{N}(0, \Sigma^\theta_i)$ où Σ^θ_i est une matrice définie positive de $\mathbb{R}^{d\times d}$dont la plus grande et la plus petite valeur propre sont bornées respectivement par $\lambda^i_{\max}$ et $\lambda^i_{\min}$.

2. Identifiabilité : Pour tout $\theta, \theta' \in \Theta$, pour tout $i \in S$ on aura $\omega_i = \omega_i' \Leftrightarrow F_{e_i}^{\theta} = F_{e_i}^{\theta'}$, $\beta_i = \beta_i' \Leftrightarrow \Sigma_i^{\theta} = \Sigma_i^{\theta'}$, $\theta_A = \theta_A' \Leftrightarrow A_\theta = A_{\theta'}$ et pour $i, j \in S^2$, $i \neq j$ on a, soit $\omega_i \neq \omega_j$, soit $\beta_i \neq \beta_j$

Remarque 21 *Si les fonctions de régression sont linéaires, ces conditions peuvent être affaiblies (cf Francq et Roussignol [28]).*

8.1.5.1 Identifiabilité du modèle

Supposons que pour deux paramètres θ et θ', on ait

$$\widetilde{L}_\theta \left(Y_1^{p+1}\right) \overset{p.s.}{=} \widetilde{L}_{\theta'} \left(Y_1^{p+1}\right), \tag{8.10}$$

il est facile de vérifier qu'alors

$$\widetilde{L}_\theta \left(Y_1^{p+1}, Y_2^{q+2}\right) \overset{p.s.}{=} \widetilde{L}_{\theta'} \left(Y_1^{p+1}, Y_2^{q+2}\right). \tag{8.11}$$

Soit $\hat{p}_1^\theta$ défini comme dans la preuve de la proposition 6, le lemme 9 de [28] donne :

Lemme 10 *Sous les hypothèses (**G**), pour tout $\theta \in \Theta$ et tout $i \in I$,*

$$\hat{p}_1^\theta (i) > 0$$

En particulier, pour presque tout y_{-p+1}^0, $\widetilde{p}_{y_{-p+1}^0}^{\theta} := E_\theta \left[\hat{p}_1^\theta \left| y_{-p+1}^0 \right.\right]$ est à coordonnées strictement positives.

Si la première égalité (8.10) est vérifiée, on a pour presque tout y_{-p+1}^0

$$\sum_{i=1}^{N} \widetilde{p}_{y_{-p+1}^0}^{\theta} (i)\, \Phi_{e_i}^{\theta} \left(y_1 - F_{e_i}^{\theta} \left(y_{-p+1}^0\right)\right) \overset{p.s}{=} \sum_{i=1}^{N} \widetilde{p}_{y_{-p+1}^0}^{\theta'} (i)\, \Phi_{e_i}^{\theta'} \left(y_1 - F_{e_i}^{\theta'} \left(y_{-p+1}^0\right)\right).$$

La densité du bruit étant gaussienne, Teicher [59] montre que

$$\sum_{i \in S} \widetilde{p}_{y_{-p+1}^0}^{\theta} (i)\, \delta_{\left(F_{e_i}^{\theta}\left(y_{-p+1}^0\right), \Sigma_{e_i}^{\theta}\right)} \overset{p.s}{=} \sum_{i \in S} \widetilde{p}_{y_{-p+1}^0}^{\theta'} (i)\, \delta_{\left(F_{e_i}^{\theta'}\left(y_{-p+1}^0\right), \Sigma_{e_i}^{\theta'}\right)} \tag{8.12}$$

où δ représente ici la mesure de Dirac. On déduit de (**G**)-2 qu'à une permutation sur $\{1, \cdots, N\}$ près, on a

$$\omega_i = \omega_i', \; i \in S \tag{8.13}$$

$$\beta_i = \beta_i', \; i \in S \tag{8.14}$$

et

$$\widetilde{p}_{y_{-p+1}^0}^{\theta} \overset{p.s.}{=} \widetilde{p}_{y_{-p+1}^0}^{\theta'}. \tag{8.15}$$

Maintenant, de (8.11), on déduit que p.s. en y_{-p+1}^1 on a

$$\sum_{j=1}^{N}\sum_{i=1}^{N}\widetilde{p}_{y_{-p+1}^0}^{\theta}(j)\,\Phi_{e_j}^{\theta}\left(y_1-F_{e_j}^{\theta}\left(y_{-p+1}^0\right)\right)a_{ij}^{\theta}\Phi_{e_i}^{\theta}\left(y_2-F_{e_i}^{\theta}\left(y_{-p+2}^1\right)\right)$$

$$\overset{p.s.}{=}\sum_{j=1}^{N}\sum_{i=1}^{N}\widetilde{p}_{y_{-p+1}^0}^{\theta'}(j)\,\Phi_{e_j}^{\theta'}\left(y_1-F_{e_j}^{\theta'}\left(y_{-p+1}^0\right)\right)a_{ij}^{\theta'}\Phi_{e_i}^{\theta'}\left(y_2-F_{e_i}^{\theta'}\left(y_{-p+2}^1\right)\right).$$

La densité de y_2 est un mélange de N gaussiennes, donc (8.13) et (8.14) donnent pour tout $i\in\{1,\cdots,N\}$

$$\sum_{j=1}^{N}\widetilde{p}_{y_{-p+1}^0}^{\theta}(j)\,\Phi_{e_j}^{\theta}\left(y_1-F_{e_j}^{\theta}\left(y_{-p+1}^0\right)\right)a_{ij}^{\theta}\overset{p.s.}{=}\sum_{j=1}^{N}\widetilde{p}_{y_{-p+1}^0}^{\theta'}(j)\,\Phi_{e_j}^{\theta'}\left(y_1-F_{e_j}^{\theta'}\left(y_{-p+1}^0\right)\right)a_{ij}^{\theta'}.$$

La densité de y_1 est alors un mélange des N gaussiennes et on a presque sûrement, pour tout $i,j\in I^2$

$$\widetilde{p}_{y_{-p+1}^0}^{\theta}(j)\,a_{ij}^{\theta}\overset{p.s.}{=}\widetilde{p}_{y_{-p+1}^0}^{\theta'}(j)\,a_{ij}^{\theta'}$$

donc, comme pour tout $j\in S$, $\widetilde{p}_{y_{-p+1}^0}^{\theta}(j)\overset{p.s.}{=}\widetilde{p}_{y_{-p+1}^0}^{\theta'}(j)>0$, on a

$$\forall i,j\in\{1,\cdots,N\}\ a_{ij}^{\theta}=a_{ij}^{\theta'}\Leftrightarrow\theta_A=\theta'_A,$$

le modèle est identifiable.

8.1.5.2 Vérification de l'hypothèse (H)-4

Notons $0<\lambda_{\max}=\sup_i\lambda_{\max}^i<\infty$ et $0<\lambda_{\min}=\min_i\lambda_{\min}^i<\infty$, l'hypothèse (**S**)-1 donne l'existence d'une constante C telle que :

$$\sup_{1\le i\le N}\sup_{\theta\in\Theta}\left|\ln\left(\Phi_{e_i}(y_{p+1}-F_{e_i}(y_1^p))\right)\right|$$

$$\le d\times\sup\left(\left|\ln\left(\lambda_{\min}\right)\right|,\left|\ln\left(\lambda_{\max}\right)\right|\right)+\frac{C}{\lambda_{\min}}\left(\sup_i\left(\alpha_{e_i}\right)\left\|y_1^{p+1}\right\|^2+\sup_i\left(\beta_{e_i}\right)\right):=h\left(y_1^{p+1}\right)$$

avec

$$h\left(y_1^{p+1}\right)\le const.\left(\left\|y_1^{p+1}\right\|^{s-\eta}+1\right)$$

pour un $\eta>0$, ce qui assure que l'hypothèse (**H**)-4 est vérifié.

On peut alors appliquer le théorème 17.

8.2 Normalité asymptotique du maximum de vraisemblance

La démonstration de la normalité asymptotique de l'estimateur du maximum de vraisemblance, pour notre modèle, est une adaptation de celle de Bickel, Ritov et Rydén [10], car ces auteurs ont traité le cas des modèles de chaînes de Markov cachées sans fonctions autorégressives, mais à observations continues. Il est suffisant de montrer que les lemmes préliminaires permettant d'établir des majorations similaires à [10], sont vérifiés dans notre cas et la démonstration se déduit aisément de la leur.

8.2.1 Hypothèses et simplifications

On se reportera aux hypothèses suivantes (**N**) dans toute la suite

1. La chaîne de Markov $(X_t)_{t\in\mathbb{Z}}$ est supposée irréductible et apériodique. Cela implique qu'il existe un entier $r > 0$ tel que $\forall X \in \mathbb{E}$, $A^r(X) > 0$. Afin de simplifier les notations on suppose que $r = 1$. Cependant les propriétés s'étendent facilement au cas $r > 1$ (cf Bickel et al. [10]). De même pour simplifier les notations, on suppose que l'ordre maximal des fonctions de régression est 1 (i.e. $p = 1$) l'extension à $p > 1$ est immédiate.
2. Pour tout $e_i \in \mathbb{E}$, les applications $\theta \longmapsto A^\theta$, $\theta \longmapsto \pi_\theta$[1], $\theta \longmapsto F^\theta_{e_i}$ et $\theta \longmapsto \phi^\theta_{e_i}$ sont deux fois continûment dérivables dans un voisinage $|\theta - \theta_0| < \delta$ avec $\delta > 0$.
3. Notons $\theta = (\theta_1, \cdots, \theta_B)$. Il existe $\delta > 0$ tel que
   (a) Pour tout $1 \leq i \leq B$ et tout k, $1 \leq k \leq N$
   $$E_0\left[\sup_{|\theta-\theta_0|<\delta}\left|\frac{\partial}{\partial\theta_i}\ln\left(\phi^\theta_{e_k}\left(Y_1 - F^\theta_{e_k}(Y_0)\right)\right)\right|^2\right] < \infty.$$
   (b) Pour tout $1 \leq i, j \leq B$ et tout k, $1 \leq k \leq N$
   $$E_0\left[\sup_{|\theta-\theta_0|<\delta}\left|\frac{\partial^2}{\partial\theta_i\partial\theta_j}\ln\left(\phi^\theta_{e_k}\left(Y_1 - F^\theta_{e_k}(Y_0)\right)\right)\right|\right] < \infty.$$
   (c) Pour tout $j = 1, 2$, tout $1 \leq i_l \leq B$, $l = 1, \cdots, j$, tout k, $1 \leq k \leq N$ et tout $y_0 \in \mathbb{R}^d$
   $$\int_{\mathbb{R}^d}\sup_{|\theta-\theta_0|<\delta}\left|\frac{\partial^j}{\partial\theta_{i_1}\cdots\partial\theta_{i_l}}\left(\phi^\theta_{e_k}\left(Y_1 - F^\theta_{e_k}(y_0)\right)\right)\right| d\lambda(Y_1) < \infty.$$

[1] On trouvera dans Bickel et al. [10] des exemples où cette propriété est vérifiée, elle le sera notamment pour la paramétrisation du chapitre 7.

4. Il existe $\delta > 0$, tel qu'avec

$$\rho_0(y_0, y_1) = \sup_{|\theta-\theta_0|<\delta} \max_{1\leq i,j\leq N} \frac{\phi^\theta_{e_i}\left(y_1 - F^\theta_{e_i}(y_0)\right)}{\phi^\theta_{e_j}\left(y_1 - F^\theta_{e_j}(y_0)\right)}$$

on a pour tout i, $1 \leq i \leq N$

$$P_0\left(\rho_0(Y_0, Y_1) = \infty \,|X_1 = e_i\right) < 1.$$

5. Le vrai paramètre θ_0 appartient à l'intérieur de Θ, pour simplifier les notations on supposera que ce paramètre est unidimensionnel, cela n'enlève pas de difficulté principale à la démonstration et cela nous dispense d'utiliser des notations comme uu^T.
6. Les observations $(Y_t)_{t\in\mathbb{Z}}$ sont générées par la solution stationnaire de (8.1) et l'estimateur du maximum de vraisemblance est fortement consistant.

Sans perte de généralité, on suppose que δ convient à (**N**)-2, (**N**)-3 et (**N**)-4.

Notation 22

- *On remplace souvent dans la suite l'indice θ_0 par 0 pour simplifier légèrement les notations.*
- *Pour $m \leq n \in \mathbb{N}^*$, les suites $Y_m, \cdots, Y_n$ (resp. $X_m, \cdots, X_n$) seront abrégées en Y_m^n (resp. X_m^n).*
- *Le symbole ∇ notera la dérivée du premier ordre par rapport à θ et H celle du second ordre.*
- *C désignera une constante positive et finie.*
- *Dans toute la suite, pour tout entier $k < l$ et toute suite réelle $(x_t)_{t\in\mathbb{Z}}$, on utilisera la convention :*

$$\prod_{t=l}^{k} x_t := 1$$

c'est-à-dire que si l'indice du bas est plus grand que celui du haut, le produit ci-dessus vaut 1.

La vraisemblance du modèle Soit π_θ la loi invariante de la chaîne de Markov $(X_t)_{t\in\mathbb{Z}}$ de matrice de transition A_θ.

La vraisemblance du modèle pour une suite d'observations de la série $y := (y_{-p+1}, \cdots, n)$ pour un chemin réalisé $x := \{x_t,\, t = 1, ..., n\}$ s'écrit :

$$L_\theta(y, x) = \prod_{t=1}^{n}\prod_{i=1}^{N}\left[\Phi^\theta_{e_i}(y_t - F^\theta_{e_i}(y_{t-p}^{t-1}))\right]^{1_{\{e_i\}}(x_t)} \times \prod_{t=1}^{n-1}\prod_{i,j=1}^{N}\left(a^\theta_{ij}\right)^{1_{\{e_j,e_i\}}(x_t,x_{t+1})} \times \pi_\theta(X_1) \tag{8.16}$$

La vraisemblance globale des observations peut donc s'écrire :

$L_\theta(y) = \sum_x L_\theta(y,x)$, où $\sum_x$ représente la somme sur tous les chemins possibles de la chaîne de Markov cachée.

8.2.2 Un théorème central limite pour la fonction score

8.2.2.1 Introduction

Soit $n \in \mathbb{N}^*$ et $L_\theta\left(Y_1 \left| Y_{-n}^0\right.\right)$ la densité conditionnelle de Y_1 sachant $Y_0, \cdots, Y_{-n}$. Par la définition du modèle

$$L_\theta\left(Y_1 \left| Y_{-n}^0\right.\right) = \sum_{i=1}^{N} \phi_{e_i}^\theta\left(Y_1 - F_{e_i}^\theta(Y_0)\right) P_\theta\left(X_1 = e_i \left| Y_{-n}^0\right.\right).$$

Il est facile de vérifier (cf théorème 16) que, P_θ-p.s.,

$$P_\theta\left(X_1 = e_i \left| Y_{-n}^0\right.\right) \overset{n\to\infty}{\longrightarrow} P_\theta\left(X_1 = e_i \left| Y_{-\infty}^0\right.\right).$$

Ainsi, on a P_θ-p.s.

$$L_\theta\left(Y_1 \left| Y_{-n}^0\right.\right) \overset{n\to\infty}{\longrightarrow} L_\theta\left(Y_1 \left| Y_{-\infty}^0\right.\right).$$

Nous allons donc étudier les propriétés de la dérivée de cette densité conditionnelle.

8.2.2.2 Dérivée de la densité conditionnelle.

Grâce à une égalité générale pour des modèles avec données manquantes (cf [48]) on a

$$\nabla \ln L_\theta\left(Y_1 \left| Y_{-n}^0\right.\right) = \nabla \ln L_\theta\left(Y_{-n}^1\right) - \nabla \ln L_\theta\left(Y_{-n}^0\right)$$
$$= E_\theta\left[\nabla \ln L_\theta\left(X_{-n+1}^1, Y_{-n}^1\right) \left| Y_{-n}^1\right.\right] - E_\theta\left[-\nabla \ln L_\theta\left(X_{-n+1}^1, Y_{-n}^0\right) \left| Y_{-n}^0\right.\right]$$

Notation 23 *Ecrivons* $\lambda_\theta(e_j, e_i) = \nabla \ln\left(a_{ij}^\theta\right)$, $\gamma_\theta(y_0, y_1 | e_i) = \nabla \ln\left(\phi_{e_i}^\theta\left(y_1 - F_{e_i}^\theta(y_0)\right)\right)$ *et* $\tau_\theta(e_i) = \nabla \ln(\pi_\theta(e_i))$

On a

$$\nabla \ln L_0\left(Y_1 \left| Y_{-n}^0\right.\right) =$$
$$\sum_{t=-n+1}^{0} \left\{E_0\left[\gamma_0\left(Y_{t-1}, Y_t \left| X_t\right.\right) + \lambda_0\left(X_t, X_{t+1}\right) \left| Y_{-n}^1\right.\right]\right.$$
$$\left. - E_0\left[\gamma_0\left(Y_{t-1}, Y_t \left| X_t\right.\right) + \lambda_0\left(X_t, X_{t+1}\right) \left| Y_{-n}^0\right.\right]\right\} \tag{8.17}$$
$$+E_0\left[\gamma_0\left(Y_0, Y_1 \left| X_1\right.\right) \left| Y_{-n}^1\right.\right] + E_0\left[\tau_0\left(X_{-n+1}\right) \left| Y_{-n}^1\right.\right] - E_0\left[\tau_0\left(X_{-n+1}\right) \left| Y_{-n}^0\right.\right].$$

Définissons

$$\eta_1 = \sum_{t=-\infty}^{0} \left\{ E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) + \lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-\infty}^{1} \right. \right] \right.$$

$$\left. - E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) + \lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-\infty}^{0} \right. \right] \right\} \tag{8.18}$$

$$+ E_0 \left[\gamma_0 \left(Y_0, Y_1 \left| X_1 \right. \right) \left| Y_{-\infty}^{1} \right. \right].$$

La somme dans (8.18) est absolument convergente dans $\mathbb{L}_2 \left(P_0 \right)$, ainsi la partie droite de l'égalité (8.18) est une variable aléatoire dans $\mathbb{L}_2 \left(P_0 \right)$. Nous ne montrons pas cela ici, mais cela est une conséquence de la démonstration du lemme 14. Nous définissons la matrice d'information de Fisher comme

$$\mathcal{J}_0 = E_0 \left[\eta_1 \eta_1^T \right].$$

8.2.2.3 Quelques lemmes préliminaires

Sous les conditions (**N**)-1 et (**N**)-2 il existe $\delta > 0$ et $\sigma_0 > 0$ tels que

$$\inf_{|\theta - \theta_0| < \delta} \left\{ a_{ij}^{\theta},\ i, j \in \{1, \cdots, N\} \right\} \geq \sigma_0$$

et

$$\inf_{|\theta - \theta_0| < \delta} \left\{ \pi_\theta \left(i \right),\ i \in \{1, \cdots, N\} \right\} \geq \sigma_0.$$

Sans perte de généralité, on suppose que ce δ convient aux hypothèses (**N**)-2, (**N**)-3 et (**N**)-4.

Notation 24 *Soit*

$$\mu_0 \left(y_0, y_1 \right) = \left\{ 1 + \left(N - 1 \right) \left(\sigma_0^{-2} \rho_0 \left(y_0, y_1 \right) \right)^2 \right\}^{-1}.$$

On établit alors le lemme suivant

Lemme 11 *Soit $-n \leq l \leq t \leq 0$, et G_t un évènement concernant uniquement X_t^0 et Y_t^0, alors pour tout θ tel que $|\theta - \theta_0| < \delta$,*

$$\max_{e_i \in E} P_\theta \left(G_t \left| Y_{-n-1}^{0}, X_l = e_i \right. \right) - \min_{e_i \in E} P_\theta \left(G_t \left| Y_{-n-1}^{0}, X_l = e_i \right. \right)$$

$$\leq \prod_{i=l+1}^{t-1} \left(1 - 2\mu_0 \left(Y_{i-1}, Y_i \right) \right) \leq \prod_{i=l+1}^{t-1} \exp \left(-2\mu_0 \left(Y_{i-1}, Y_i \right) \right).$$

Preuve Soit $e_j, e_k \in E$ et $-n \leq t < 0$, on a

$$\frac{P_\theta\left(X_{t+1}=e_j \left| X_t=e_i, Y_{-n-1}^0\right.\right)}{P_\theta\left(X_{t+1}=e_k \left| X_t=e_i, Y_{-n-1}^0\right.\right)}$$

$$= \frac{L_\theta\left(X_{t+1}=e_j, X_t=e_i, Y_{-n-1}^0\right)}{L_\theta\left(X_{t+1}=e_k, X_t=e_i, Y_{-n-1}^0\right)}$$

$$= \frac{L_\theta\left(X_{t+1}=e_j, X_t=e_i, Y_{t+1}^0 \left| Y_t\right.\right) \times L_\theta\left(Y_t \left| Y_{-n-1}^{t-1}\right.\right) \times L_\theta\left(Y_{-n-1}^{t-1}\right)}{L_\theta\left(X_{t+1}=e_k, X_t=e_i, Y_{t+1}^0 \left| Y_t\right.\right) \times L_\theta\left(Y_t \left| Y_{-n-1}^{t-1}\right.\right) \times L_\theta\left(Y_{-n-1}^{t-1}\right)}$$

$$= \frac{L_\theta\left(X_{t+1}=e_j, Y_{t+1}^0 \left| X_t=e_i, Y_t\right.\right)}{L_\theta\left(X_{t+1}=e_k, Y_{t+1}^0 \left| X_t=e_i, Y_t\right.\right)}.$$

Supposons que $t < -1$, cette quantité vaudra alors

$$= \frac{\sum_{j_0 \in N} \phi_{e_j}^\theta\left(Y_{t+1} - F_{e_j}^\theta\left(Y_t\right)\right) a_{ji} a_{j_0 j} \times L_\theta\left(Y_{t+2}^0 \left| X_{t+2}=e_{j_0}, Y_{t+1}\right.\right)}{\sum_{j_0 \in N} \phi_{e_k}^\theta\left(Y_{t+1} - F_{e_k}^\theta\left(Y_t\right)\right) a_{ki} a_{j_0 k} \times L_\theta\left(Y_{t+2}^0 \left| X_{t+2}=e_{j_0}, Y_{t+1}\right.\right)}$$

$$\leq \rho_0\left(Y_t, Y_{t+1}\right) \max_{i,j,j_0,k \in \{1,\cdots,N\}} \frac{a_{ji} a_{j_0 j}}{a_{ki} a_{j_0 k}} \leq \rho_0\left(Y_t, Y_{t+1}\right) \sigma_0^{-2} \leq \left(\rho_0\left(Y_t, Y_{t+1}\right) \sigma_0^{-2}\right)^2.$$

Maintenant, si $t = -1$ on aura

$$\frac{L_\theta\left(X_{t+1}=e_j, Y_{t+1}^0 \left| X_t=e_i, Y_t\right.\right)}{L_\theta\left(X_{t+1}=e_k, Y_{t+1}^0 \left| X_t=e_i, Y_t\right.\right)} = \frac{\phi_{e_j}^\theta\left(Y_{t+1} - F_{e_j}^\theta\left(Y_t\right)\right) a_{ji}}{\phi_{e_k}^\theta\left(Y_{t+1} - F_{e_k}^\theta\left(Y_t\right)\right) a_{ki}} \leq \rho_0\left(Y_t, Y_{t+1}\right) \sigma_0^{-2}.$$

On en déduit donc

$$P_\theta\left(X_{t+1}=e_j \left| X_t=e_i, Y_{-n-1}^0\right.\right) \geq \left(1 + (N-1)\left(\rho_0\left(Y_t, Y_{t+1}\right) \sigma_0^{-2}\right)^2\right)^{-1}$$

La preuve se finit alors exactement comme le lemme 2.2 et le corollaire 2.3 de Baum et Petri [7] ■

On établit maintenant le lemme suivant

Lemme 12 *Soit $n \in \mathbb{N}^*$ et $-n \leq t \leq 0$, on définit*

$$S_\theta(n,t) = \max_{e_i,e_j,e_k \in E} \left| P_\theta\left(X_t=e_i \left| Y_{-n-1}^0, X_1=e_j\right.\right) - P_\theta\left(X_t=e_i \left| Y_{-n-1}^0, X_1=e_k\right.\right)\right|$$

alors, pour tout θ tel que $|\theta - \theta_0| < \delta$,

$$S_\theta(n,t) \leq \prod_{l=t+1}^{0} \exp\left(-2\mu_0\left(Y_{l-1}, Y_l\right)\right).$$

Preuve Soit $e_i, e_j, e_k \in E$ et $-n \leq t \leq 0$, on a

$$\frac{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_j\right.\right)}{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_k\right.\right)}$$

$$= \frac{L_\theta\left(X_{t+1} = e_j, X_t = e_i, Y_{-n-1}^0\right) \times L_\theta\left(Y_{-n-1}^0, X_{t+1} = e_k\right)}{L_\theta\left(X_{t+1} = e_k, X_t = e_i, Y_{-n-1}^0\right) \times L_\theta\left(Y_{-n-1}^0, X_{t+1} = e_j\right)}$$

$$= \frac{L_\theta\left(X_{t+1} = e_j, X_t = e_i, Y_{-n-1}^0\right)}{L_\theta\left(X_{t+1} = e_k, X_t = e_i, Y_{-n-1}^0\right)} \times \frac{\sum_{l=1}^N L_\theta\left(Y_{-n-1}^0, X_{t+1} = e_k, X_t = e_l\right)}{\sum_{l=1}^N L_\theta\left(Y_{-n-1}^0, X_{t+1} = e_j, X_t = e_l\right)}.$$

Supposons que $t < 0$, alors il découle de la preuve du lemme 11 que

$$\frac{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_j\right.\right)}{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_k\right.\right)} \leq \left(\rho_0\left(Y_t, Y_{t+1}\right)\sigma_0^{-2}\right)^2.$$

Maintenant si $t = 0$, on a :

$$\frac{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_j\right.\right)}{P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_k\right.\right)} \leq \sigma_0^{-2} \leq \left(\rho_0\left(Y_t, Y_{t+1}\right)\sigma_0^{-2}\right)^2$$

On en déduit donc

$$P_\theta\left(X_t = e_i \left| Y_{-n-1}^0, X_{t+1} = e_j\right.\right) \geq \left(1 + (N-1)\left(\rho_0\left(Y_t, Y_{t+1}\right)\sigma_0^{-2}\right)^2\right)^{-1}$$

et la démonstration se finit de la même façon que le lemme précédent. ■

On établit aussi le lemme

Lemme 13 *Soit $-m \leq -n < t \leq 0$, pour tout θ tel que $|\theta - \theta_0| < \delta$,*

1.

$$\max_{e_i \in E} \left|P_\theta\left(X_t = e_i \left| Y_{-n-1}^1\right.\right) - P_\theta\left(X_t = e_i \left| Y_{-n-1}^0\right.\right)\right|$$

$$\leq \prod_{l=t+1}^{0} \exp\left(-2\mu_0\left(Y_{l-1}, Y_l\right)\right)$$

2.

$$\max_{e_i, e_j \in E} \left|P_\theta\left(X_t = e_i, X_{t+1} = e_j \left| Y_{-n-1}^1\right.\right) - P_\theta\left(X_t = e_i, X_{t+1} = e_j \left| Y_{-n-1}^0\right.\right)\right|$$

$$\leq \prod_{l=t+2}^{0} \exp\left(-2\mu_0\left(Y_{l-1}, Y_l\right)\right)$$

3.

$$\max_{e_i \in E} \left| P_\theta \left(X_t = e_i \left| Y_{-n-1}^1 \right. \right) - P_\theta \left(X_t = e_i \left| Y_{-m-1}^0 \right. \right) \right|$$

$$\leq \prod_{l=-n+1}^{t-1} \exp \left(-2\mu_0 \left(Y_{l-1}, Y_l \right) \right)$$

4.

$$\max_{e_i, e_j \in E} \left| P_\theta \left(X_t = e_i, X_{t+1} = e_j \left| Y_{-n-1}^1 \right. \right) - P_\theta \left(X_t = e_i, X_{t+1} = e_j \left| Y_{-m-1}^0 \right. \right) \right|$$

$$\leq \prod_{l=-n+1}^{t-1} \exp \left(-2\mu_0 \left(Y_{l-1}, Y_l \right) \right).$$

Les deux premières conclusions sont aussi vraies P_θ-p.s. si on remplace $-n$ par $-\infty$ et les deux dernières conclusions sont aussi vraies si on remplace $-m$ par $-\infty$. Dans les deux dernières conclusions, on peut aussi remplacer Y_{-n-1}^1 (resp. Y_{-m-1}^1) par Y_{-n-1}^0 (resp. Y_{-m-1}^0), et étendre comme ci-dessus la conclusion à un m infini.

Preuve Supposons d'abord que n et m sont finis. On utilise le même raisonnement que Bickel et Ritov [11], ainsi

$$P_\theta \left(X_t = e_i \left| Y_{-n-1}^0 \right. \right) = \sum_{j=1}^N P_\theta \left(X_t = e_i \left| Y_{-n-1}^0, X_1 = e_j \right. \right) \times P_\theta \left(X_1 = e_j \left| Y_{-n-1}^0 \right. \right)$$

et

$$P_\theta \left(X_t = e_i \left| Y_{-n-1}^1 \right. \right) = \sum_{j=1}^N P_\theta \left(X_t = e_i \left| Y_{-n-1}^0, X_1 = e_j \right. \right) \times P_\theta \left(X_1 = e_j \left| Y_{-n-1}^0 \right. \right)$$

donc, on a

$$\max_{e_i \in E} \left| P_\theta \left(X_t = e_i \left| Y_{-n-1}^1 \right. \right) - P_\theta \left(X_t = e_i \left| Y_{-n-1}^0 \right. \right) \right|$$

$$\leq \max_{e_i, e_j, e_k \in E} \left| P_\theta \left(X_t = e_i \left| Y_{-n-1}^0, X_1 = e_j \right. \right) - P_\theta \left(X_t = e_i \left| Y_{-n-1}^0, X_1 = e_k \right. \right) \right| = S_\theta \left(n, t \right).$$

La première inégalité du lemme est alors une conséquence du lemme 12. Les inégalités suivantes s'obtiennent exactement de la même façon que dans la démonstration du lemme 5 de [10] ■

On peut maintenant prouver le résultat suivant :

Lemme 14 *Il existe des constantes $C_0 \in \mathbb{R}_+^*$, $\beta_0 \in [0,1[$ telles que*

$$\left\| \nabla \ln L_0 \left(Y_1 \left| Y_0, \cdots, Y_{-n-1} \right. \right) - \eta_1 \right\| \leq C_0 \beta_0^n$$

Preuve En comparant (8.17) et (8.18) on voit qu'il est suffisant de prouver que

–
$$\left\| E_0 \left[\tau_0 \left(X_{-n} \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\tau_0 \left(X_{-n} \right) \left| Y_{-n-1}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.19}$$

–
$$\left\| E_0 \left[\gamma_0 \left(Y_0, Y_1 \left| X_1 \right. \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\gamma_0 \left(Y_0, Y_1 \left| X_1 \right. \right) \left| Y_{-\infty}^1 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.20}$$

–
$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \left\| E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n-1}^0 \right. \right] - E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-\infty}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.21}$$

–
$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \left\| E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-\infty}^1 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.22}$$

–
$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \left\| E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-n-1}^0 \right. \right] - E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-\infty}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.23}$$

–
$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \left\| E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-\infty}^1 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.24}$$

–
$$\sum_{t=-n}^{-\left[\frac{n}{2}\right]-1} \left\| E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n-1}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.25}$$

–
$$\sum_{t=-n}^{-\left[\frac{n}{2}\right]-1} \left\| E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-n-1}^1 \right. \right] - E_0 \left[\lambda_0 \left(X_t, X_{t+1} \right) \left| Y_{-n-1}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.26}$$

–
$$\sum_{t=-\infty}^{-\left[\frac{n}{2}\right]-1} \left\| E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-\infty}^1 \right. \right] - E_0 \left[\gamma_0 \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-\infty}^0 \right. \right] \right\|_2 \leq C_0 \beta_0^n. \tag{8.27}$$

–

$$\sum_{t=-\infty}^{-\left[\frac{n}{2}\right]-1} \left\| E_0\left[\lambda_0\left(X_t, X_{t+1}\right)\left|Y_{-\infty}^1\right.\right] - E_0\left[\lambda_0\left(X_t, X_{t+1}\right)\left|Y_{-\infty}^0\right.\right]\right\|_2 \leq C_0\beta_0^n \quad (8.28)$$

où l'on note [.] la partie entière.

Preuve Puisque que nous avons obtenu des majorations similaires à Bickel et al. [10], leur preuve s'applique à notre cas. Nous allons, par exemple, montrer l'inégalité (8.20) qui n'est pas traitée en détail dans leur article. On a

$$\left|E_0\left[\gamma_0\left(Y_0, Y_1\left|X_1\right.\right)\left|Y_{-n-1}^1\right.\right] - E_0\left[\gamma_0\left(Y_0, Y_1\left|X_1\right.\right)\left|Y_{-\infty}^1\right.\right]\right|$$

$$= \left|\sum_{i=1}^N \gamma_0\left(Y_0, Y_1\left|X_1 = e_i\right.\right)\left[P_0\left(X_1 = e_i\left|Y_{-n-1}^1\right.\right) - P_0\left(X_1 = e_i\left|Y_{-\infty}^1\right.\right)\right]\right|$$

$$\leq \max_{1\leq i\leq N}\left|\gamma_0\left(Y_0, Y_1\left|X_1 = e_i\right.\right)\right| C \prod_{t=-n+1}^0 \exp\left(-2\mu_0\left(Y_{t-1}, Y_t\right)\right).$$

Donc par la définition du modèle

$$\left\|E_0\left[\gamma_0\left(Y_0, Y_1\left|X_1\right.\right)\left|Y_{-n-1}^1\right.\right] - E_0\left[\gamma_0\left(Y_0, Y_1\left|X_1\right.\right)\left|Y_{-\infty}^1\right.\right]\right\|_2^2$$

$$\leq CE_0\left[\prod_{t=-n+1}^0 \exp\left(-4\mu_0\left(Y_{t-1}, Y_t\right)\right)\right]$$

$$= CE_0\left[E_0\left[\prod_{t=-n+1}^0 \exp\left(-4\mu_0\left(Y_{t-1}, Y_t\right)\right)\left|X_{-n+1}^0\right.\right]\right]$$

$$= CE_0\left[\prod_{t=-n+1}^0 E_0\left[\exp\left(-4\mu_0\left(Y_{t-1}, Y_t\right)\right)\left|X_t\right.\right]\right]$$

$$\leq CE_0\left[\prod_{t=-n+1}^0 \max_{1\leq i\leq N} E_0\left[\exp\left(-4\mu_0\left(Y_{t-1}, Y_t\right)\right)\left|X_t = e_i\right.\right]\right] = C\beta^n$$

pour un $\beta \in]0,1[$, on en déduit l'inégalité (8.20).

Remarque 22 *Les inégalités (8.27) et (8.28) montrent que* $\eta_1 \in \mathbb{L}_2\left(P_0\right)$.

8.2.2.4 Normalité asymptotique de la fonction score

Soit $\xi_t := \nabla \ln\left(L_0\left(Y_t \left| Y_0^{t-1}\right.\right)\right)$, on a, en notant

$$\nabla \ln L_0\left(y_0^n\right) := \nabla l_0\left(y_0^n\right) = \sum_{t=1}^{n} \xi_t.$$

Soit

$$\eta_t = \sum_{l=-\infty}^{t-1} \left\{ E_0\left[\gamma_0\left(Y_{l-1}, Y_l \left| X_l\right.\right) + \lambda_0\left(X_l, X_{l+1}\right) \left| Y_{-\infty}^{t}\right.\right]\right.$$

$$\left. - E_0\left[\gamma_0\left(Y_{l-1}, Y_l \left| X_l\right.\right) + \lambda_0\left(X_l, X_{l+1}\right) \left| Y_{-\infty}^{t-1}\right.\right]\right\}$$

$$+ E_0\left[\gamma_0\left(Y_{t-1}, Y_t \left| X_t\right.\right) \left| Y_{-\infty}^{t}\right.\right].$$

On a

$$E_0\left[\gamma_0\left(Y_0, Y_1 \left| X_1\right.\right) \left| Y_{-\infty}^{0}\right.\right]$$

$$= E_0\left[E_0\left[\gamma_0\left(Y_0, Y_1 \left| Y_{-\infty}^{0}, X_1\right.\right)\right] \left| Y_{-\infty}^{0}\right.\right]$$

$$= E_0\left[E_0\left[\gamma_0\left(Y_0, Y_1 \left| Y_0, X_1\right.\right)\right] \left| Y_{-\infty}^{0}\right.\right].$$

Mais, pour tout $i, 1 \leq i \leq N$, et tout $y_0 \in \mathbb{R}^d$:

$$\int \phi_{e_i}\left(y_1 - F_{W_i}\left(y_0\right)\right) dy_1 = 1$$

donc, grâce à l'hypothèse (**N**)-3c,

$$E_0\left[\gamma_0\left(Y_0, Y_1 \left| Y_0, X_1\right.\right)\right] = 0.$$

Ainsi, $\left(\eta_t\right)_{t\in\mathbb{N}}$ est un incrément de martingale stationnaire et ergodique par rapport à la tribu $\left\{\mathcal{F}_t := \sigma\left(Y_{-\infty}^{t}\right)\right\}$ dans $\mathbb{L}_2\left(P_0\right)$. Sa matrice de covariance est $\mathcal{J}_0$.

Montrons maintenant que $\left(\eta_t\right)$ satisfait la condition de Lindeberg suivante (Duflo [25]) :

Proposition 7 *Pour tout* $\epsilon > 0$

$$\frac{1}{n}\sum_{t=1}^{n} E_0\left[\left\|\eta_t\right\|^2 \mathbb{I}_{\left\{\left\|\eta_t\right\| \geq \epsilon\sqrt{n}\right\}} \left| \mathcal{F}_{t-1}\right.\right] \xrightarrow{P_0} 0.$$

Preuve Soit $A > 0$ et :

$$F_n(A) := \frac{1}{n}\sum_{t=1}^{n} E_0\left[\|\eta_t\|^2 \mathbb{I}_{\{\|\eta_t\|\geq \epsilon A\}} |\mathcal{F}_{t-1}\right].$$

On a

$$E_0\left[\|\eta_t\|^2 \mathbb{I}_{\{\|\eta_t\|\geq \epsilon A\}} |\mathcal{F}_{t-1}\right] \leq E_0\left[\|\eta_t\|^2 |\mathcal{F}_{t-1}\right]$$

et comme $\|\eta_t\|^2$ est intégrable, par la loi forte des grands nombres, on a :

$$F_n(A) \xrightarrow{p.s.} \Phi(A) = E_0\left[E_0\left[\|\eta_t\|^2 \mathbb{I}_{\{\|\eta_t\|\geq \epsilon A\}} |\mathcal{F}_{t-1}\right]\right]$$

Φ est décroissante et positive. Le théorème de convergence dominée montre que, quand A tend vers ∞, $\Phi(A)$ tend vers 0 . Enfin, pour A fixé, on a si n est assez grand : $\epsilon\sqrt{n} > A$, et $F_n(\epsilon\sqrt{n}) \leq F_n(A)$, donc p.s. $\limsup_n F_n(\epsilon\sqrt{n}) \leq \Phi(A)$. Finalement, en faisant tendre $A \longrightarrow \infty$, on obtient p.s. :

$$\lim_{n\longrightarrow\infty} \frac{1}{n}\sum_{t=1}^{n} E\left[\|\eta_t\|^2 \mathbb{I}_{\left\{\|\eta_t\|\geq \epsilon\sqrt{n}\right\}} |\mathcal{F}_{t-1}\right] = 0.$$

■

On peut maintenant établir le théorème de normalité asymptotique :

Théorème 18 *Supposons les hypothèses (**N**) vérifiées, alors on a*

$$n^{-\frac{1}{2}} \nabla l_0(y_0^n) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathcal{J}_0),$$

lorsque $n \longrightarrow \infty$.

Preuve La proposition 7 assure que

$$n^{-\frac{1}{2}} \sum_{t=1}^{n} \eta_t \xrightarrow{\mathcal{L}} \mathcal{N}(0, \mathcal{J}_0).$$

De plus, par stationnarité de $(Y_t)_{t\in\mathbb{N}}$ et $(\eta_t)_{t\in\mathbb{N}}$, on aura

$$\left\| n^{-\frac{1}{2}} \sum_{t=1}^{n} \xi_t - n^{-\frac{1}{2}} \sum_{t=1}^{n} \eta_t \right\| \leq n^{-\frac{1}{2}} \sum_{t=1}^{n} \|\xi_t - \eta_t\|_2$$

$$= n^{-\frac{1}{2}} \sum_{t=1}^{n} \left\| \nabla \ln L_0\left(Y_1 \left| Y_{-k+1}^0\right.\right) - \eta_1 \right\|_2$$

et par le lemme 14, on a

$$n^{-\frac{1}{2}} \sum_{t=1}^{n} \left\| \nabla \ln L_0 \left(Y_1 \left| Y_{-k+1}^0 \right. \right) - \eta_1 \right\|_2 \overset{n\to\infty}{\longrightarrow} 0$$

donc

$$n^{-\frac{1}{2}} \nabla l_0 \left(y_0^n \right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \mathcal{J}_0\right).$$

■

8.2.3 Une loi des grands nombres pour la dérivée du second ordre

On prouve ici une loi des grands nombres uniforme pour la Hessienne de la log-vraisemblance. L'approche est similaire à celle de la section 8.2.2.

8.2.3.1 Dérivée d'ordre 2 de la densité conditionnelle

De nouveau, grâce à une égalité générale pour des modèles avec données manquantes (cf [48]) on a :

$$\begin{aligned}
&H \ln L_\theta \left(Y_1 \left| Y_{-n}^0 \right. \right) = H \ln L_\theta \left(Y_{-n}^1 \right) - H \ln L_\theta \left(Y_{-n}^0 \right) \\
&= E_\theta \left[H \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \left| Y_{-n}^1 \right. \right] + E_\theta \left[\left(\nabla \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \right)^2 \left| Y_{-n}^1 \right. \right] \\
&\quad - \left\{ E_\theta \left[\nabla \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \left| Y_{-n}^1 \right. \right] \right\}^2 \\
&- E_\theta \left[H \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \left| Y_{-n}^0 \right. \right] - E_\theta \left[\left(\nabla \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \right)^2 \left| Y_{-n}^0 \right. \right] \\
&\quad + \left\{ E_\theta \left[\nabla \ln L_\theta \left(X_{-n+1}^1, Y_{-n}^1 \right) \left| Y_{-n}^0 \right. \right] \right\}^2 \\
&= \sum_{t=-n+1}^{0} \left\{ E_\theta \left[\nabla \gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) + \nabla \lambda_\theta \left(X_t, X_{t+1} \right) \left| Y_{-n}^1 \right. \right] \right. \\
&\quad \left. - E_\theta \left[\nabla \gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) + \nabla \lambda_\theta \left(X_t, X_{t+1} \right) \left| Y_{-n}^0 \right. \right] \right\} \\
&+ E_\theta \left[\nabla \gamma_\theta \left(Y_0, Y_1 \left| X_1 \right. \right) \left| Y_{-n}^1 \right. \right] + E_\theta \left[\nabla \tau_\theta \left(X_{-n+1} \right) \left| Y_{-n}^1 \right. \right] - E_\theta \left[\nabla \tau_\theta \left(X_{-n+1} \right) \left| Y_{-n-1}^0 \right. \right] \\
&+ \sum_{t=-n+1}^{0} \sum_{l=-n+1}^{0} \left\{ E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^1 \right. \right] \right. \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n}^1 \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^1 \right. \right] \\
&\quad \left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^0 \right. \right] \right.
\end{aligned}$$

$$
\begin{aligned}
&+E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^0\right.\right]E_\theta\left[\gamma_\theta\left(Y_{l-1},Y_l\left|X_l\right.\right)\left|Y_{-n}^0\right.\right]\\
&+E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^1\right.\right]-E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^1\right.\right]\\
&-E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^0\right.\right]+E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^0\right.\right]E_\theta\left[\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^0\right.\right]\\
&+2E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^1\right.\right]-2E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^1\right.\right]\\
&-2E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^0\right.\right]+2E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^0\right.\right]E_\theta\left[\lambda_\theta\left(X_l,X_{l+1}\right)\left|Y_{-n}^0\right.\right]\Big\}\\
&+E_\theta\left[\gamma_\theta^2\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right]-\left\{E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right]\right\}^2\\
&+\sum_{t=-n+1}^{0}\Big\{2E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^1\right.\right]\\
&-2E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^1\right.\right]\\
&+2E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^1\right.\right]-2E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^1\right.\right]\Big\}\\
&+E_\theta\left[\tau_\theta^2\left(X_{-n+1}\right)\left|Y_{-n}^1\right.\right]-\left\{E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^1\right.\right]\right\}^2\\
&-E_\theta\left[\tau_\theta^2\left(X_{-n+1}\right)\left|Y_{-n}^0\right.\right]-\left\{E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^0\right.\right]\right\}^2\\
&+\sum_{t=-n+1}^{0}\Big\{2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^1\right.\right]\\
&-2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^1\right.\right]\\
&-2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^0\right.\right]+2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^0\right.\right]E_\theta\left[\gamma_\theta\left(Y_{t-1},Y_t\left|X_t\right.\right)\left|Y_{-n}^0\right.\right]\\
&+2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^1\right.\right]-2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^1\right.\right]\\
&-2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^0\right.\right]+2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^0\right.\right]E_\theta\left[\lambda_\theta\left(X_t,X_{t+1}\right)\left|Y_{-n}^0\right.\right]\Big\}\\
&+2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right]-2E_\theta\left[\tau_\theta\left(X_{-n+1}\right)\left|Y_{-n}^1\right.\right]E_\theta\left[\gamma_\theta\left(Y_0,Y_1\left|X_1\right.\right)\left|Y_{-n}^1\right.\right].
\end{aligned}
\tag{8.29}
$$

De nouveau il faut établir quelques lemmes préliminaires

8.2.3.2 Quelques lemmes préliminaires

Le lemme suivant se démontre d'une façon entièrement similaire au lemme 13 :

Lemme 15 *Soit $-m \leq -n < t, l \leq 0$, on a pour tout θ tel que $|\theta - \theta_0| < \delta$,*

1.

$$
\begin{aligned}
&\max_{i,j\in\{1,\cdots,N\}}\left|P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-n-1}^1\right.\right)\right|\\
&\qquad\leq\prod_{i=t\vee l+1}^{0}\exp\left(-2\mu_0\left(Y_{i-1},Y_i\right)\right),
\end{aligned}
$$

2.

$$\max_{i,j\in\{1,\cdots,N\}} \left|P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-m-1}^1\right.\right)\right|$$

$$\leq \prod_{i=-n+1}^{t\wedge l-1}\exp\left(-2\mu_0\left(Y_{i-1},Y_i\right)\right).$$

La seconde égalité est aussi vraie si Y_{-n-1}^1 et Y_{-m-1}^1 sont remplacés respectivement par Y_{-n-1}^0 et Y_{-m-1}^0.

On établit maintenant le lemme :

Lemme 16 *Soit $-n\leq t,l\leq 0$, on a pour tout θ tel que $|\theta-\theta_0|<\delta$,*

$$\max_{i,j\in\{1,\cdots,N\}} \left|P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|Y_{-n-1}^1\right.\right)P_\theta\left(X_t=e_j\left|Y_{-n-1}^1\right.\right)\right|$$

$$\leq \prod_{i=t\wedge l+1}^{t\vee l-1}\exp\left(-2\mu_0\left(Y_{i-1},Y_i\right)\right).$$

Cette égalité est aussi vraie si Y_{-n-1}^1 est remplacé par Y_{-n-1}^0.

Preuve Supposons que $t\geq l$, alors

$$\left|P_\theta\left(X_t=e_i,X_l=e_j\left|Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|Y_{-n-1}^1\right.\right)P_\theta\left(X_t=e_j\left|Y_{-n-1}^1\right.\right)\right|$$

$$=\left|P_\theta\left(X_t=e_i\left|X_l=e_j,Y_{-n-1}^1\right.\right)P_\theta\left(X_t=e_j\left|Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|Y_{-n-1}^1\right.\right)P_\theta\left(X_t=e_j\left|Y_{-n-1}^1\right.\right)\right|$$

$$\leq\left|P_\theta\left(X_t=e_i\left|X_l=e_j,Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|Y_{-n-1}^1\right.\right)\right|$$

$$=\left|\sum_{k=1}^N\left[P_\theta\left(X_t=e_i\left|X_l=e_j,Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|X_l=e_k,Y_{-n-1}^1\right.\right)\right]P_\theta\left(X_l=e_k\left|Y_{-n-1}^1\right.\right)\right|$$

$$\leq\max_{i,j,k\in\{1,\cdots,N\}}\left|P_\theta\left(X_t=e_i\left|X_l=e_j,Y_{-n-1}^1\right.\right)-P_\theta\left(X_t=e_i\left|X_l=e_k,Y_{-n-1}^1\right.\right)\right|$$

$$\leq\prod_{i=l+1}^{t-1}\exp\left(-2\mu_0\left(Y_{i-1},Y_i\right)\right)$$

où la dernière inégalité est une conséquence du lemme 11. La preuve avec Y_{-n-1}^0 est analogue ■

Notons V le voisinage $\{\theta:|\theta-\theta_0|<\delta\}$ de θ_0, on a alors le lemme :

Lemme 17 *Lorsque $m,n\longrightarrow 0$,*

$$\left\|\sup_{\theta\in V}\left|H\ln L_\theta\left(Y_1\left|Y_{-m-1}^1\right.\right)-H\ln L_\theta\left(Y_1\left|Y_{-n-1}^1\right.\right)\right|\right\|_1\longrightarrow 0.$$

Preuve Tout comme dans [10], considérons (8.29), on voit alors que l'on doit prouver, par exemple,

$$\begin{aligned}
&\Big\| \sup_{\theta\in V} \Big| \sum_{t=-m+1}^{0} \sum_{l=-m+1}^{0} \Big\{ E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{1} \right] \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{1} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{1} \right] \\
&\qquad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \\
&\quad + E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{0} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \Big\} \qquad (8.30) \\
&- \sum_{t=-n+1}^{0} \sum_{l=-n+1}^{0} \Big\{ E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-n}^{1} \right] \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{1} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{1} \right] \\
&\qquad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \\
&+ E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{0} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \Big\} \Big| \Big\|_1 \overset{n,m\to\infty}{\longrightarrow} 0.
\end{aligned}$$

Les autres inégalités, similaires à (8.30) et qui avec (8.30) prouvent le lemme, peuvent être montrées avec de légères adaptations. Pour prouver (8.30), en supposant que $m \geq n$, il est suffisant de montrer que :

–

$$\begin{aligned}
&\sum_{t=-m+1}^{-\left[\frac{n}{2}\right]} \sum_{l=t}^{\left[\frac{t}{2}\right]} \Big\| \sup_{\theta\in V} \Big| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{1} \right] \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \Big| \Big\|_1 \longrightarrow 0 \qquad (8.31)
\end{aligned}$$

–

$$\begin{aligned}
&\sum_{t=-m+1}^{-\left[\frac{n}{2}\right]} \sum_{l=t}^{\left[\frac{t}{2}\right]} \Big\| \sup_{\theta\in V} \Big| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{1} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{1} \right] \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \Big| Y_{-m}^{0} \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-m}^{0} \right] \Big| \Big\|_1 \longrightarrow 0 \qquad (8.32)
\end{aligned}$$

–

$$\begin{aligned}
&\sum_{t=-n+1}^{-\left[\frac{n}{2}\right]} \sum_{l=t}^{\left[\frac{t}{2}\right]} \Big\| \sup_{\theta\in V} \Big| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-n}^{1} \right] \\
&\quad - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \,|X_t \right) \gamma_\theta \left(Y_{l-1}, Y_l \,|X_l \right) \Big| Y_{-n}^{0} \right] \Big| \Big\|_1 \longrightarrow 0 \qquad (8.33)
\end{aligned}$$

–

$$\sum_{t=-n+1}^{-\left[\frac{n}{2}\right]} \sum_{l=t}^{\left[\frac{t}{2}\right]} \left\| \sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n}^1 \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^1 \right. \right] \right.\right.$$

$$\left.\left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n}^0 \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^0 \right. \right] \right| \right\|_1 \longrightarrow 0 \qquad (8.34)$$

et pour $j = 0, 1$

–

$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \sum_{l=-\left[\frac{n}{2}\right]}^{0} \left\| \sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^j \right. \right] \right.\right.$$

$$\left.\left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^j \right. \right] \right| \right\|_1 \longrightarrow 0 \qquad (8.35)$$

–

$$\sum_{t=-\left[\frac{n}{2}\right]}^{0} \sum_{l=-\left[\frac{n}{2}\right]}^{0} \left\| \sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-m}^j \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^j \right. \right] \right.\right.$$

$$\left.\left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n}^j \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^j \right. \right] \right| \right\|_1 \longrightarrow 0 \qquad (8.36)$$

–

$$\sum_{t=-m+1}^{-\left[\frac{n}{2}\right]} \sum_{l=-\left[\frac{t}{2}\right]}^{0} \left\| \sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^j \right. \right] \right.\right.$$

$$\left.\left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-m}^j \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^j \right. \right] \right| \right\|_1 \longrightarrow 0 \qquad (8.37)$$

–

$$\sum_{t=-n+1}^{-\left[\frac{n}{2}\right]} \sum_{l=-\left[\frac{t}{2}\right]}^{0} \left\| \sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^j \right. \right] \right.\right.$$

$$\left.\left. - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \left| Y_{-n}^j \right. \right] E_\theta \left[\gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-n}^j \right. \right] \right| \right\|_1 \longrightarrow 0 \qquad (8.38)$$

pour $n, m \to \infty$.

Puisque nous avons obtenu les mêmes inégalités que Bickel et al. [10], la démonstration est totalement similaire à la leur en remplaçant dans leur preuve, pour tout $t \in \mathbb{Z}$, $\gamma_\theta \left(Y_t \left| X_t \right. \right)$ par $\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right)$

Par exemple, pour montrer (8.31), par le lemme 15, on a

$$\sup_{\theta \in V} \left| E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^1 \right. \right] - E_\theta \left[\gamma_\theta \left(Y_{t-1}, Y_t \left| X_t \right. \right) \gamma_\theta \left(Y_{l-1}, Y_l \left| X_l \right. \right) \left| Y_{-m}^0 \right. \right] \right|$$

$$\leq \sup_{\theta\in V}\sum_{i,j=1}^{N}\left\{\left|\gamma_\theta\left(Y_{t-1},Y_t\left|e_i\right.\right)\right|\left|\gamma_\theta\left(Y_{l-1},Y_l\left|e_j\right.\right)\right|\right.$$

$$\left.\left|P_\theta\left(X_t=e_i,X_l=e_j\left|Y_m^1\right.\right)-P_\theta\left(X_t=e_i,X_l=e_j\left|Y_m^0\right.\right)\right|\right\}$$

$$\leq C\left(\sup_{\theta\in V}\max_{i\in\{1,\cdots,N\}}\left|\gamma_\theta\left(Y_{t-1},Y_t\left|e_i\right.\right)\right|\right)\left(\sup_{\theta\in V}\max_{j\in\{1,\cdots,N\}}\left|\gamma_\theta\left(Y_{l-1},Y_l\left|e_j\right.\right)\right|\right)\prod_{i=t\vee l+1}^{0}\exp\left(-2\mu_0\left(Y_{i-1},Y_i\right)\right)$$

$$\leq C\beta^{|t|\wedge|l|}$$

pour un $\beta\in]0,1[$. Donc la partie gauche de (8.31) est bornée par

$$C\sum_{t=\left[\frac{n}{2}\right]}^{m-1}\sum_{l=\left[\frac{t}{2}\right]}^{m-1}\beta^l\leq C\sum_{t=\left[\frac{n}{2}\right]}^{m-1}\beta^{\left[\frac{t}{2}\right]}\leq C\beta^{\frac{n}{4}}.$$

L'adaptation de Bickel et al. [10] aux autres inégalités est tout aussi facile.

■

Ainsi $\left(H\ln L_\theta\left(Y_1\left|Y_{-n}^0\right.\right)\right)$ est une "suite de Cauchy uniforme" dans $\mathbb{L}_1\left(P_0\right)$ et le résultat suivant est immédiat

Lemme 18 *Il existe une fonction continue $\zeta_1(\theta)$ de V dans $\mathbb{L}_1(P_0)$ tel que*

$$\left\|\sup_{\theta\in V}\left|H\ln L_\theta\left(Y_1\left|Y_{-n}^0\right.\right)-\zeta_1\left(\theta\right)\right|\right\|\overset{n\to\infty}{\longrightarrow}0$$

On peut maintenant prouver le théorème :

Théorème 19 *Sous les hypothèses (**N**), soit $(\theta_n)_{n\in\mathbb{N}^*}$ une suite aléatoire dans Θ telle que $\theta_n\overset{p.s.}{\longrightarrow}\theta_0$ quand $n\to\infty$, alors*

$$n^{-1}H\ln L_{\theta_n}\left(Y_0^n\right)\overset{\mathbb{P}_0}{\longrightarrow}-\mathcal{J}_0$$

lorsque $n\to\infty$.

Preuve Soit $\zeta_t(\theta)$ la limite dans $\mathbb{L}_1(\mathbb{P}_0)$ de

$$H\ln L_\theta\left(Y_t\left|Y_{-n}^{t-1}\right.\right)$$

et soit $V'\subseteq V$ un voisinage de θ_0. On a

$$\limsup_{n\to\infty}P_0\left(\left|n^{-1}H\ln L_{\theta_n}\left(Y_0^n\right)-n^{-1}\sum_{t=1}^{n}\zeta_t\left(\theta_0\right)\right|>\varepsilon\right)$$

$$= \limsup_{n\to\infty} P_0\left(\left|n^{-1}\sum_{t=1}^{n}\left\{H\ln L_{\theta_n}\left(Y_t\left|Y_0^{t-1}\right.\right)-\zeta_t\left(\theta_0\right)\right\}\right|>\varepsilon\right)$$

$$\leq \limsup_{n\to\infty} P_0\left(n^{-1}\sum_{t=1}^{n}\sup_{\theta\in V'}\left|\left\{H\ln L_\theta\left(Y_t\left|Y_0^{t-1}\right.\right)-\zeta_t\left(\theta_0\right)\right\}\right|>\varepsilon\right)+\limsup_{n\to\infty} P_0\left(\theta_n\notin V'\right)$$

et par l'inégalité de Markov et la stationnarité de $(Y_t)_{t\in\mathbb{Z}}$,

$$\leq \limsup_{n\to\infty} n^{-1}\varepsilon^{-1}\sum_{t=1}^{n}\left\|\sup_{\theta\in V'}\left|H\ln L_\theta\left(Y_1\left|Y_{-t+1}^{0}\right.\right)-\zeta_1\left(\theta_0\right)\right|\right\|_1$$

$$\leq \limsup_{n\to\infty} n^{-1}\varepsilon^{-1}\sum_{t=1}^{n}\left\|\sup_{\theta\in V'}\left|H\ln L_\theta\left(Y_1\left|Y_{-t+1}^{0}\right.\right)-\zeta_1\left(\theta\right)\right|\right\|_1$$

$$+\limsup_{n\to\infty} n^{-1}\varepsilon^{-1}\sum_{t=1}^{n}\left\|\sup_{\theta\in V'}\left|\zeta_1\left(\theta\right)-\zeta_1\left(\theta_0\right)\right|\right\|_1$$

$$= \varepsilon^{-1}\left\|\sup_{\theta\in V'}\left|\zeta_1\left(\theta\right)-\zeta_1\left(\theta_0\right)\right|\right\|_1,$$

où la dernière égalité est une conséquence du lemme 18. Soit $(V_n')_{n\in\mathbb{N}^*}$, $V_n'\subseteq V$ décroissant vers $\{\theta_0\}$, la continuité de $\zeta(.)$ implique que

$$n^{-1}H\ln L_{\theta_n}\left(Y_0^n\right)-n^{-1}\sum_{t=1}^{n}\zeta_t\left(\theta_0\right)\xrightarrow{\mathbb{P}_0}0$$

lorsque $n\to\infty$.

Maintenant, comme $(Y_t)_{t\in\mathbb{Z}}$ est ergodique et donc $(\zeta_t)_{t\in\mathbb{N}}$ aussi, on a

$$n^{-1}\sum_{t=1}^{n}\zeta_t\left(\theta_0\right)\xrightarrow{p.s.}\mathcal{J}$$

pour une matrice $\mathcal{J}=E_0\zeta_1\left(\theta_0\right)$. Cependant la condition (**N**)-3c implique que

$$E_0\left[-H\ln\phi_i^{\theta_0}\left(Y_1-F_{(W_0)_i}\left(y_0\right)\right)\right]=E_0\left[\left(\nabla\ln\phi_i^{\theta_0}\left(Y_1-F_{(W_0)_i}\left(y_0\right)\right)\right)^2\right]$$

donc

$$E_0\left[H\ln L_{\theta_0}\left(Y_1\left|Y_{-n}^{0}\right.\right)\right]=-E_0\left[\left(\nabla\ln L_{\theta_0}\left(Y_1\left|Y_{-n}^{0}\right.\right)\right)^2\right]$$

pour chaque n, donc $\mathcal{J}=-\mathcal{J}_0$ ■

On en déduit le théorème

Théorème 20 *Supposons les hypothèses (**N**) vérifiées et que $\mathcal{J}_0$ est inversible, alors*

$$n^{\frac{1}{2}}\left(\hat{\theta}_n-\theta_0\right)\xrightarrow{\mathcal{L}}\mathcal{N}\left(0,\mathcal{J}_0^{-1}\right)$$

lorsque $n\to\infty$.

Preuve Elle est classiquement basée sur la formule de Taylor. Comme pour n assez grand, on a $\hat{\theta}_n \in V$, en notant $l_n(\theta) := \ln L_\theta(y_0^n)$ on a

$$0 = \nabla l_n(\theta) = \nabla l_n(\theta_0) + Hl_n\left(\overline{\theta}_n\right)\left(\hat{\theta}_n - \theta_0\right)$$

ou $\overline{\theta}_n$ est un point du segment joignant θ_0 et $\hat{\theta}_n$. On obtient donc

$$n^{\frac{1}{2}}\left(\hat{\theta}_n - \theta_0\right) = \left[-n^{-1}Hl_n\left(\overline{\theta}_n\right)\right]^{-1} n^{-\frac{1}{2}}\nabla l_n(\theta_0).$$

Le théorème résulte alors des deux théorèmes précédents ■

8.3 Conclusion

Nous avons montré, sous des conditions aisément vérifiables, la consistance forte et la normalité asymptotique de l'estimateur du maximum de vraisemblance pour le modèle (8.1). On doit cependant noter que l'hypothèse d'identifiabilité suppose de connaître exactement le nombre d'états cachés du modèle. Nous touchons là le point "sensible" de ce modèle. En effet si on surestime le nombre d'états de la chaîne de Markov cachée, nous perdons l'identifiabilité et tout espoir d'avoir une matrice d'information de Fisher définie positive. Ainsi, il n'est pas possible d'obtenir, par les mêmes méthodes que le chapitre 4, un critère d'identification presque sûre du modèle. Il est à noter cependant que Francq, Roussignol et Zakoïan [29] ont montré pour un modèle proche du nôtre que minimiser un contraste pénalisé, avec un terme de pénalisation qui tend vers 0 lorsque le nombre d'observations augmente, ne sous-estime pas le vrai modèle.

Trouver le bon nombre d'états pour ce modèle reste donc un problème ouvert et fondamental pour pouvoir aller plus loin dans l'étude statistique de ce modèle.

Chapitre 9

Etude des pics de pollution en niveau d'ozone

9.1 Préambule

9.1.1 Situation actuelle

Aujourd'hui la pollution photochimique est devenue un problème crucial dans la plupart des cités modernes de grande taille. La croissance importante des principales sources d'émission (transport, centrales thermiques, chauffage urbain, etc...) alliée à des conditions météorologiques favorables conduisent à l'apparition dans ce milieu de concentrations atmosphériques de polluants supérieures aux normes de la qualité de l'air. C'est notamment le cas en France, des villes comme Paris, Lyon ou Strasbourg enregistrent pendant plusieurs jours dans l'année des concentrations dépassant les seuils d'alerte fixés par les pouvoirs publics. Face à cette situation, il devient nécessaire de pouvoir prévoir à l'avance et en temps réel les pics de pollution susceptibles de se produire.

Cet objectif peut s'atteindre en utilisant deux approches différentes

- Utiliser un modèle déterministe intégrant la modélisation des mécanismes d'évolution chimique des composés incriminés
- Entreprendre une modélisation par des méthodes statistiques, n'utilisant pas les équations mécanistiques précédentes.

9.1.1.1 Le modèle déterministe

L'utilisation des modèles déterministes dans la modélisation des processus photochimiques présente en général une difficulté importante qui tient aux limites de la

paramétrisation des processus de diffusion turbulente lorsque le vent est très faible. Cependant, les épisodes de pollution atmosphériques se rencontrent principalement dans ces conditions. Mais, dans des conditions géographiques particulières comme celles du tissu urbain, ces paramètres sont encore plus difficilement modélisables à l'heure actuelle. Lorsque l'on tente d'utiliser ces modèles déterministes pour une prédiction en temps réel, une autre difficulté surgit qui réside dans le temps de calcul nécessaire aux sorties du modèle et la nécessité d'avoir recours à des calculateurs très puissants. Néanmoins de tels modèles sont disponibles actuellement.

9.1.1.2 Le modèle statistique

Des études préliminaires [22], [13] ont prouvé que les variations des polluants de l'atmosphère, régies à la fois par la réactivité chimique et la dynamique de l'atmosphère, sont bien mieux expliquées par des modèles non-linéaires et plus particulièrement des réseaux de neurones. Un avantage des modèles statistiques est qu'une fois le modèle ajusté, les temps de réponse pour la prévisions des pics sont plusieurs centaines de fois inférieurs à ceux des modèles déterministes.

9.1.2 Mise en oeuvre de cette étude

Le but de cette étude est donc de modéliser les variations de concentration d'ozone en milieu urbain. Nous souhaitons reproduire la variabilité des concentrations d'ozone en fonction des paramètres chimiques et dynamiques du système. Ces prévisions sont basés sur les *observations* en temps réel. On pense en effet que cette modélisation est essentielle, pour pouvoir ensuite remplacer les *observations* météorologiques par les *prévisions* atmosphériques et construire ainsi un modèle prédictif final.

9.1.2.1 Les données disponibles

Les données de base de l'étude sont constituées, pour la chimie, des mesures enregistrées par le réseau de surveillance AIRPARIF et pour les données météorologiques observées, de celles des stations parisiennes de Météo-France. Il est important de posséder une base de données suffisamment grande qui contienne le plus de situations possible. La périodicité des variations des concentrations d'ozone, ainsi que la faible fréquence d'occurrence des pics de pollution sur l'ensemble de l'année, nous impose de travailler sur une base pluriannuelle la plus large possible. Les données retenus seront ici les moyennes horaires des concentrations des polluants enregistrées de 1994 à 1997.

9.1.2.2 Les différentes modélisations

Dans un premier temps, nous travaillerons sur les moyennes journalières des données et nous essaierons de prévoir le maximum du taux d'ozone de la journée. Le

principal inconvénient de cette méthode est le manque de données (moins de 1000).

Nous étudions ensuite cette série en utilisant toutes les données (i.e. les moyennes horaires) et nous essaierons de prévoir l'ozone 24 heures plus tard. Le grand nombre de données disponibles (environ 25000) nous permet de plus d'ajuster un modèle intégrant des changements de régimes modélisés par une chaîne de Markov cachée.

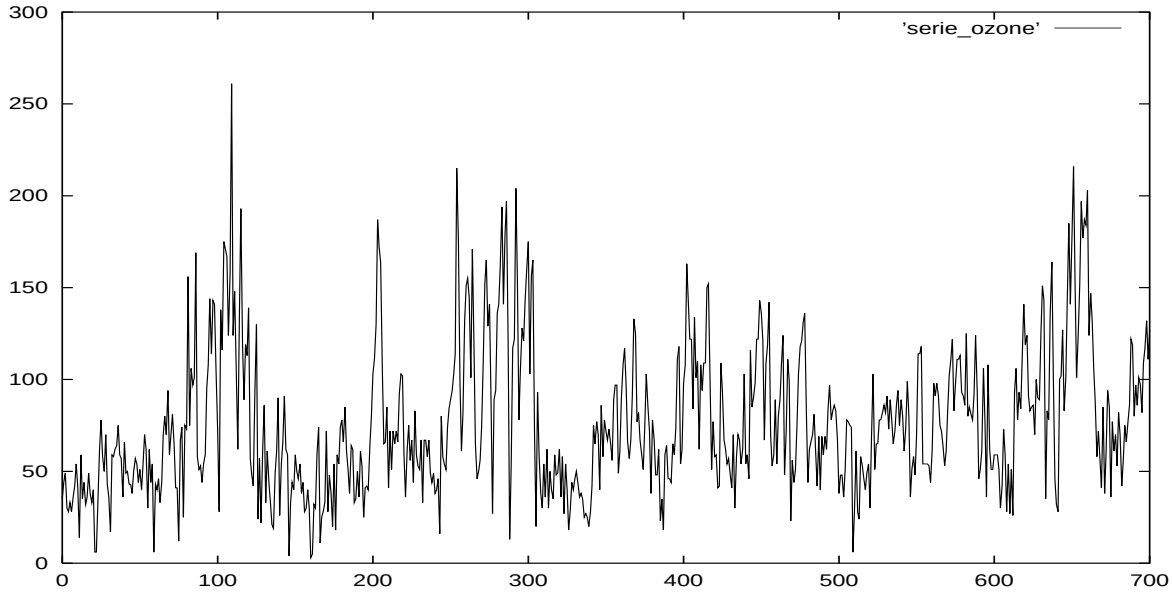
9.2 Etude sur les moyennes journalières

9.2.1 Les données

9.2.1.1 Les données à expliquer

La série regroupe la période du 1 avril au 30 septembre sur les années 1994 jusqu'à 1997. Il y a 701 observations du maximum horaire du taux de l'ozone au cours de la journée. Pour comparer nos résultats avec ceux du LISA à l'université de Créteil (laboratoire qui nous fournit les données), nous estimons notre modèle sur les 550 premières données. Après l'estimation et l'identification de notre modèle, on valide le modèle sur les 151 données restantes. Nous travaillons sur les données centrées et normées. En effet, bien que cette précaution ne soit pas utile en théorie, elle se révèle absolument nécessaire dans la pratique, sinon on est confronté à des problèmes numériques issus essentiellement des phénomènes de saturation des fonctions d'activation sigmoïdales des unités cachées.

FIG. 9.1 – La série des maximums du taux d'ozone au cours de la journée



9.2.1.2 Les données explicatives

Ces données ont été choisies par A. Dutot, chimiste au LISA. On donne ici leurs descriptions suivies du symbole qui les représente sur le graphiques qui suivent.

- Le maximum horaire de l'ozone au cours du jour $t-1$ (l'unité de temps est ici le "jour") : $MAXO3(t-1)$.
- Le rayonnement global : RGB.
- La vitesse moyenne du vent : $VMVS$.
- La température maximum de la journée : $TEMPMAX$.
- Le gradient de température sur 1, 2 et 3 jours. : $GT1$, $GT2$ et $GT3$.
- La fréquence de la direction du vent suivant les intervalles d'angles (en degrés) $[0,45]$, $[45,90]$, $[90,135]$, $[135,180]$, $[180,225]$, $[225,270]$, $[270,315]$, $[315,360]$: $FV0-45$, $FV45-90$, $FV90-135$, $FV135-180$, $FV180-225$, $FV225-270$, $FV270-315$ et $FV315-360$.
- Le taux de NO_x (No_x est la somme du monoxyde et du dioxyde d'azote) : NOx.

Il faut prévoir le maximum d'ozone pour le jour t : $MAXO3(t)$.

9.2.2 Etudes préliminaires

9.2.2.1 Le modèle linéaire

La première chose à faire est d'estimer cette série par un modèle linéaire, pour vérifier qu'un modèle non-linéaire du type MLP améliore les estimations. Les erreurs spécifiées correspondent à l'écart-type (la racine carré de l'erreur quadratique moyenne). En estimant la série avec un modèle linéaire (ARX 16) on obtient une erreur de 19.65 sur la base d'apprentissage (les 550 premières données) et de 19.3 sur la base de validation (les 151 dernières données). Le découpage des données n'est pas réellement utile dans le cas linéaire, il ne s'agit ici que de pouvoir comparer les résultats avec ceux du MLP.

9.2.2.2 Estimation par l'algorithme SSM (cf section 2.2.3)

Nous estimons notre modèle grâce au programme REGRESS (cf annexe A) avec la recherche automatique d'architecture. En effet, comme le nombre d'entrées est grand par rapport au nombre d'observations on va ainsi essayer d'enlever les entrées non pertinentes, ce qui permettra de réduire le nombre de paramètres.

La procédure d'identification est la suivante

- On fait 20 initialisations aléatoires pour chaque estimation préliminaire de recherche de modèle dominant, les architectures dominantes auront alors k et $k+1$ unités cachées (cf A.1.2.1 et A.1.2.3).

– On garde les 5 meilleurs MLP dominants, pour chaque architecture avec k et $k+1$ unités sur la couche cachée et on les élague avec l'algorithme SSM. On obtient ainsi 10 MLP élagués.

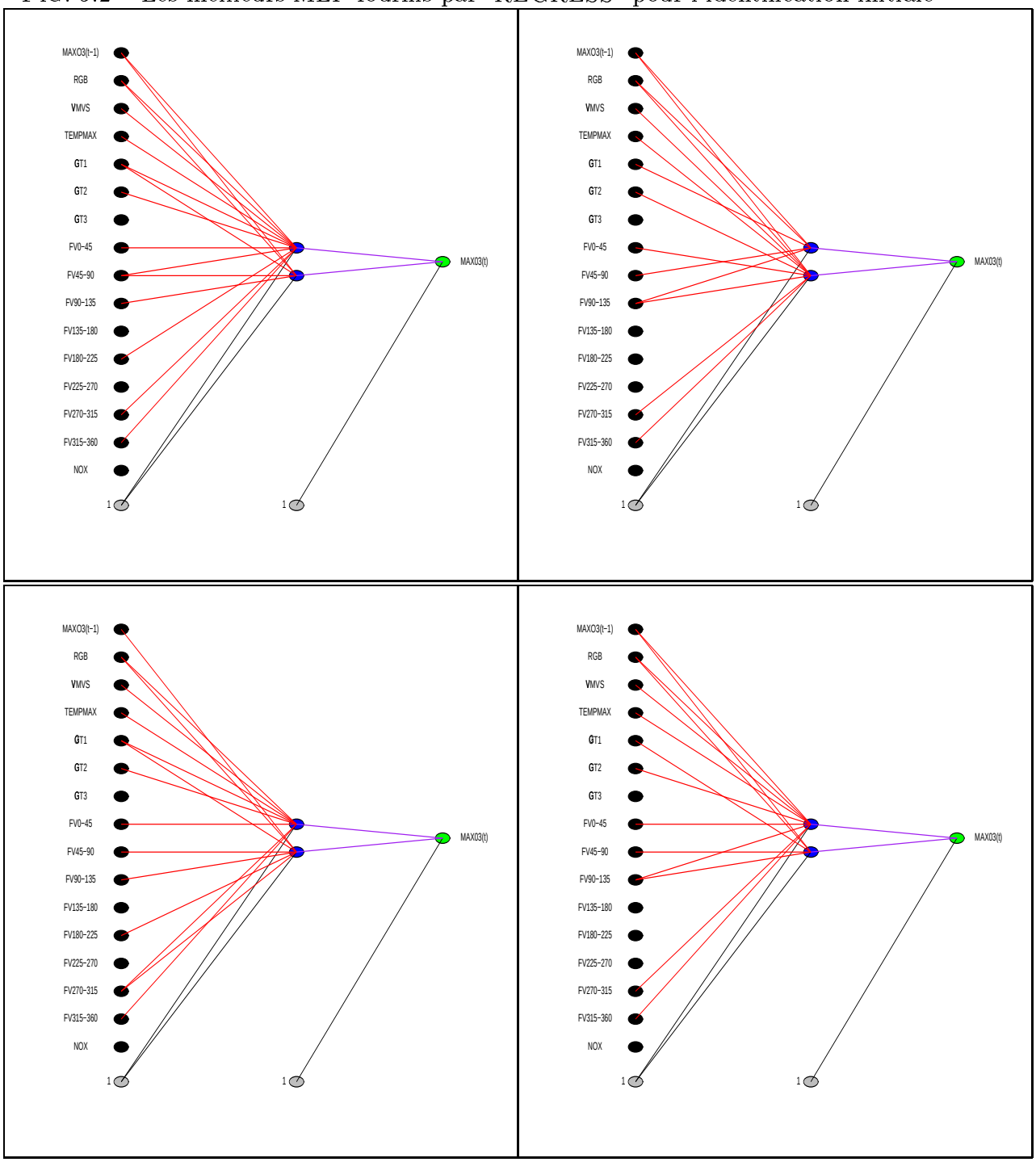
La figure 9.2 représente l'architecture des 4 meilleurs MLP élagués, parmi les 10 obtenus.

Interprétation des résultats Après cette première procédure d'identification sur les 550 premières données, les meilleurs MLP n'ont que deux unités sur la couche cachée. Le très faible nombre d'unités cachées s'explique par le fait du trop grand nombre d'entrées et du trop petit nombre de données.

Il est important de remarquer que le retard du gradient de température à $t-3$, la fréquence du vent pour les angles $[225, 270]$ et $[135, 180]$ ainsi que NOx ne sont pas explicatifs pour ce modèle puisqu'ils n'ont pas été retenus comme explicatifs par l'algorithme pour les 4 meilleurs modèles. On va donc recommencer cette étude en éliminant dés le début ces entrées inutiles.

La meilleure erreur sur la base de validation (exemples $551-701$) est de 18.48 alors que l'erreur sur la base d'apprentissage $(1-550)$ était de 14.63. Ce comportement de l'erreur (bien plus petite sur la base d'apprentissage que de validation) est typique d'un phénomène de "surapprentissage". On peut quand même remarquer que le modèle neuronal améliore déjà le modèle linéaire.

FIG. 9.2 – Les meilleurs MLP fournis par "REGRESS" pour l'identification initiale



9.2.3 Etude finale

On va recommencer l'identification sur la série où les variables non explicatives sont enlevées. Il reste ainsi 12 variables explicatives. Cette nouvelle estimation permet un identification plus précise du modèle, car il y aura moins de paramètres de nuisance au début de l'apprentissage.

9.2.3.1 Les résultats de la procédure

On lance de nouveau la procédure de recherche automatique de modèles avec la même procédure d'apprentissage que précédemment. Le fait d'enlever ces entrées, nous permet de faire une estimation avec des MLP ayant jusqu'à 4 unités sur la couche cachées en évitant au possible le surapprentissage. Les meilleurs MLP sont représentés figure 9.3. On remarque que les entrées sont toute utiles, sauf pour un modèle qui est d'ailleurs le meilleur.

9.2.3.2 Performances du meilleur modèle

Le meilleur MLP (celui qui a le meilleur BIC (cf section 2.2.3) sur la base d'apprentissage) a maintenant une erreur sur la base de validation de 17.80, il a 4 unités cachées et 30 paramètres. Son erreur sur la base d'apprentissage est de 15.48, il reste donc un peu de surapprentissage même après élagage. Il y a trop peu de données et le critère d'information qui n'est justifié qu'asymptotiquement ne suffit pas. Néanmoins, les résultats sont meilleurs que précédemment et meilleurs que le modèle linéaire. La figure 9.4 représentent l'architecture du meilleur MLP. La figure montre ses prévisions sur la base de validation. On peut remarquer que si les valeurs moyennes de la série sont bien prévues, ce n'est pas le cas pour les pics. Comme on cherchait essentiellement à prévoir les pics, ce modèle n'est pas totalement satisfaisant.

FIG. 9.3 – Les meilleurs MLP fournit par "REGRESS" pour l'identification finale

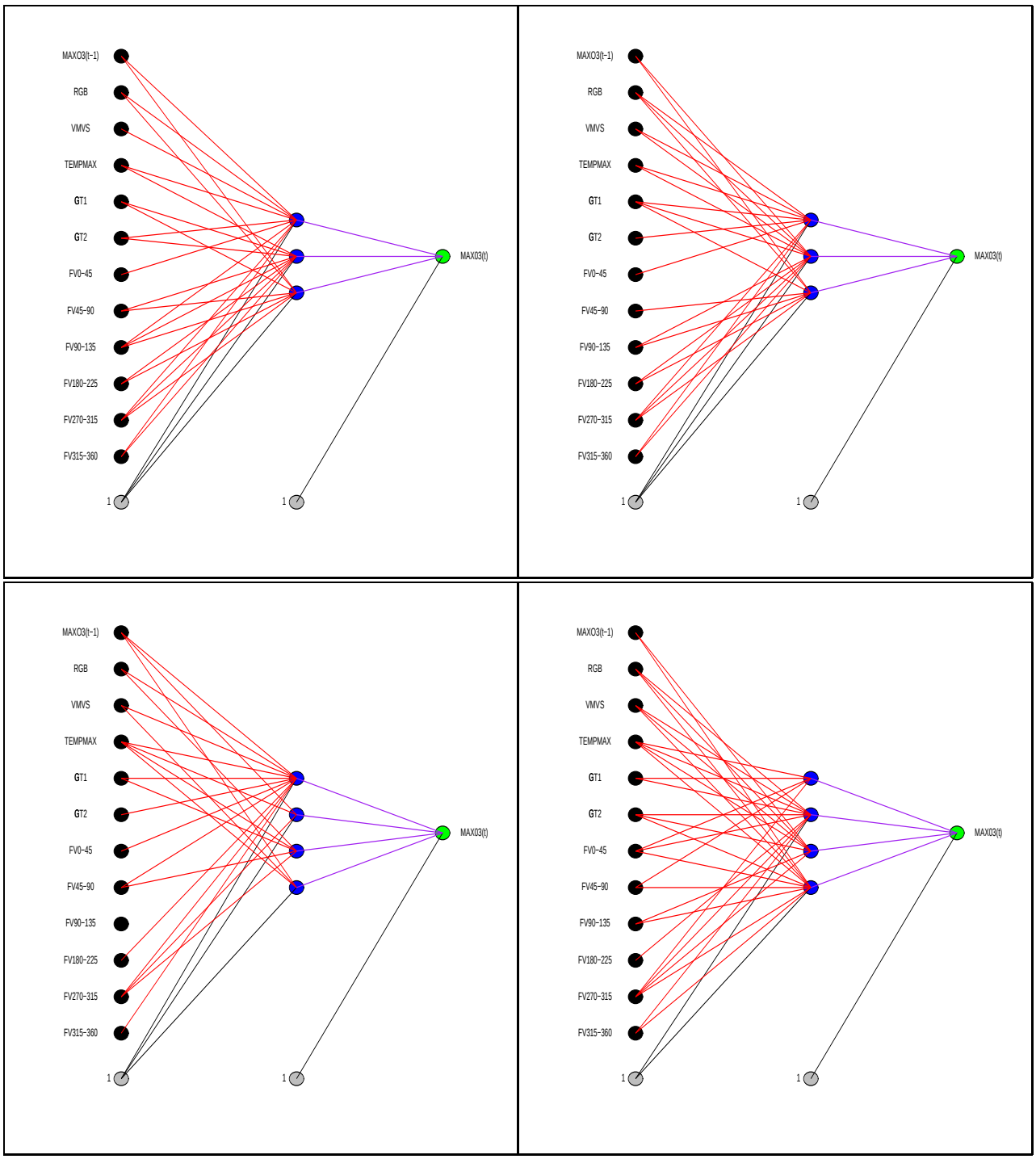
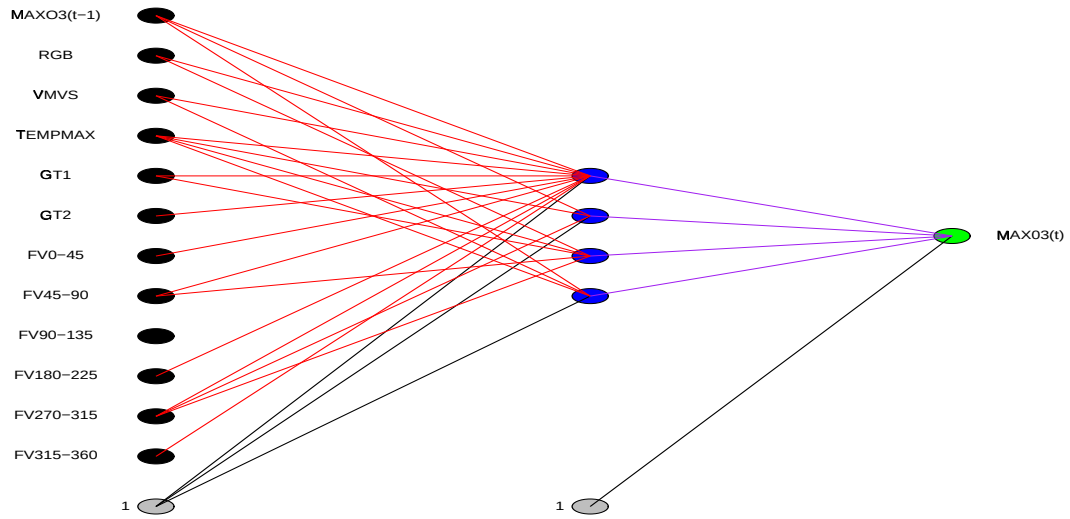
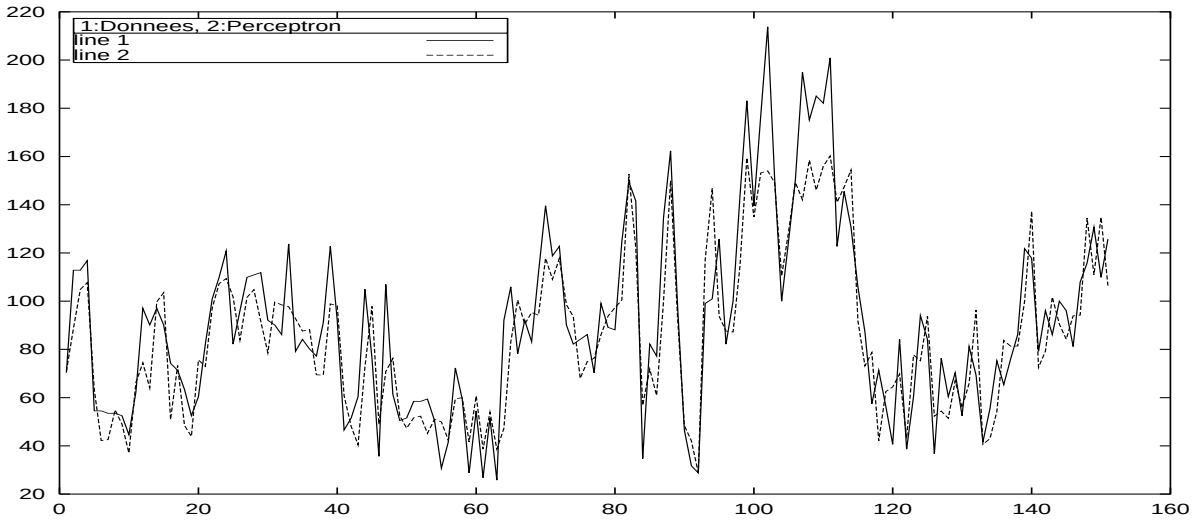


FIG. 9.4 – Meilleur MLP, et ses prévisions sur l'ensemble de validation.





9.2.4 Conclusion de cette étude

Après comparaison avec les résultats du LISA, il apparaît que les données n'étaient pas adaptées au problème. En effet le LISA obtient des résultats quasi identiques avec une méthode différente. Leur modèle est bien plus complexe puisqu'il s'agit d'un MLP avec 16 entrées, deux couches cachées de, respectivement, 5 et 2 unités cachées (ce qui correspond à 100 paramètres), alors que notre meilleur modèle (cf figure 9.4) a seulement 12 entrées, une couche cachée de 4 unités et est élagué (ce qui correspond à 30 paramètres). Ils ont utilisé la méthode du "early stopping" pour éviter le surapprentissage et une sur-pondération des pics pour essayer de modéliser les grands pics. Ainsi, ils doivent donc régler plusieurs paramètres (découpage de la base d'apprentissage pour le "early stopping", stratégie de sur-pondération des pics) qui dépendent totalement de la série étudiée pour mettre au point leur modèle. Leur méthode est donc moins "universelle" que la nôtre. Hélas, notre modèle, bien que plus parcimonieux, ne prévoit pas mieux les pics de pollutions, c'est pourquoi, d'un commun accord avec le LISA, nous avons recommencé une étude avec de nouvelles données, les données horaires.

9.3 Etude sur les moyennes horaires

9.3.1 Description des données

Ces données, fournies par le LISA, ont encore été sélectionnées par A. Dutot. On dispose des données heure par heure de 1994 à 1997 pour les diverses variables. Il y a des heures et des jours où certains capteurs n'ont pas fonctionné, le nombres de données non consécutives est de 623 pour 30677 observations.

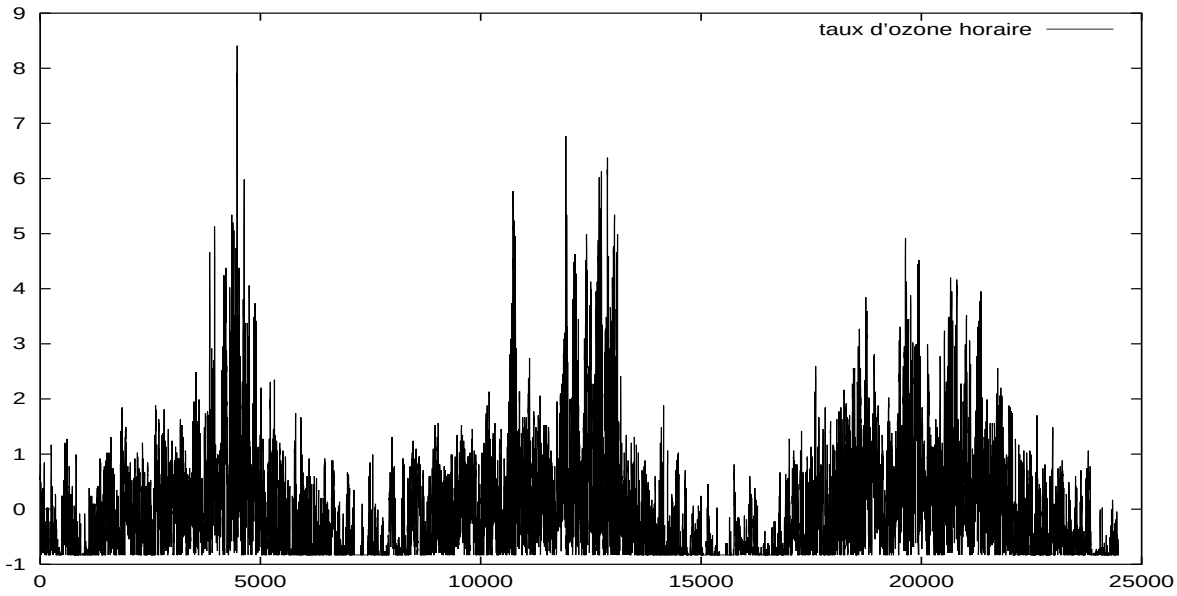
Afin de pouvoir comparer nos résultats avec ceux du LISA, nous adoptons le même découpage que le leur. Ainsi, dans toutes les estimations, on utilise les données de 1994-1996 pour estimer le modèle (24474 données dont 559 non consécutives), une fois l'apprentissage terminé, on validera les modèles sur les données de 1997 (6203 données dont 64 non consécutives). Le nombre de données est donc bien plus grand, et les résultats basés sur des propriétés asymptotiques devraient donc être meilleurs. Cependant, cette nouvelle estimation basée sur les données horaires et non plus leur moyenne journalière nous oblige à faire une prévision à $t + 24$, ce qui complexifie la tâche. L'écart-type de la variable du taux d'ozone ($O3(t)$) de la série utilisée pour estimer le modèle est de 28.03, celle de la série pour valider le modèle estimé est de 33.72. On centre et normalise les données des deux séries séparément. Les variables explicatives du taux d'ozone au temps t (en heure) sont :

- L'heure en temps universel.
- Le sinus du jour de l'année.
- Le cosinus du jour de l'année.
- Une indicatrice du Week end (1 si jour du week-end, 0 sinon).

- Le sinus de la direction du vent en degré géographique.
- Le cosinus de la direction du vent en degré géographique.
- La vitesse du vent.
- L'humidité relative.
- Le rayonnement global.
- La température de l'air.
- La pression atmosphérique.
- La concentration en NO_x (la somme du monoxide et du dioxyde d'azote).
- La concentration d'ozone 24 heures avant $(O3\,(t-24))$.

La figure 9.5 montre la série dans son entier, on utilise ici toutes les données.

FIG. 9.5 – La série (centrée normée) des moyennes horaires de 1994 jusqu'à 1997



9.3.2 Etude préliminaire

Nous commençons cette étude par l'estimation et l'identification d'un modèle MLP simple. On espère ainsi détecter d'éventuelles variables d'entrées inutiles. On utilise le programme "Regress" et sa recherche automatique d'architecture pour estimer et identifier un modèle MLP selon la même procédure que section 9.2.2.2.

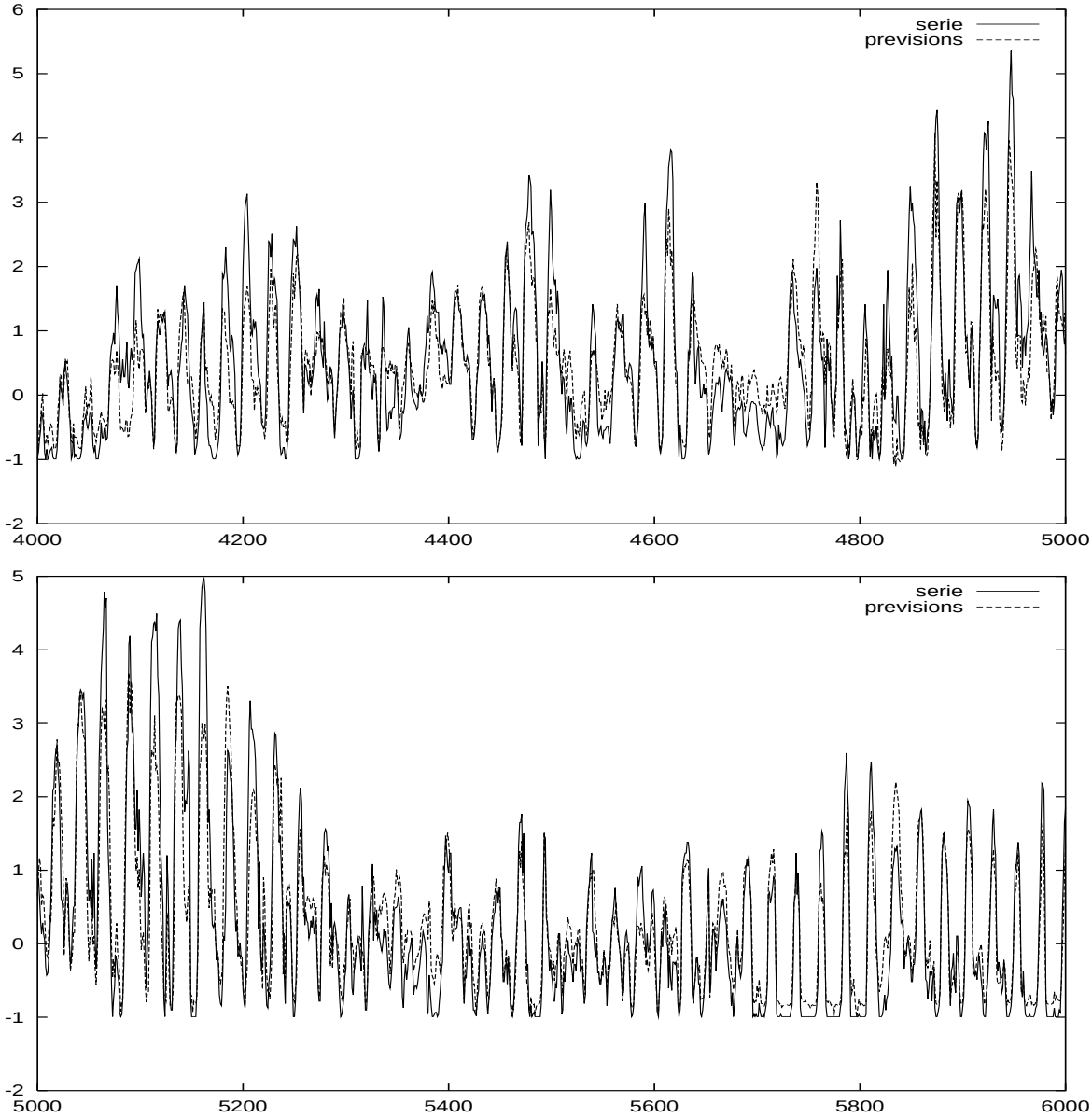
Le meilleur MLP obtenu a une couche cachée de 11 unités. Toutes les entrées (13) sont largement connectées, il semble donc que toutes les variables explicatives soient "utiles". Le nombre de paramètres du modèle est de 129. Le MLP obtient les résultats résumés dans le tableau 9.1. On rappelle que les données de 1994 jusqu'à 1996 ont servi à l'estimation, alors que celle de 1997 n'ont jamais été vues par le modèle.

On peut remarquer que l'erreur sur les données de 1997 est légèrement plus élevée que sur les données 1994-1996. Pourtant, le nombre important d'observations pour l'estimation devrait éviter les phénomènes de surapprentissage. C'est certainement un indice du changement de dynamique de la série. La figure 9.6 montre les prévisions de ce MLP sur la série de validation sur les données $4000 - 5000$ et $5000 - 6000$. Il s'agit là d'une des parties de la série où il y a le plus de pics. On peut encore remarquer que ce modèle a du mal à prévoir les pics de pollution même si il est assez bon pour prévoir les valeurs moyennes.

TAB. 9.1 – Erreur et écart-type du MLP sur les bases.

Résultats du MLP	1994-1996	1997
variance sur série centrée normée	0.12	0.164
écart-type sur la série brute	9.71	13.66

FIG. 9.6 – Prévision sur la période 1997, données 4000-5000 et 5000-6000.



9.3.3 Estimation par modèle hybride HMM/MLP

9.3.3.1 Motivations

Il est difficile de dire, en regardant la série, si la dynamique change au cours du temps. En effet les données exogènes peuvent expliquer à elles seules les périodicités de la série. Nous allons cependant ajuster un modèle à changements de régime markoviens, car ce modèle doit en principe détecter d'éventuels changements.

9.3.3.2 Le modèle

On utilise un modèle avec deux experts. On suppose donc ici, qu'il y a au moins 2 régimes, qui vont être modélisés par ces experts. L'innovation du modèle est supposée normale, on pourra ainsi appliquer les algorithmes du chapitre 7. Nous avons vu lors de l'étude théorique du modèle que l'on est un peu désarmé pour trouver une bonne architecture. Nous ne pouvons pas automatiser la recherche d'architecture comme avec le programme REGRESS. De plus, le modèle avec une chaîne de Markov cachée demande en principe de respecter la chronologie des données. Cependant comme il y a seulement 2% de données non consécutives, on considère que la chronologie du modèle est à peu près respectée.

9.3.3.3 La procédure d'estimation

Nous avons estimé le modèle de la façon suivante :

- On maximise la log-vraisemblance par un algorithme du second ordre (BFGS), ce qui est possible grâce au calcul du gradient du chapitre 7. Pour éviter le surapprentissage, on garde le modèle qui obtient la meilleure prévision au sens des moindres carrés. Il n'y a pas de justification théorique à cette méthodologie, mais c'est celle qui fournit les meilleurs résultats.
- On estimera des modèles avec de 3 jusqu'à 10 unités sur la couche cachée. Les deux MLP "experts" ont le même nombre d'unités cachées. Cela permet de limiter le nombre d'estimations, car une estimation prend plus d'une journée de calcul sur PC.

9.3.3.4 Les résultats du meilleur modèle

Le meilleur modèle est constitué de 2 MLP avec chacun 6 unités sur la couche cachée. La matrice de transition estimée $\hat{A}$ pour la chaîne de Markov cachée est

$$\hat{A} = \begin{pmatrix} 0.93 & 0.03 \\ 0.07 & 0.97 \end{pmatrix}$$

Les variances associées à chaque état sont

$$\begin{cases} \hat{\sigma}_1 = 0.08 \\ \hat{\sigma}_2 = 0.04 \end{cases}$$

Les résultats sur les deux séries (apprentissage et validation) sont résumés tableau 9.2.

TAB. 9.2 – Erreur et écart-type du modèle hybride sur les bases.

Résultats du modèle hybride	1994-1996	1997
variance sur série centrée normée	0.078	0.089
écart-type sur la série brute	7.83	10.06

Si on compare ces résultat avec ceux du MLP seul (cf tableau 9.1), le modèle hybride, HMM/MLP, améliore l'erreur de prévision de façon significative et ce malgré les données manquantes. En effet celle-ci est pratiquement divisée par deux sur la série d'apprentissage et celle de validation. On notera que le nombre de paramètres n'est pas beaucoup plus élevé que pour le modèle avec un seul MLP (188 paramètres ici, contre 129 pour le MLP simple).

9.3.3.5 La prévision des pics

Les figures 9.7 et 9.8 montrent les prévisions de ce modèle sur la série de validation sur les données $4000-5000$ et $5000-6000$ ainsi que la probabilité conditionnelle de l'état 1 sachant les toutes les données. On remarque que pour un taux d'ozone relativement faible, la probabilité de l'état 1 est faible (donc celle de l'état 2 est forte), par contre c'est l'inverse pour les grandes valeurs. Il semble donc bien que la série change de comportement au cours du temps.

FIG. 9.7 – Série, prévisions sur 1997 (données 4000-5000) et probabilités conditionnelles de l'état 1

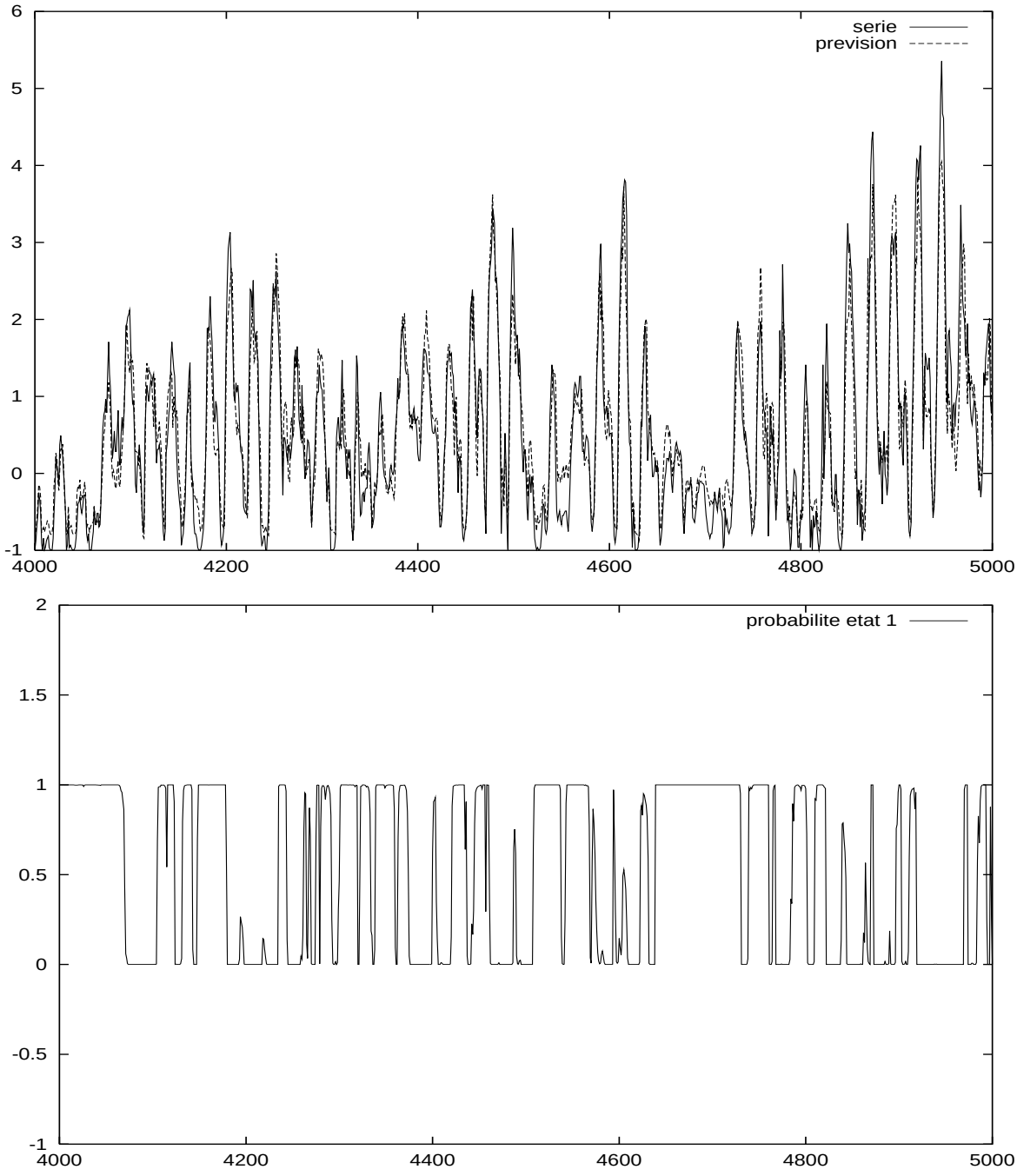
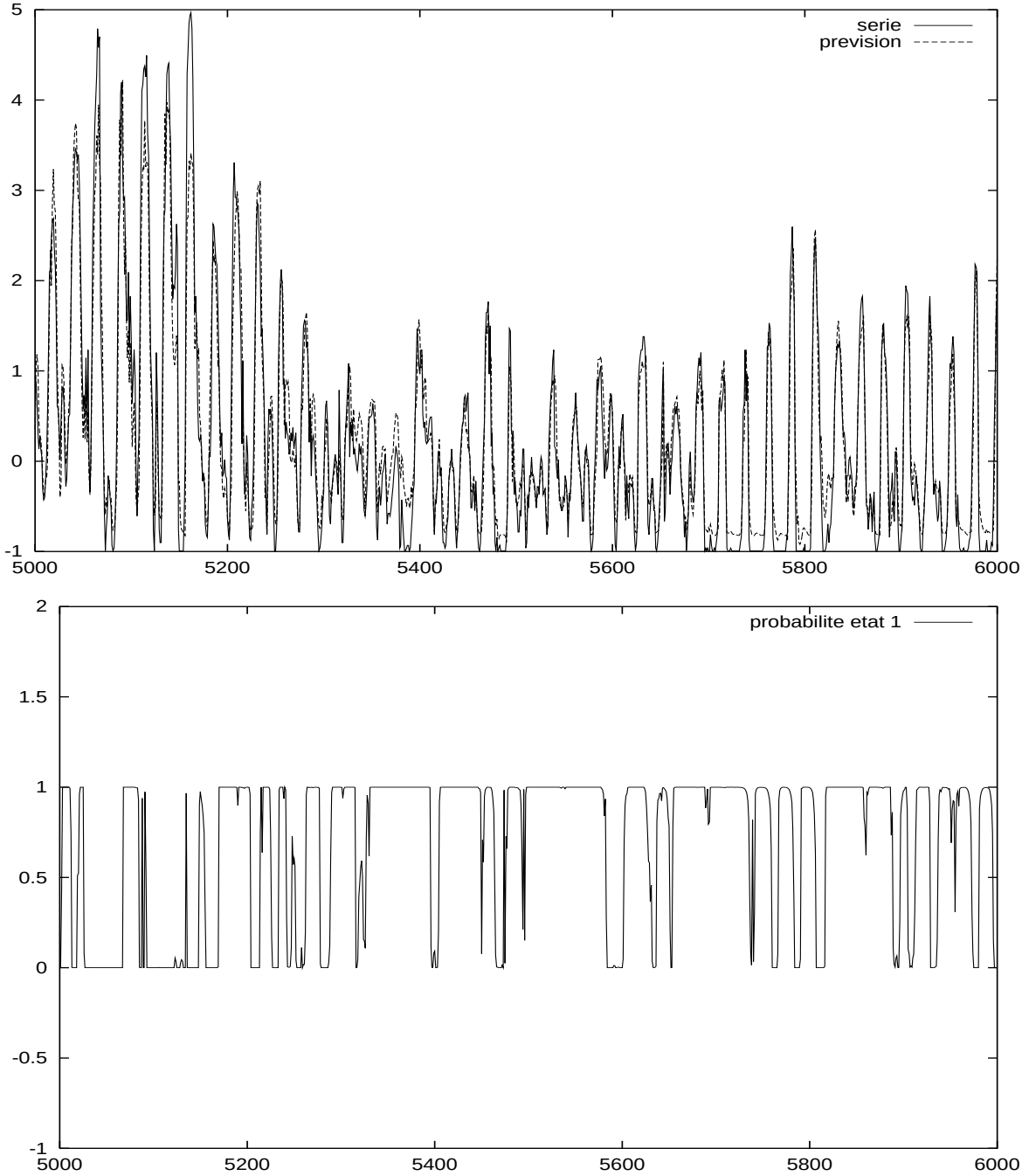


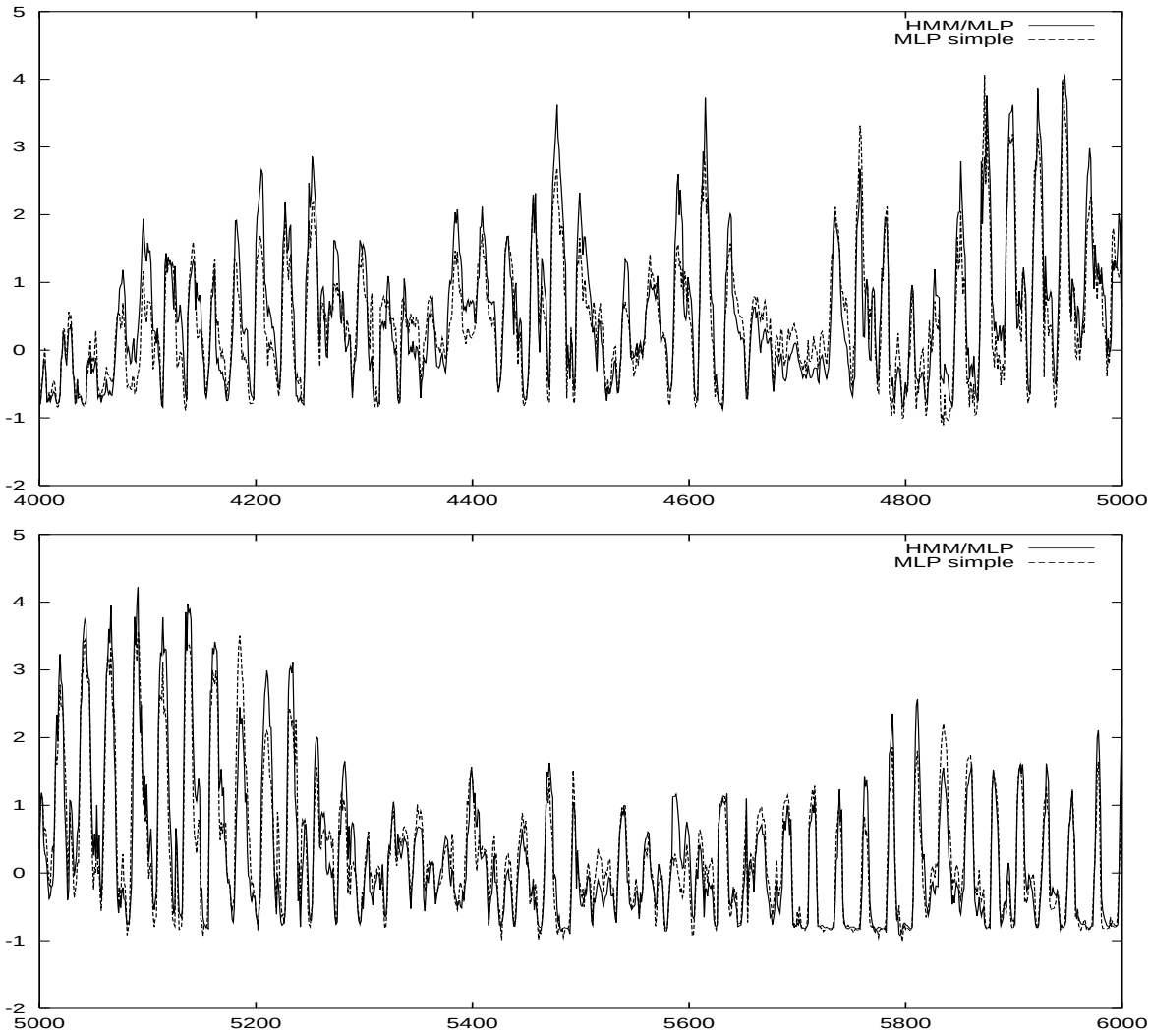
FIG. 9.8 – Série, prévisions sur 1997 (données 5000-6000) et probabilités conditionnelles de l'état 1



9.3.3.6 Comparaison de la prévision des pics des deux modèles

La figure 9.9 compare les prévisions du MLP simple et celle du modèle hybride HMM/MLP. Le modèle avec changements de régime est meilleur sur une très grande majorité de pics.

FIG. 9.9 – prévision sur 1997 (données 4000-5000 et 5000-6000) du modèle hybride et du MLP simple



9.3.4 Conclusion

Plusieurs avantages sont apparus lors des tentatives de modélisation de la série des taux d'ozone.

- Les performances peuvent être supérieures aux modèles qui ne modélisent pas les changements de régime.
- Une interprétation des relations entre les données explicatives et expliquées par les probabilités des régimes.
- Des possibilités de fournir des intervalles de confiance plus précis (par la variance associée à chaque régime).

L'étude de cette série continue. Deux voies apparaissent pour améliorer ces résultats :

- Remplacer les données manquantes par des prévisions.
- Coupler les prévisions sur différents sites de pollutions et utiliser les outils développés pour étudier des séries multidimensionnelles.

Chapitre 10

Conclusions et perspectives

Cette thèse est organisée autour de la prévision des séries temporelles par des modèles paramétriques non-linéaires et apporte une réponse à la modélisation des séries stationnaires par morceaux. On s'est restreint au cas des modèles de perceptrons multicouches, mais de nombreuses parties sont généralisables à tout autre modèle paramétrique non-linéaire. Une attention particulière est apportée aux problèmes numériques et algorithmiques, car ceux-ci sont fondamentaux pour une utilisation pratique de ces modèles. On étudie par exemple les apports du recuit simulé pour l'initialisation d'algorithme d'optimisation différentielle. Hélas, cette méthode devra certainement attendre une grande amélioration de la vitesse de calcul des ordinateurs, ce qui ne veux pas forcément dire longtemps, à la vitesse à laquelle les technologies évoluent.

Il est important de remarquer que rien n'est plus pratique qu'une bonne théorie, c'est pourquoi on précise les conditions assurant certaines propriétés asymptotiques utiles dans les étapes d'estimation et d'identification des modèles autorégressifs paramétriques non-linéaires. On justifie notamment l'utilisation du critère BIC pour identifier le modèle, et on améliore l'estimateur des moindres carrés dans le cas multidimensionnel.

Le passage à des modèles intégrant des chaînes de Markov cachées ajoute de nouvelles difficultés, tant numériques que théoriques. Nous avons étudié particulièrement les différentes façons de calculer les quantités utiles de ces modèles. Dans un premier temps, nous avons introduit des estimateurs utiles pour appliquer l'algorithme E.M. qui, grâce à sa simplicité, est très populaire pour l'estimation de ce genre de modèle. Enfin, pour améliorer la vitesse de traitement de ces modèles, nous avons développé des algorithmes plus rapides qui nous ont permis de travailler sur une série réelle regroupant plusieurs dizaines de milliers d'observations.

Au delà de ces problèmes numériques, on précise les conditions assurant certaines propriétés asymptotiques utiles dans les étapes d'estimation des modèles autorégressifs non-linéaires à changements de régime markoviens. Nous avons mis en évidence les problèmes théoriques qui restaient à résoudre et notamment le problème épineux du choix du nombre de régimes.

Enfin nous avons appliqué les outils développés dans ce document à la prévision des pics de pollution en niveau d'ozone. L'introduction d'une chaîne de Markov cachée améliore sensiblement le pouvoir prédictif du modèle et fournit des informations sur les différences de comportement de la série au cours du temps.

A travers ces travaux, plusieurs aspects inhérents à la prévision de séries temporelles sont apparues

1. La maîtrise de la complexité du modèle qui assure de bonne capacité de généralisation
2. La mise en évidence de régimes par l'introduction d'une chaîne de Markov cachée.

Il est à noter que la plupart des idées de ce document peuvent être généralisées immédiatement de deux façons :

1. Remplacer les MLP par d'autres fonctions non-linéaires paramétriques
2. Remplacer l'hypothèse de normalité du bruit par une autre densité pour les modèles intégrant des chaînes de Markov cachées.

Par contre, il est apparu une difficulté nouvelle, spécifique aux chaînes de Markov cachées, qui est la détermination du nombre d'états à prendre en considération. Les outils classiques sont impuissants pour traiter ce genre de problème, car si on surestime le nombre d'états cachés, le modèle n'est plus identifiable et l'on perd tous les résultats supposant que l'information de Fischer du modèle est définie positive. Ce problème reste donc ouvert. Néanmoins, des avancées sur les problèmes de mélanges de lois (cf Dacunha-Castelle et Gassiat [20]), qui peuvent être vus comme un cas particulier de chaînes de Markov cachées, donne une voie pour traiter ce problème. En effet, ces auteurs traitent des test pour des modèles non-identifiables, mais l'adaptation de leur méthode aux chaînes de Markov cachées n'est pas encore totalement établie.

Ce document étudie aussi les formes d'estimations récursives, comme le gradient stochastique. Nous nous sommes restreints à montrer l'intérêt numérique de ce genre de méthode sans pour autant en étudier les propriétés théoriques. Cette étude, bien que certainement calculatoire, ne semble pas hors de portée. Mevel [50] étudie par exemple, dans un cas particulier, les propriétés de ces algorithmes. On doit cependant noter que ses hypothèses sont très fortes et non vérifiées même dans des cas relativement simples.

Une généralisation importante des modèles étudiés est de considérer un espace d'états cachés non fini. On pourra par exemple étudier le cas où il est compact avant de passer à un ensemble général comme les ensembles polonais. Cela conduirait tout naturellement aux modèles avec des MLP récurrents qui utilisent les erreurs de prévision passées pour prévoir les valeurs venir. Ces modèles bien que déjà largement utilisés dans la pratique, manquent en effet cruellement d'outils théoriques qui pourrait guider l'utilisateur notamment dans son choix d'architecture.

Table des figures

Liste des tableaux

Bibliographie

[1] S. I. Amari. Information geometry of the E.M. and e.m. algorithms for neural networks. *Neural networks*, 8 :9 :1379–1408, 1995.

[2] S. I. Amari, H. Park, and K. Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation, à paraître*, 1999.

[3] J.K. Baker. The dragon system : An overview. *IEEE Trans. Acoust. Speech Signal Processing*, 23 :1 :24–29, 1975.

[4] A.R. Barron. Universal aproximation bounds for superpositions of a sigmoid functions. *IEEE Transaction on Information Theory*, 39 :3 :930–945, 1993.

[5] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probalistic functions of Markov processes. *Inequalities*, 3 :1–8, 1972.

[6] L.E. Baum and A. Egon. An inequality with applications to statistical estimation for probalistic functions of a Markov process and to a model for ecology. *Bull. Amer. Meteorol. Soc.*, 73 :360–363, 1967.

[7] L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite Markov chains. *Annals of Mathematical statistics*, 37 :1559–1563, 1966.

[8] L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximisation technique occuring in the statistical estimation of probabilistic functions of Markov processes. *Annals of Mathematical statistics*, 41 :1 :164–171, 1970.

[9] L.E. Baum and G.R. Sell. Growth functions for transformation on manifolds. *Pac. J. Math.*, 27 :2 :211–227, 1968.

[10] P. J. Bickel, Y. Ritov, and T. Rydén. Asymptotic normality of the maximum-likelihood estimator for general hidden Markov models. *The Annals of Statistics*, 26 :4 :1614–1635, 1998.

[11] P.J. Bickel and Y. Ritov. Inference in hidden Markov models I : local asymptotic normality in the stationary case. *Bernouilli*, 2 :199–228, 1996.

[12] Hervé A. Bourlard and Nelson Morgan. *Connectionist speech recognition : a hybrid approach.* Kluwer academic publ., 1994.

[13] M. Boznar, M. Lesjak, and P. Mlakar. A neural network based method for short-term predictions of ambient SO_2 concentrations in highly polluted industrial areas of complex terrain. *Atm. Env.*, 27B :2 :225–230, 1993.

[14] H. Cartan. *Calcul différentiel.* Herman, 1970.

[15] H. F. Chen, L. Guo, and A.-J. Gao. Convergence and robustness of the Robbins-Monro algorithm truncated at randomly varying bounds. *Stoch. Proc. Appl.*, 27 :2 :217–231, 1988.

[16] M. Cottrell, et al. Neural modeling for time series : a statistical stepwise method for weight elimination. *IEEE Transaction on Neural Networks*, 6 :1355–1364, 1995.

[17] W.M. Cullogh and W. Pitts. A logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 1943.

[18] G. Cybenko. Approximation by superposition of sigmoidal functions. *Mathematics of Control, Signals and Systems*, 2 :303–314, 1989.

[19] D. Dacunha-Castelle and M. Duflo. *Probabilités et statistiques 2. Problèmes à temps mobile.* Masson, 1993.

[20] D. Dacunha-Castelle and E. Gassiat. Testing in locally conic models, and application to mixture models. *ESAIM : Probability and statistics*, 1 :285–317, 1997.

[21] A.P. Demster, N.M. Lair, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal statistical society of London*, Series B :39 :1–38, 1977.

[22] S. R. Dorling and M. W. Gardner. Neural network based air quality modelling in London UK : What are the requirement for accurate simulations. In *The second Symposium on Urban Environment. The AMS Meeting.*, 1998.

[23] N. Draper and H. Smith. *Applied Regression analysis.* J. Wiley and Sons, 1981.

[24] M. Duflo. *Algorithmes stochastiques.* Springer-Verlag, 1996.

[25] M. Duflo. *Random iterative models.* Springer-Verlag, 1997.

[26] M. Duflo, R. Senoussi, and A. Touati. Sur la loi des grands nombres pour les martingales vectorielles et l'estimateur des moindres carrés d'un modèle de regression. *Annales de L'I.H.P.*, 26 :549–566, 1990.

[27] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov models : estimation and control.* Springer, 1997.

[28] C. Francq and M. Roussignol. Consistency of MLE for AR models with Markov switching. Prépublication 10, Université de Marne-La-Vallée, 1996. à paraître dans Statistics.

[29] C. Francq, M. Roussignol, and Zakoian J.M. Conditional heteroskedasticity driven by hidden Markov chains. Prépublication, Université de Marne-La-Vallée, 1999.

[30] R. A. Gallant. *Non linear statistical models.* J. Wiley and Sons, 1987.

[31] X. Guyon. *Random fields on a network-modeling : Modelling, statistics, and applications.* Probability and its applications. Springer-Verlag, 1995.

[32] B. Hajek. Cooling schedules for optimal annealing in general state space. *Math. Op. Research*, 13 :2 :311–329, 1988.

[33] J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57 :357–384, 1989.

[34] X. Haykin. *Neural networks.* Addison-Wesley, 1991.

[35] U. Holst, G. Lindgren, J. Holst, and M. Thuvesholmen. Recursive estimation in Switching autoregressions with a Markov regime . *Journal of time series analysis*, 77 :257–287, 1994.

[36] Roger A. Horn and Charles R. Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[37] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 4 :359–366, 1989.

[38] F. Jelinek. Continuous speech word recognition by statistical methods. *Proc. IEEE*, 64 :532–536, 1976.

[39] J. F. C. Kingman. Subadditive ergodic theory. *The annals of probability*, 6 :883–909, 1973.

[40] V. Krishnamurthy and T. Rydén. Consistent estimation of linear and non-linear autoregressive models with Markov regime. *Journal of time series analysis*, 19 :3 :291–307, 1998.

[41] V Krishnamurty and John B. Moore. On-line estimation of hidden Markov model parameters based on the Kullback-Leibler information mesure. *IEEE transaction on signal processing*, 41 :8 :2557–2573, 1993.

[42] A. Krogh et al. Hidden Markov models in computational biology : Applications to protein modeling. *J. Mol. Biol.*, 235 :1501–1531, 1994.

[43] Y. LeCun. Une procédure d'apprentissage pour réseau à seuil assymétrique. *Cognitiva*, 85 :599–604, 1985.

[44] F. LeGLand and L. Mevel. Basic properties of the projective product, with application to products of column-allowable nonnegative matrix. *Mathematics of control, Signal, and Systems, à paraître*, 1999.

[45] F. LeGLand and L. Mevel. Exponential forgetting and geometric ergodicity in Hidden Markov Models. *Mathematics of control, Signal, and Systems, à paraître*, 1999.

[46] B. G. Leroux. Maximum-likelihood estimation for hidden Markov models. *Stochastic Process. Appl.*, 40 :127–143, 1992.

[47] B. G. Leroux and M.L. Puterman. Maximum-penalized-likelihood estimation for independent and Markov-dependent mixture models. *Biometrics*, 48 :545–558, 1992.

[48] T.A. Louis. Finding the observed information matrix when using the EM algorithm. *J. Royal Statist. Soc. Ser.*, B :44 :226–233, 1982.

[49] M. Mangeas. Propriétés statistiques des modèles paramétriques non-linéaires de prévision de série temporelles : Etude des réseaux de neurones à propagation directe. Thèse, Université de Paris 1, 1997.

[50] L. Mevel. Statistique asymptotique pour les modèles de Markov cachées. Thèse, Université de Rennes 1, 1997.

[51] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability.* Springer-Verlag, 1994.

[52] A.B. Poritz. Linear predictive hidden Markov models and the speech signal. In *IEEE Proc. intl. conf. on acoustic, speech, ans signal processing*, pages 1291–1294. Pergamon, 1982.

[53] William H. Press, et al. *Numerical recipes in C : The art of scientific computing.* Cambridge University Press, 1992.

[54] L.R. Rabiner. A tutorial on hidden Markov models and selected application in speech application. *Proceedings of the IEEE*, 77 :257–287, 1993.

[55] F. Rosenblatt. *Principles of neurodynamics.* Spartan, New York, 1962.

[56] D.E. Rumelhart and J.L. McClelland, editors. *Parallel distributed processing : exploration in the microstructure of cognition*, volume 1, pages 318–362. MIT Press/Bradford Books, 1986.

[57] R. Senoussi. Statistique asymptotique presque sûre de modèle convexe. *Ann. IHP. (Probabilités et Statistiques)*, 26 :19–44, 1990.

[58] H.J. Sussmann. Uniqueness of the weights for minimal feedforward nets with a given input-output Map. *Neural Networks*, 5 :589–593, 1992.

[59] H. Teicher. Identifiability of finite mixtures. *Annals of Mathematical statistics*, 34 :1265–1269, 1963.

[60] Vladimir Vapnik. *The nature of statistical learning theory.* Springer, 1995.

[61] A. Weigend, M. Mangeas, and A. Srivasta. Nonlinear gated expert for time series : Discovering regimes and avoiding overfitting. *International journal of Neural systems*, 6 :4 :373–399, 1995.

[62] A.S. Weigend and N.A. Gershenfeld. *Time series prediction.* Addison-Wesley, 1993.

[63] C. Wu. On the convergence property of the EM algorithm. *The Annals of Statistic*, 11 :95–103, 1983.

[64] J. Yao. On least square estimation for stable nonlinear AR processes. *The Annals of Institut of Mathematical Statistics*, 52 :316–331, 2000.

[65] J. Yao and J.G. Attali. On stability of nonlinear AR processes with Markov switching. Prépublication du SAMOS 101, Université Paris 1, Avril 1999. à paraître dans Adv. Applied Probab., 32(2).

Annexe A

Manuel de Regress et quelques autres programmes

A.1 Introduction

Ce programme utilise les réseaux de neurones pour faire de la régression non-linéaire (ou linéaire si le MLP n'a pas de couche cachée). Il traite aussi bien des observations indépendantes, que des séries temporelles, scalaires ou vectorielles. L'objectif est de fournir un outil simple pour ajuster un modèle. La principale originalité de ce logiciel est l'utilisation de méthodes statistiques pour identifier le modèle. Ainsi on n'utilise pas la procédure du "Early Stopping" qui consiste à découper une base d'apprentissage en 2 (un ensemble pour l'apprentissage et un ensemble pour tester la généralisation du MLP au cours de l'apprentissage). La méthode utilisée ici est l'algorithme SSM (Statistical Stepwise, voir 2.2.3) qui est un raffinement de "l'Optimal Brain Dommage", associé à un critère d'information. Ce logiciel, écrit principalement en C++, est prévu pour fonctionner sur de petits ordinateurs (PC ou tout autre ordinateur supportant Linux ou autre Unix-Like). Les temps de calcul restent donc raisonnables pour des problèmes statistiques de taille réelle (généralement entre quelques centaines et quelques milliers de données), sachant qu'un temps de calcul raisonnable signifie ici de quelques minutes à quelques heures. On pourra trouver ce programme (binaire Linux et sources) à l'adresse : ftp ://ftp.univ-paris1.fr/pub/SAMOS/joseph.

A.1.1 Problèmes pour l'ajustement d'un modèle

Lors de l'ajustement d'un modèle, on est confronté à deux problèmes principaux :

1. Les minima locaux pour la minimisation de la fonction de coût.
2. Le surapprentissage.

Ce programme utilise les stratégies suivantes pour contourner ces problèmes.

A.1.1.1 Eviter les minima locaux

La stratégie employée ici est simple. D'une part, les principaux algorithmes d'apprentissage (le gradient conjugué et le BFGS) utilisent volontairement des recherches unidimensionnelles. Cela n'est pas très efficace en temps de calcul, mais l'expérience prouve que cette méthode est extrêmement robuste vis à vis des minima locaux. En outre le programme est prévu pour faire automatiquement plusieurs apprentissages partant d'initialisations différentes (une dizaine suffira pour la plupart des cas). Les deux méthodes combinées donnent des résultats très satisfaisants pour un temps de calcul raisonnable.

A.1.1.2 Eviter le surapprentissage

Le principe de parcimonie permet d'éviter le surapprentissage. Ainsi, si on n'utilise que le nombre de paramètres nécessaires, il sera réduit à quasiment zéro. Pour enlever des paramètres on détermine d'abord les paramètres les moins significatifs, on élague le MLP et on utilise un critère d'information (BIC ou BIC* [1]) pour stopper l'élagage. L'algorithme SSM fonctionne bien si le modèle de départ n'est pas trop sur-paramétrisé. Le logiciel utilise une stratégie simple pour déterminer à partir de quels modèles (MLP dominants) il faut commencer l'élagage.

A.1.2 La stratégie d'identification

La recherche automatique ne fonctionne pour l'instant qu'avec des MLP ayant une seule couche cachée. Il faut d'abord rechercher un MLP totalement connecté dominant qui ne soit pas trop sur-paramétrisé. On commence donc par estimer différents MLP qui ont de plus en plus d'unités sur la couche cachée. On procède de la façon suivante pour trouver de bons modèles dominants.

A.1.2.1 Recherche de MLP dominants

1. Soit k le nombre d'unités cachées du MLP, au départ on pose $k = 0$ ou un petit nombre (au choix de l'utilisateur).

[1] En notant V_T la vraisemblance gaussienne pour une série de longueur T, S_T la moyenne de la norme au carré des erreurs, σ une estimation de S_T et d le nombre de paramètres du modèle, on définit le BIC par :

$$BIC = V_T + d \times \frac{\ln(T)}{T}$$

et le BIC^* (cf Mangeas [49]) par :

$$BIC^* = S_T + d \times \sigma \frac{\ln(T)}{T}$$

2. Estimer les poids du MLP ayant $k+1$ unités cachées. Calculer le critère d'information du modèle à $k+1$ unités cachées.
3. Si le critère d'information est meilleur que celui du modèle à k unités cachées, on remplace k par $k+1$ et on recommence à l'étape (2), sinon on garde les MLP ayant k et $k+1$ unités cachées pour les élaguer par l'algorithme SSM.

FIG. A.1 – Le meilleur MLP dominant a k+1 unités cachées

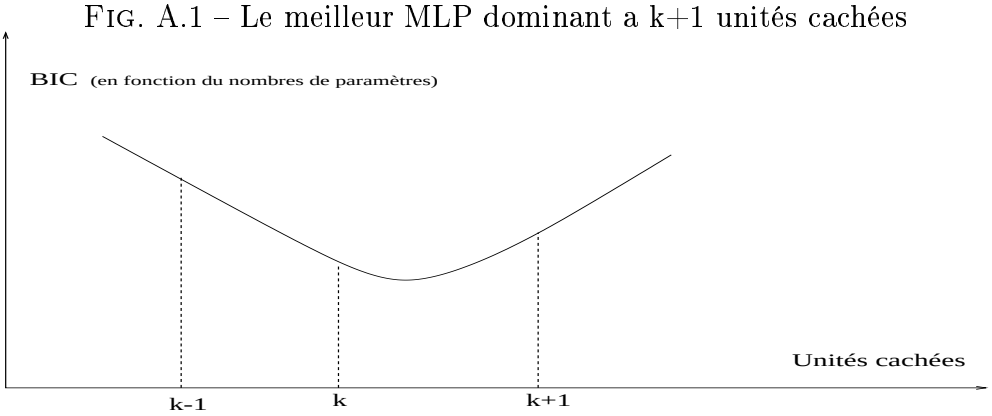
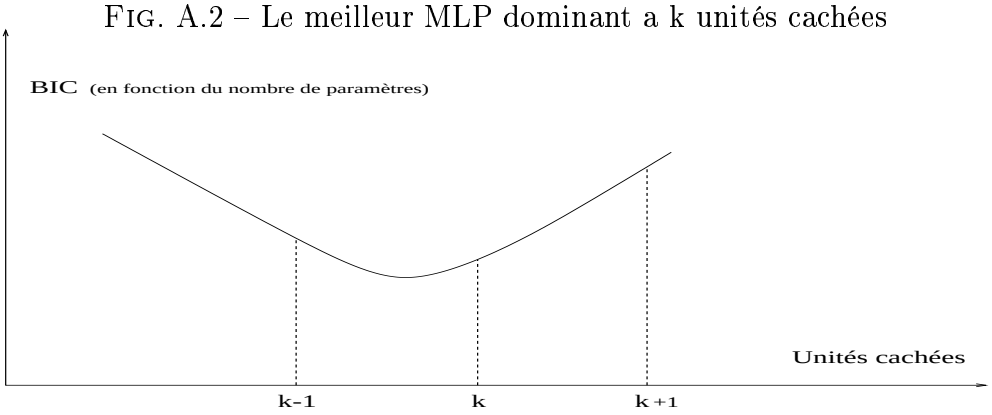


FIG. A.2 – Le meilleur MLP dominant a k unités cachées



On montre que le modèle minimisant le critère d'information (BIC (cf chapitre 4) ou BIC* (cf Mangeas [49])) convergence asymptotiquement le vrai modèle.

A.1.2.2 Identification des modèles dominants

On utilise l'algorithme SSM pour élaguer les MLP dominants, l'élagation continuera tant que le critère d'information s'améliore. On verra que les résultats sont sauvegardés dans différents fichiers, un fichier résume les résultats et donne le nom du fichier où est sauvé le meilleur MLP.

A.1.2.3 La pratique

Pour éviter les minima locaux, il faut, pour chaque architecture, faire plusieurs initialisations aléatoires. L'utilisateur peut spécifier le nombre d'estimations N_1 à faire par architecture pour la recherche de modèles dominants. Il donne aussi le nombre $N_2 \leq N_1$ de modèles dominants pour les deux architectures (avec k et $k+1$ unités cachées), qu'il faut élaguer à l'aide de l'algorithme SSM. La stratégie se résume ainsi :

1. Recherche de modèles dominants
   (a) On commence avec un petit nombre d'unités cachées ($k = 1$)
      i. On fait N_1 estimations avec cette architecture, on sauvegarde le meilleur (plus petit) BIC : BIC_k
   (b) On rajoute une unité cachée, on refait N_1 estimations on sauvegarde le meilleur BIC : BIC_{k+1}
   (c) Si $BIC_{k+1} < BIC_k$ on recommence l'étape (b). Sinon on passe à (2).
2. Identification des meilleurs modèles dominants
   (a) On identifie les N_2 meilleurs MLP avec k unités cachées
   (b) On identifie les N_2 meilleurs MLP avec $k+1$ unités cachées

Les résultats des identifications et le nom du meilleur MLP sont sauvegardés dans un fichier. Bien sûr, le logiciel permet aussi de ne faire qu'une partie de cette stratégie. Il contient aussi quelques petits utilitaires comme la visualisation des architectures des MLP. C'est le sujet du chapitre suivant.

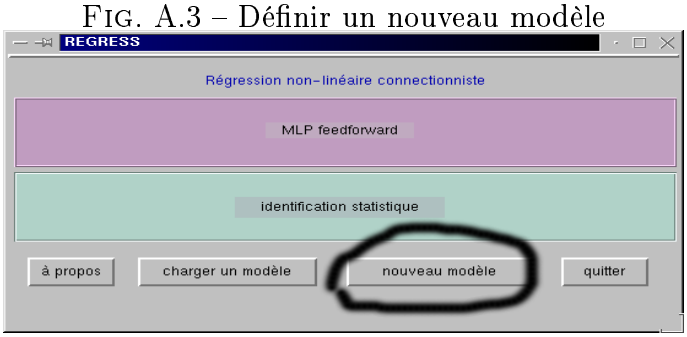
A.2 Utilisation du logiciel "Regress"

Pour lancer le logiciel, il faut l'avoir installé et taper "regress" dans une fenêtre d'émulation de terminal X (sous Linux, ce sont les programmes xterm ou rxvt). Dans toute la suite, l'utilisateur fait apparaître les fenêtres en cliquant sur le bouton cité "entre guillemets", et clique généralement sur le bouton "fait" une fois les choix effectués.

A.2.1 Formatage des données et création du perceptron initial

Lorsqu'on lance le programme, l'option "*nouveau modèle*" permet de formater les données et de créer un nouveau MLP.

FIG. A.3 – Définir un nouveau modèle

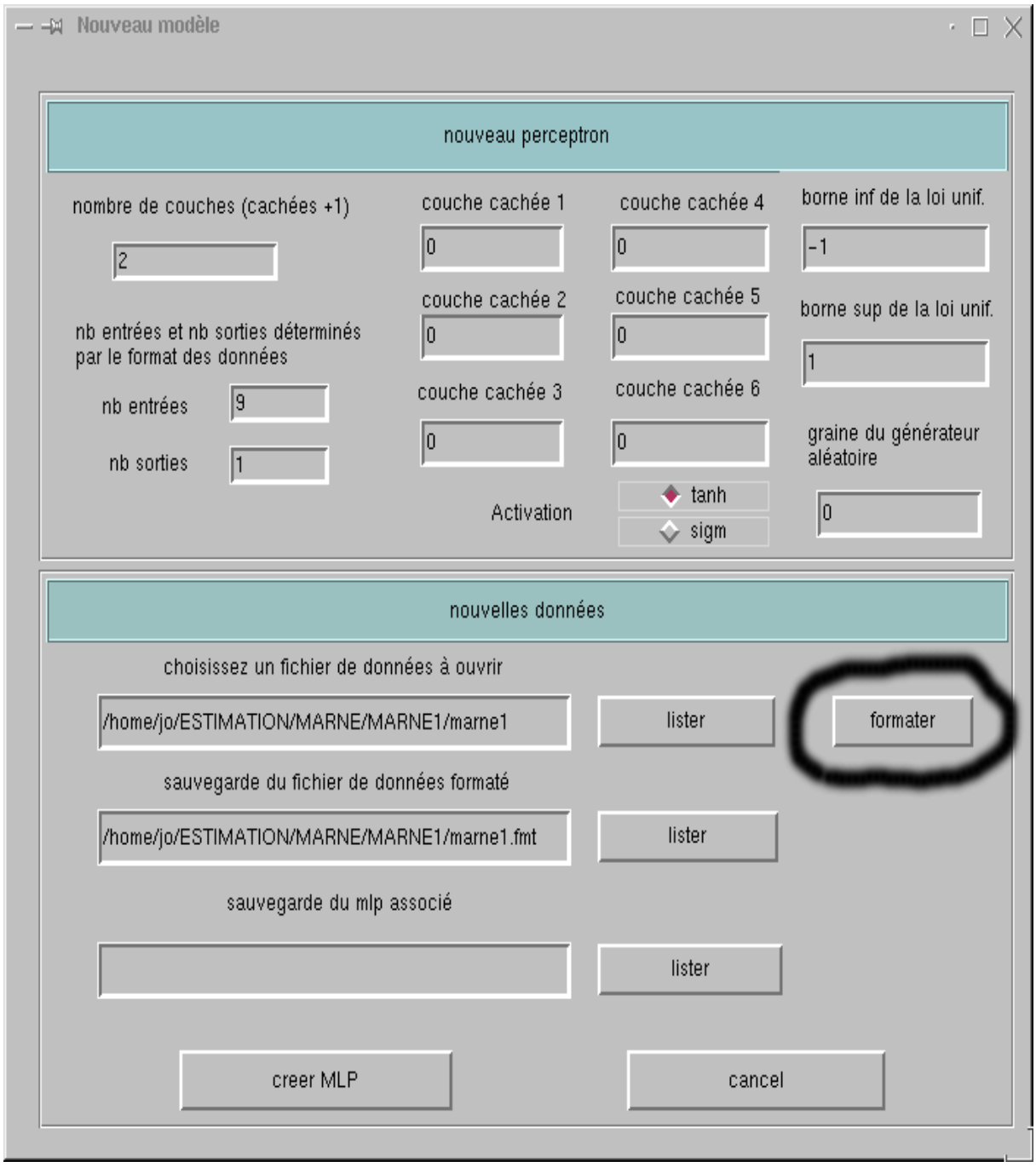


A.2.1.1 Création de la base d'apprentissage

Le MLP ne peut lire que des fichiers de données spécialement formatées, c'est-à-dire tels que les données en entrées et les sorties désirées se suivent sur une même ligne (comme dans le logiciel SNNS de l'Université de Stuttgart). La transformation d'un fichier de données en un fichier formaté est implémentée dans ce logiciel, car c'est une opération très courante.

L'utilisateur doit donc fournir un fichier dont le format est indiqué à la section suivante et demander dans une première phase de formater les données (cf figure A.4). Le fichier formaté (ici marne1.fmt) peut être réutilisé directement avec d'autres architectures initiales (pourvu que le nombre d'entrées et de sorties du perceptron soit correct). Notez que lorsqu'on formate les données, le nombre d'entrées et de sorties du nouveau MLP apparaît automatiquement (puisqu'il dépend du nombre de variables explicatives, et du nombre de variables expliquées). Néanmoins si l'on veut formater uniquement des données sans pour autant créer de nouveau modèle, il suffit de ne pas appuyer sur la touche "*créer MLP*". mais sur "fait" après avoir formaté les données.

FIG. A.4 – Formatage de Données



Le fichier de données créé par l'utilisateur On prend l'exemple d'un fichier que s'appelle marne1, mais son nom pourrait être quelconque.

Le principe est relativement simple, il faut que toutes les données soient écrites en colonnes dans un fichier "ASCII". Les différentes colonnes doivent être séparées par des espaces, l'utilisateur doit alors rajouter trois lignes au début du fichier :

- Une ligne "entrées" avec le nom des variables explicatives suivi de l'intervalle des retards éventuels.
- Une ligne "sorties" avec le nom des variables à expliquer (les retards sont implicitement 0)
- Une ligne "données" avec les noms des données dans l'ordre où elles apparaissent en colonnes.

Les séparateurs sont toujours des espaces. Par contre ces trois lignes ne doivent pas commencer par un espace.

Ici les variables explicatives sont :

- TURLD avec les retards de 1 à 3 c'est-à-dire :
  TURLD(t-1), TURLD(t-2), TURLD(t-3).
- DEBLD pour les retard 0 à 5 c'est-à-dire :
  DEBLD(t), DEBLD(t-1), DEBLD(t-2), DEBLD(t-3), DEBLD(t-4),
  DEBLD(t-5).

Les données à expliquer sont TURLD, implicitement au temps t, c'est-à-dire avec un retard 0. Le retard des données à expliquer est toujours 0. Si on veut faire par exemple des prévisions à plus long terme, il faudra décaler les entrées (variables explicatives) dans le temps.

On donne toujours un intervalle pour les retards. Si on voulait uniquement les retards 1, 4, 8 (des retards non consécutifs), il faudrait alors construire, dans le fichier à formater, une colonne de données pour chaque retard. On remarquera aussi que le fait de spécifier que la première colonne correspond à l'identifieur (cf figure [A.5], il s'agit du champ "RANG"), permet au programme d'ignorer la première colonne, puisque cette catégorie n'apparaît ni dans les entrées ni dans les sorties.

Après avoir spécifié le nom des fichiers à lire et à sauver (ici on lit "marne1" et on sauve dans "marne1.fmt", cf figure A.4) on clique sur la touche "*formater*". Le fichier au format lisible par le MLP est créé et on verra alors apparaître le nombre d'entrées et de sorties du MLP

Remarque 23 *On peut aussi saisir les dimensions du MLP directement à la main si on ne formate pas de données. Le logiciel se contente de formater des données supposées déjà pré-traitées (stationnaires, centrées, normalisées...). Le pré-traitement doit être effectué par l'utilisateur avant d'utiliser le logiciel.*

FIG. A.5 – Données de l'utilisateur, dans le fichier marne1

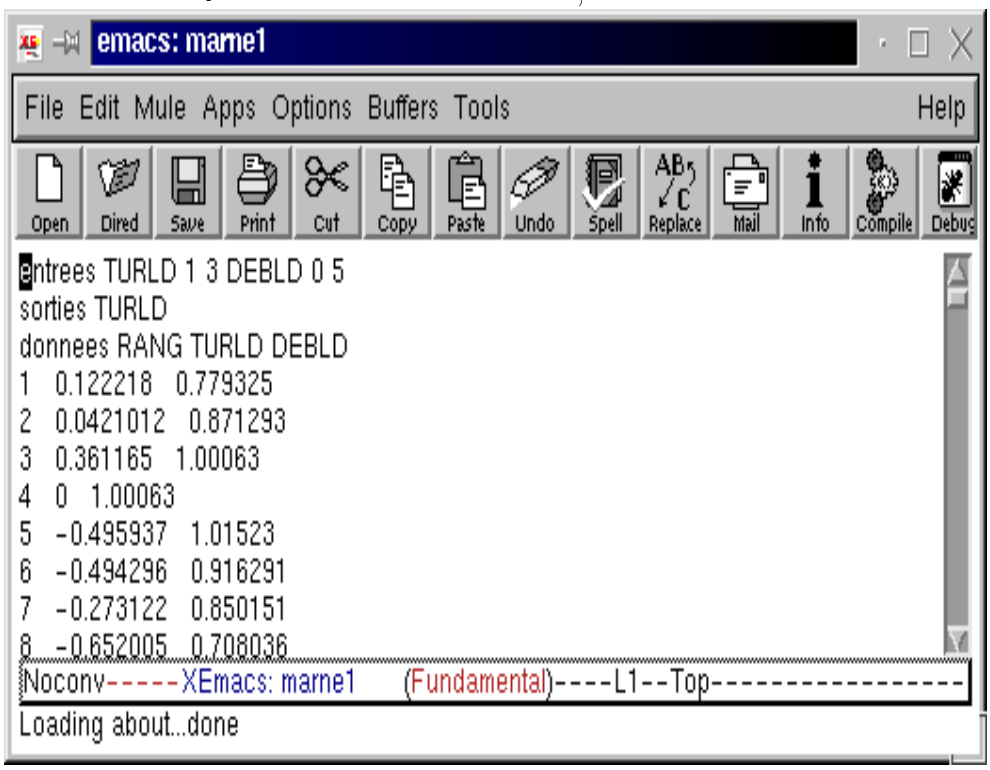
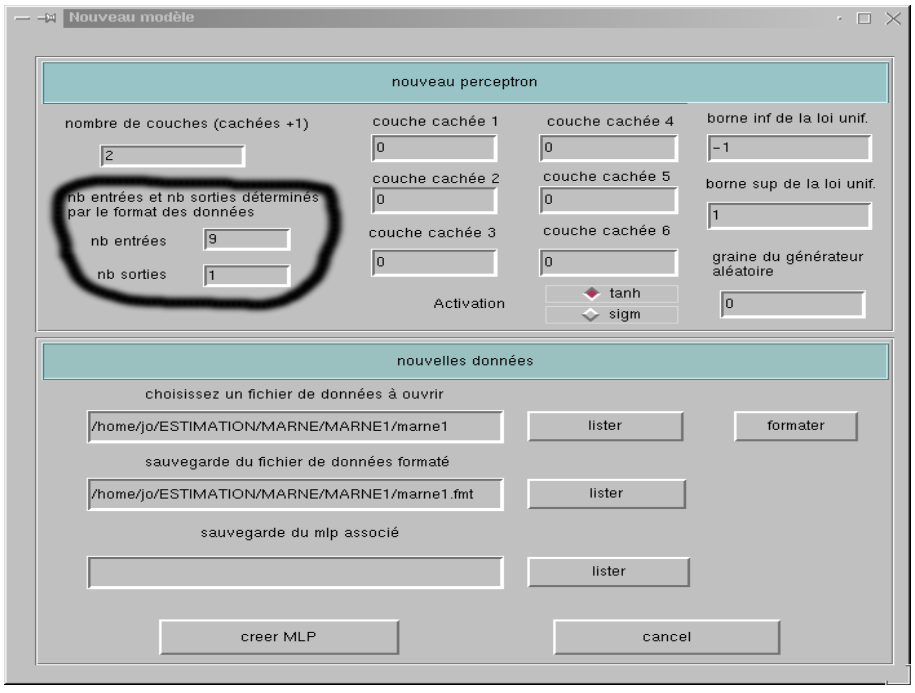


FIG. A.6 – Le nombre d'entrées et de sorties apparaît automatiquement

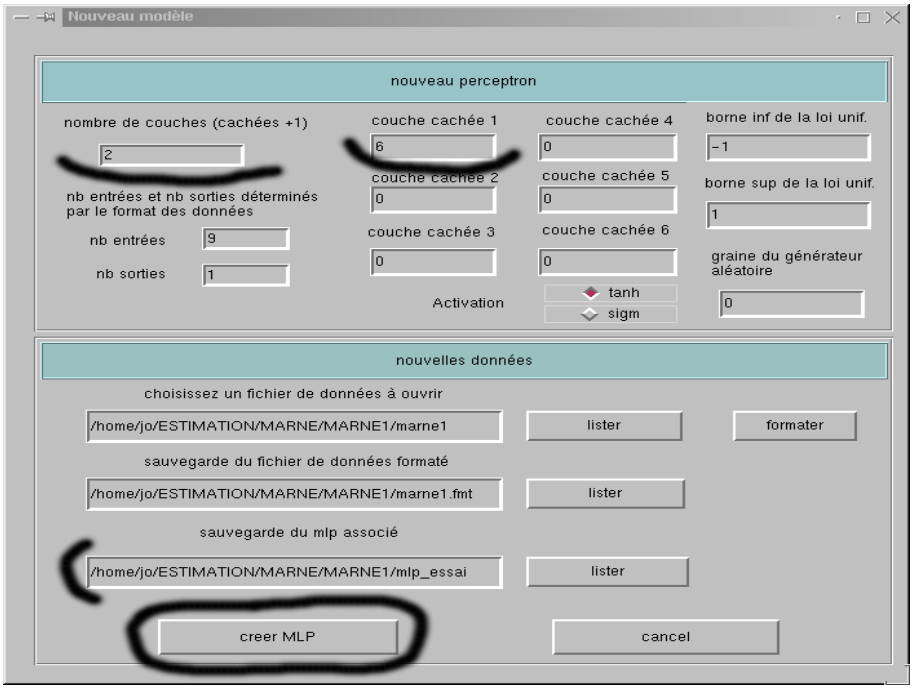


A.2.1.2 Création du MLP initial

A ce stade, il suffit d'indiquer le nombre de couches cachées, et le nombre d'unités par couches. Ensuite on peut spécifier la graine du générateur aléatoire, ainsi que l'amplitude de la loi uniforme pour le tirage des poids, ce qui permet d'initialiser plusieurs MLP de même architecture avec des poids initiaux différents. Le MLP est créé dés que l'on clique sur la touche "*créer MLP*". Le MLP est sauvé dans le fichier spécifié dans le champ ("sauvegarde du mlp associé").

Il est indispensable de créer un MLP pour tout apprentissage. Même si on veut faire une recherche automatisée d'architecture optimale, un MLP initial (avec le bon nombre d'entrées et de sorties) servira de "patron" au logiciel. Naturellement dans ce cas, le nombre d'unités cachées pour le perceptron initial est indifférent puisque le logiciel reconstruira un grand nombre de MLP suivant la stratégie d'identification expliquée dans l'introduction. Après avoir formaté la base et créé un MLP initial, il suffit de cliquer sur la touche "fait" pour retrouver la fenêtre obtenue au lancement du programme.

FIG. A.7 – Spécification des paramètres initiaux du MLP



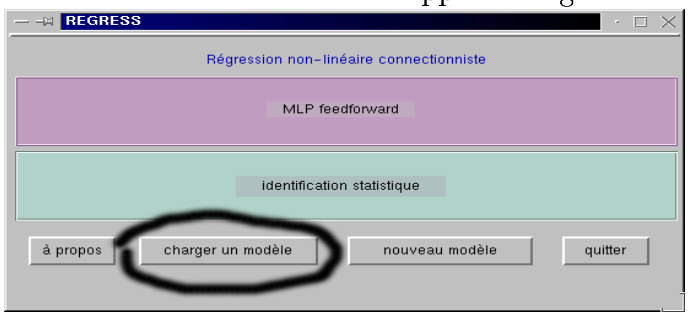
A.2.2 Paramètres de l'estimation

Pour estimer un modèle à l'aide du logiciel, il va falloir charger un modèle, configurer le type d'apprentissage souhaité et le lancer. C'est le sujet des sections suivantes.

A.2.2.1 Choisir le modèle

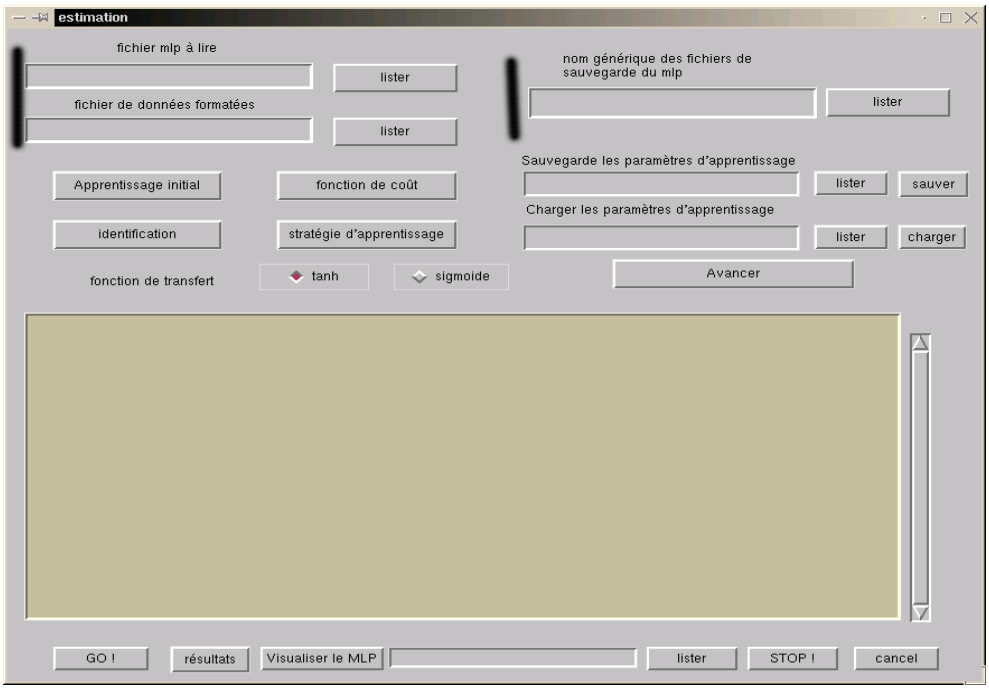
Pour ajuster le modèle il faut cliquer sur la touche "*charger un modèle* " :

FIG. A.8 – Lancer un apprentissage



On voit apparaître une nouvelle fenêtre, il faut d'abord dire quel fichier de données formatées il faut lire, quel MLP initial il faut charger, ainsi que le nom générique des fichiers destinés à sauvegarder les MLP estimés et les résultats.

FIG. A.9 – Noms des fichiers à lire et à sauvegarder

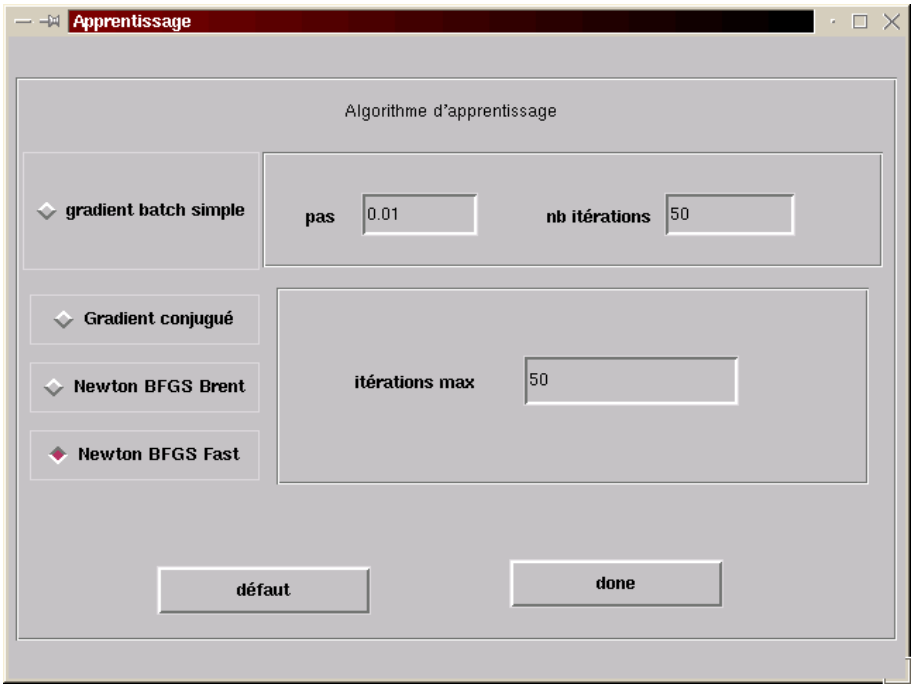


Après avoir déterminé où lire et sauver les fichiers, il faut choisir le type d'apprentissage.

A.2.2.2 Type d'optimisation pour minimiser la fonction de coût

Cliquer sur la touche "apprentissage initial" permet de faire apparaître une nouvelle fenêtre où l'on peut saisir et sélectionner les différents paramètres de l'apprentissage pour le MLP, avant la procédure d'élagage.

FIG. A.10 – Paramétrage de l'apprentissage



On a le choix entre quatre types d'apprentissage :

- Le gradient simple
- Le gradient conjugué
- Le BFGS "Brent"
- Le BFGS "Fast"

Tous ces algorithmes sont de type batch, c'est-à-dire qu'ils minimisent la fonction de coût sur tous les exemples de la base à chaque itération. Ces apprentissages sont détaillés dans le livre de Press [53].

Le gradient simple On peut donner le pas de descente le long de la direction du gradient ainsi que le nombre d'itérations maximales à faire. Cet algorithme est généralement le moins efficace pour minimiser la fonction de coût.

Le gradient conjugué C'est un algorithme robuste, plus rapide que le gradient simple, mais moins que le BFGS. Il utilise une recherche unidimensionnelle de type "Brent" (cf Press [53]).

Le BFGS C'est l'algorithme en général le plus rapide, le "Brent" utilise une recherche unidimensionnelle de type "Brent", relativement coûteuse en temps de calcul, le "Fast" utilise une recherche unidimensionnelle plus rapide, mais peut-être moins robuste. Le BFGS "Fast" est l'apprentissage par défaut.

Nombre maximal d'itérations Le nombre maximal d'itérations est fixé par le champ "itération max". Si l'erreur ne diminue plus avant ce seuil, l'algorithme détecte automatiquement le moment où il faut s'arrêter (cf paramètres avancés A.2.3.3)

A.2.2.3 Type de fonction de coût

Cliquer sur la touche "*fonction de coût*". On a le choix entre deux fonctions de coût qui sont équivalentes dans le cas où la sortie est unidimensionnelle, mais différentes si la sortie est vectorielle.

FIG. A.11 – Choisir la fonction de coût

La moyenne des carrés des erreurs résiduelles Ce sont les moindres carrés classiques, c'est-à-dire la moyenne de la norme au carré des erreurs. Si la sortie est scalaire, il est préférable de choisir cette fonction de coût car l'algorithme ira plus vite.

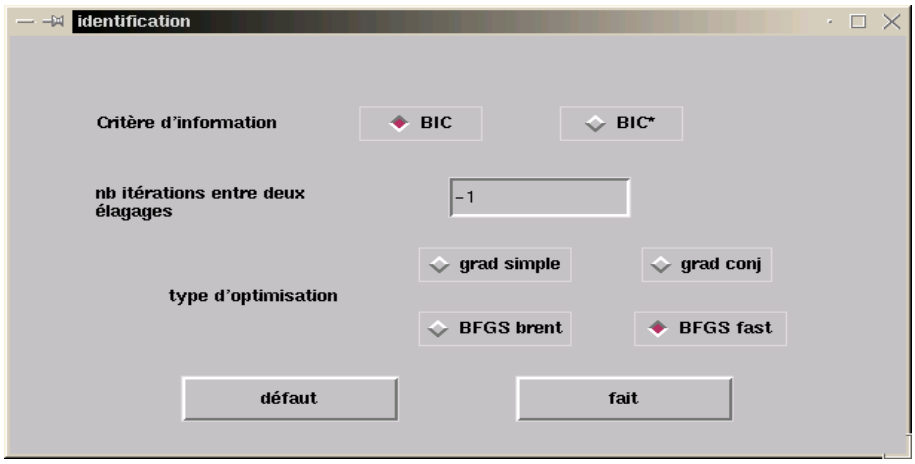
Le maximum de vraisemblance Il s'agit du maximum de vraisemblance en supposant la normalité de l'erreur résiduelle. Cette fonction de coût est équivalente aux moindres carrés dans le cas de la sortie scalaire (l'algorithme ira toutefois moins vite), mais elle est en général meilleure dans le cas vectoriel, puisqu'elle tient compte des éventuels termes diagonaux de la matrice de covariance de l'erreur (cf chapitre 4). La fonction minimisée par cet algorithme est le logarithme du déterminant de la matrice de covariance de l'erreur.

A.2.3 Paramètres de l'identification du modèle

A.2.3.1 Choix du critère d'information

Cliquer sur la touche "*identification*".

FIG. A.12 – Identification



On rappelle qu'en notant V_T la log-vraisemblance (gaussienne) pour une série de longueur T, S_T la moyenne de la norme au carré des erreurs, σ une estimation de S_T et d le nombre de paramètres du modèle, on aura le choix entre le BIC :

$$BIC = V_T + d \times \frac{ln(T)}{T}$$

et le BIC* :

$$BIC^* = S_T + d \times \sigma \frac{ln(T)}{T}$$

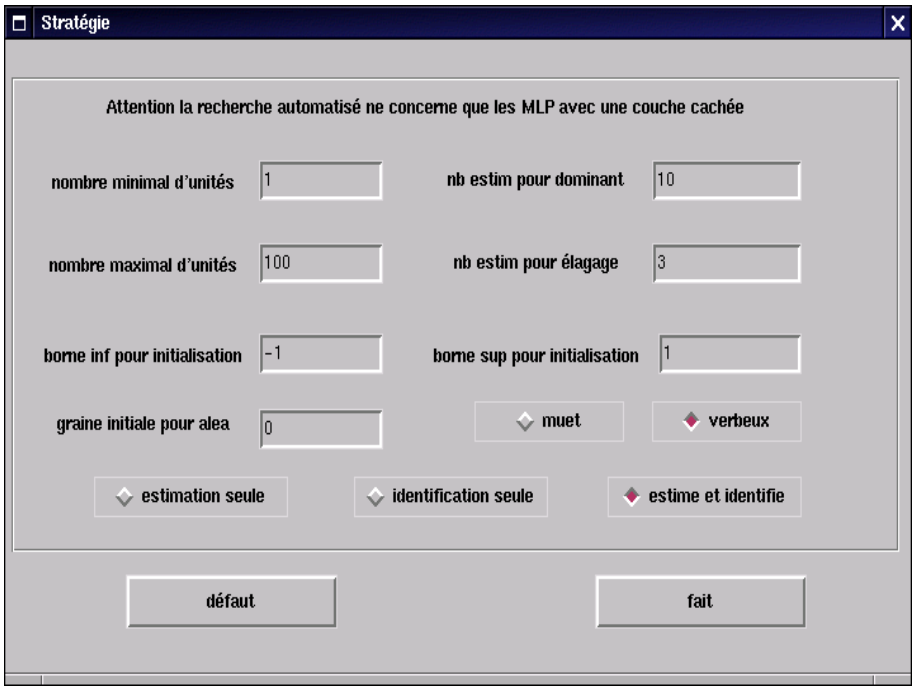
On remarque que dans le cas de MLP à sortie scalaire $V_T = \ln(S_T)$. Le BIC* s'applique aux moindres carrés, et le BIC au maximum de vraisemblance. Néanmoins, dans le cas de sortie scalaire, le BIC reste un critère valable, même si on choisit les moindres carrés comme fonction de coût. C'est le critère par défaut. On donnera aussi le nombre maximal d'itérations à effectuer pour minimiser la fonction de coût entre deux élagages (-1 correspond à un nombre d'itérations égal au nombre de paramètres, c'est le réglage par défaut). On peut aussi choisir le type d'apprentissage pour cette phase de l'apprentissage, c'est-à-dire lorsque les poids vont être élagués par la méthode SSM.

A.2.3.2 La stratégie d'identification

Il faut d'abord cliquer sur la touche "*stratégie d'apprentissage*".

On peut alors modifier les champs suivants

FIG. A.13 – Stratégie d'apprentissage



nombre minimal d'unités La recherche d'architectures dominantes commence avec des MLP ayant ce nombre d'unités cachées.

nombre maximal d'unités La recherche d'architectures dominantes ne cherchera pas de MLP ayant plus que ce nombre d'unités cachées.

nb estim pour dominant C'est le nombre d'initialisations différentes qui seront faites pour optimiser chaque architecture. Une dizaine permet d'éviter les minima locaux dans la grande majorité des cas (cela correspond au nombre N_1 dans A.1.2.3).

nb estim pour élagage C'est le nombre de MLP d'architectures dominantes qui seront élaguées pour rechercher la meilleur architecture (cela correspond aux nombre N_2 dans A.1.2.3)

borne inf et sup pour initialisation Pour chaque initialisation du MLP en début d'apprentissage les poids sont tirés au hasard suivant la loi uniforme :

$$\mathcal{U}_{[Borne\,inf,\,Borne\,sup]}$$

graine initiale pour aléa La graine initiale sera celle spécifiée. Cela permet, si on a déjà fait une étude, d'en recommencer une avec d'autres tirages aléatoires.

Choix entre muet et verbeux Le programme affichera bien plus de résultats en mode verbeux. Il est plus facile de suivre la progression de l'estimation dans ce mode, mais si l'apprentissage est très long, cela risque de consommer de la mémoire.

Choix de l'apprentissage On pourra alors choisir :

- Une estimation seule sans recherche de modèle dominant ni élagage. Les paramètres spécifiés dans "*type d'apprentissage*" seront alors utilisés, le programme effectuera le nombre d'estimations spécifiées dans "*nb estim dominant*".
- Identification seule sans recherche de modèle dominant. Le programme lira le fichier du MLP initial et effectuera une simple identification (SSM) à partir des paramètres spécifiés dans "*identification*".
- Une recherche totale du meilleur modèle ("*estime et identifie*"). Le programme recherchera les modèles dominants en estimant les poids de ces MLP suivant les paramètres spécifiés dans "*apprentissage initial*", puis ils seront élagués suivant ce qui a été spécifié dans "*identification*".

Remarque 24 *Si, on choisit "estimation seule" et que le nombre d'itérations maximal est fixé à zéro (cf A.2.2.2), le programme retourne la valeur de la fonction de coût sur la base, sans modifier le MLP.*

A.2.3.3 Paramètres avancés (il est conseillé de ne pas les modifier)

On peut fixer la valeur minimale, en deçà de laquelle les progrès sur la fonction de coût sont considérés comme nuls ("seuil redem") et le nombre de fois où les passages sous le seuil de redémarrage vont entraîner une réinitialisation de l'algorithme ("it. seuil"). Le nombre de cycles ("nb cycle") correspond au nombre maximal de redémarrages autorisés. Si ce nombre est atteint, le programme considère que la minimisation stagne et l'arrête pour le MLP en cours. Toutefois il est conseillé de garder les paramètres par défaut, à moins de savoir ce que l'on fait.

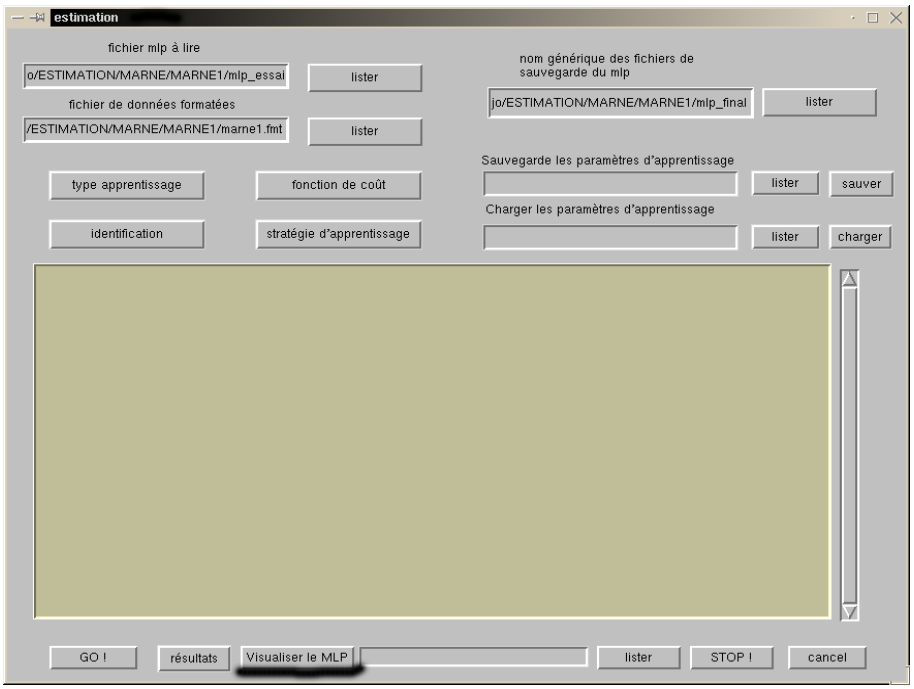
A.2.4 Déroulement de la procédure d'estimation/identification

A.2.4.1 Lancer la procédure

On pourra noter que l'on peut sauver (ou charger) tous les paramètres de l'apprentissage dans un fichier grâce aux champs "*sauvegarder les paramètres d'apprentissage*" et "*charger les paramètres d'apprentissage*". La sauvegarde et le chargement ne seront effectifs qu'après avoir cliqué sur les touches "*sauver*" ou "*charger*". Une fois toutes les

options déterminées, il suffit de cliquer sur la touche "GO"'. Si on veut arrêter l'apprentissage, il faut cliquer sur la touche "STOP". Les résultats sont sauvés au fur et à mesure, seuls ceux de l'estimation en cours seront perdus. L'évolution de l'apprentissage (erreur résiduelle, élagage) apparaît dans la fenêtre principale et permet de savoir si l'apprentissage se passe bien. En cours d'apprentissage, on peut visualiser n'importe quelle architecture en remplissant le champ après la touche "*visualiser le MLP*" et en cliquant sur cette touche.

FIG. A.14 – Fenêtre de visualisation de l'apprentissage



A.2.4.2 Les différents fichiers sauvegardés

Les noms dépendent du nom générique de sauvegarde. Pour illustrer notre propos, on supposera que celui-ci est "mlp_fin", mais il peut être quelconque.

Le MLP créé par les procédures "estimation seule" ou "identification seule" Le MLP après apprentissage sera sauvé dans le fichier portant le nom de sauvegarde générique (ici "mlp_fin").

Les MLP créés par la procédure "estime et identifie" Le programme crée des fichiers automatiquement sur la base du fichier générique. Ainsi si on donne le nom "mlp_fin", les MLP de chaque architecture dominante seront sauvés dans un fichier dont

le nom suit la règle : "mlp_fin"+"nombre d'unités cachées"+"C"+"graine du générateur aléatoire" et après élagage il aura en plus la terminaison "_ssm". Par exemple le MLP dominant avec 3 unités cachées qui a été initialisé avec la graine 2 sera sauvé après élagage sous la forme :"mlp_fin3C2_ssm".

Les résumés de résultats

estimation seule (cf A.2.3.2) L'erreur de ce mlp est conservée dans le fichier avec l'extension "_err" (ici "mlp_fin_err").

identification seule (cf A.2.3.2) L'erreur et le BIC de ce mlp sont conservés dans le fichier avec l'extension "_bic" (ici "mlp_fin_bic").

estime et identifie Le fichier "mlp_fin_best_ssm" donne un résumé des résultats avec le nom du meilleur MLP, d'après le critère d'information.

A.2.5 Tester un modèle sur une base de données

Une fois qu'un modèle a été estimé, on peut étudier ses performances sur n'importe quel jeu de données. On peut ainsi obtenir d'un modèle les erreurs sur toute la base d'apprentissage, ou sur une base de validation. Il suffit de rentrer son nom dans le champ "*fichier de MLP à lire*", le nom de la base formatée désirée dans le champ "*fichier de données formatées*" de cliquer sur la touche "*résultats*" au lieu de "GO" et le programme créera les fichiers résultats sans modifier le MLP.

Le fichier résultat (avec l'extension "_*res*" comportera sur chaque ligne le numéro de l'observation, l'observation, la prévision du MLP, l'erreur entre l'observation et la prévision. Le fichier avec l'extension "_*res_cov*" regroupe toutes les évaluations utiles (la covariance de l'erreur, son déterminant, sa trace).

FIG. A.15 – Pour obtenir les résultats d'un modèle estimé

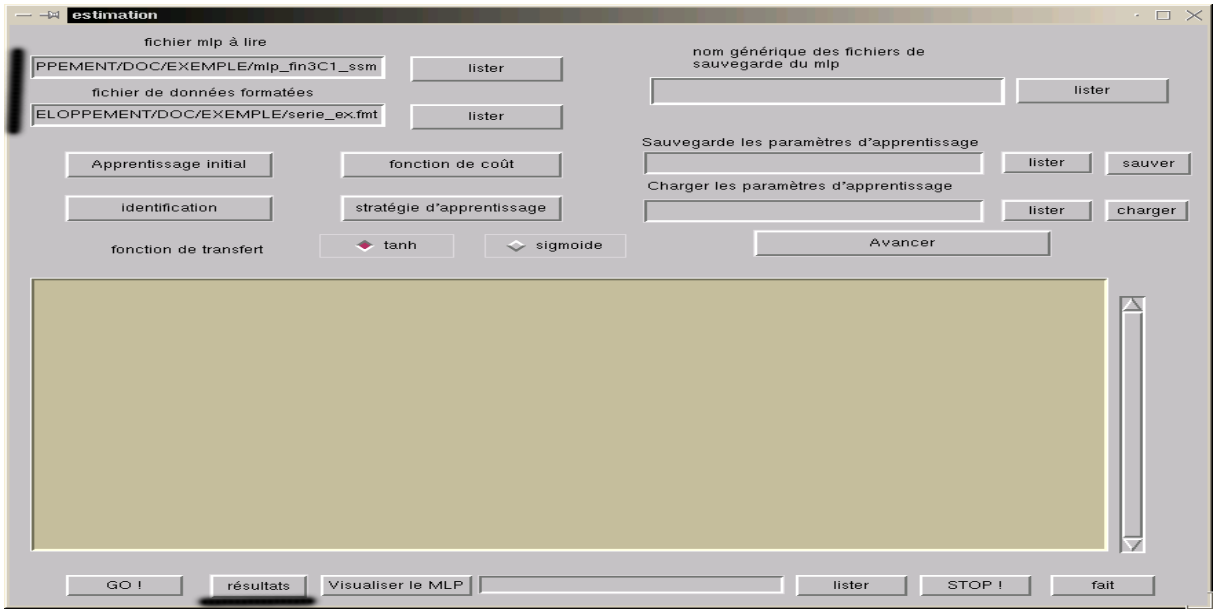
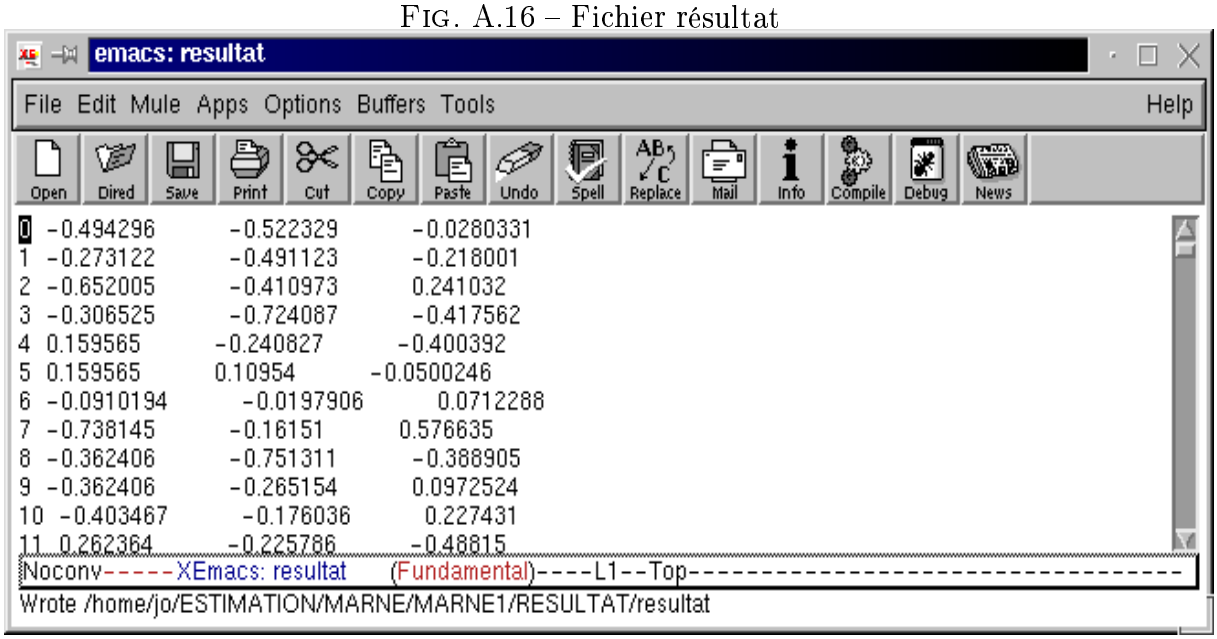


FIG. A.16 – Fichier résultat



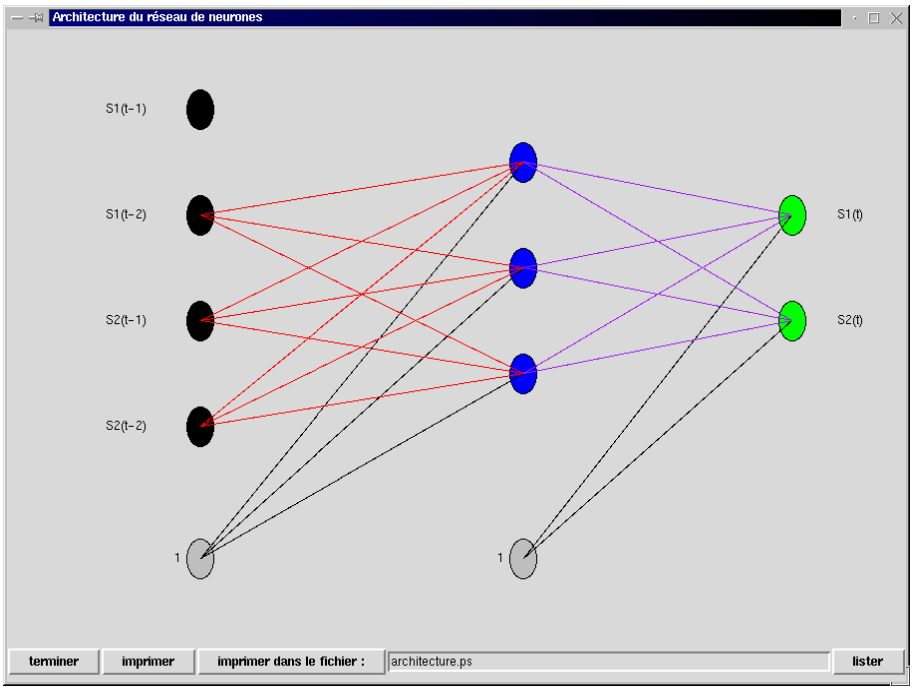
A.3 Exemple d'identification d'un modèle

Voici un exemple, pour aider l'utilisateur à prendre en main le logiciel. Les données se trouvent dans le répertoire EXEMPLE.

A.3.1 La série

On a simulé une série (y_t), $1 \leq t \leq 1000$ à l'aide d'un MLP dont le figure suivante décrit l'architecture :

FIG. A.17 – MLP pour simulation



C'est une série de $\mathbb{R}^2$, $Y_t = (y_t^1, y_t^2)^T$. on note $F_{MLP}(y_{t-2}^1, y_{t-2}^2, y_{t-1}^2)$ la fonction du MLP appliquée au retards y_{t-2}^1, y_{t-1}^2, y_{t-2}^2, qui est donc une application continue de $\mathbb{R}^3 \to \mathbb{R}^2$. On remarque que le retard y_{t-1}^1 n'intervient pas, puisque l'entrée correspondante du MLP n'est pas connectée aux unités cachées. La matrice de covariance du bruit (gaussien) est :

$$\Sigma = \begin{pmatrix} 1.04 & 0.5 \\ 0.5 & 1.09 \end{pmatrix}$$

$\Sigma = AA^T$, avec

$$A = \begin{pmatrix} 1 & 0.3 \\ 0.2 & 1 \end{pmatrix}.$$

Le modèle correspond donc à l'équation :

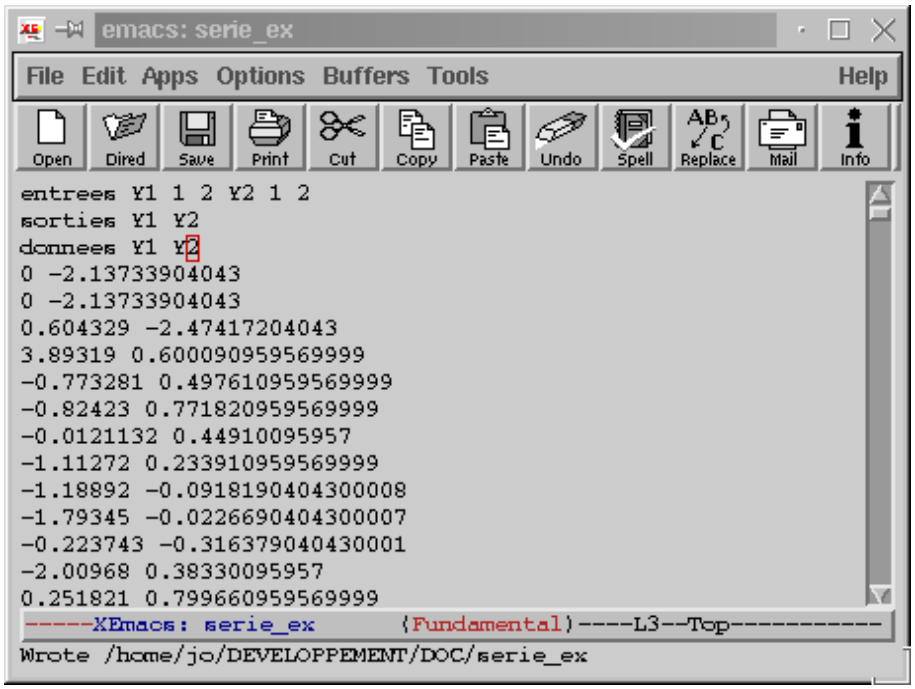
$$Y_{t+1} = \begin{pmatrix} Y^1_{t+1} \\ Y^2_{t+1} \end{pmatrix} = F_{MLP}(y^1_{t-2},\, y^2_{t-2},\, y^2_{t-1}) + A \begin{pmatrix} \varepsilon^1_{t+1} \\ \varepsilon^2_{t+1} \end{pmatrix}$$

où (ε^i_t), $i \in \{1,\, 2\}$ sont i.i.d. $\mathcal{N}(0, 1)$. On simule une série bidimensionnelle de longueur 1000. Cette série est sauvée dans le fichier "*série_ ex*". On va maintenant modéliser cette série à l'aide du logiciel.

A.3.2 Formater la série et créer un MLP initial

On va estimer la série en laissant un retard de 2 sur chaque dimension. On spécifie donc que les entrées seront les retards d'ordre deux sur les deux dimensions. La figure suivante montre les préambules adéquats.

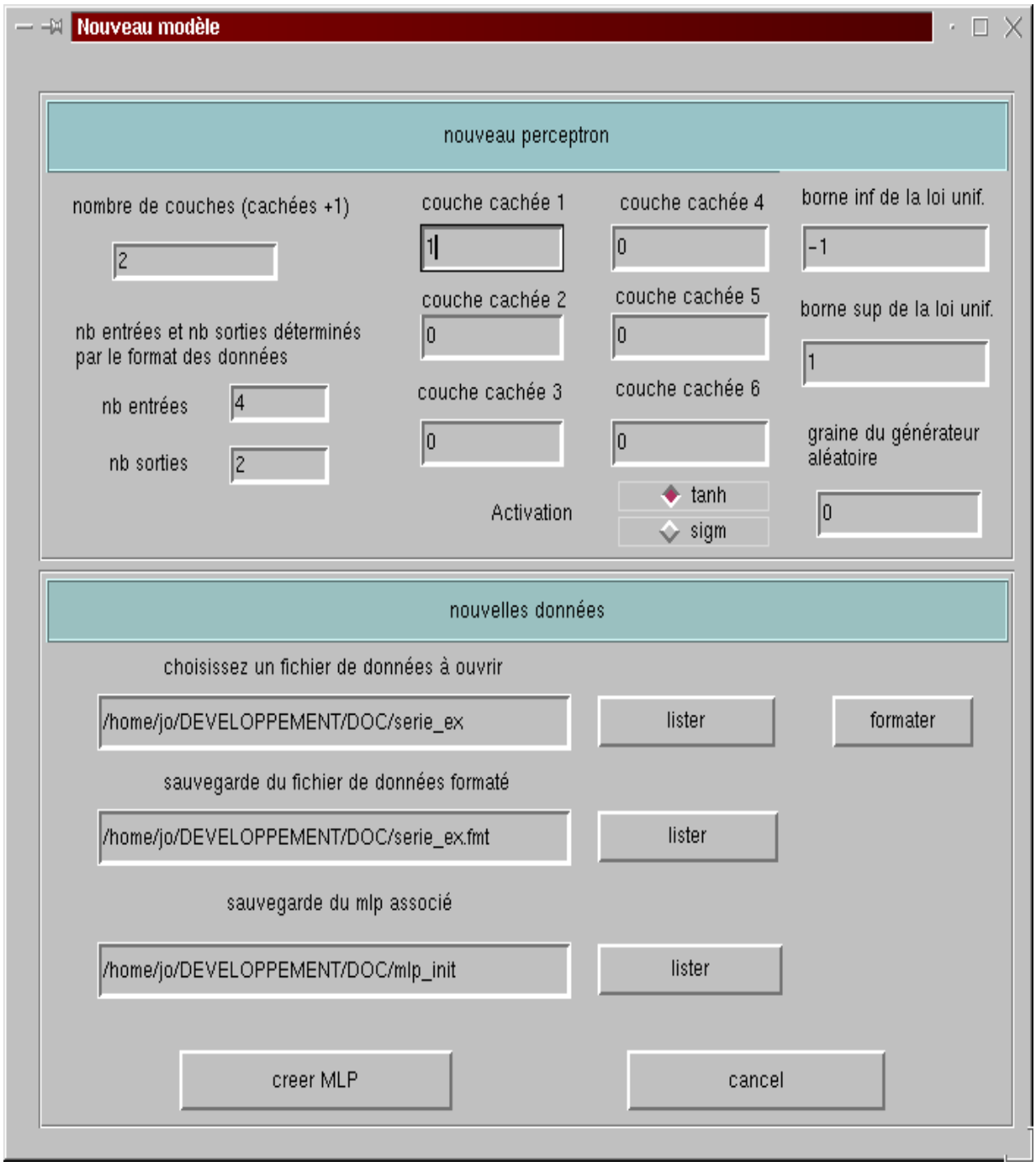
FIG. A.18 – Le fichier à formater



Admettons que l'on sauve la série formatée dans le fichier "*série_ ex.fmt*". On indique au programme le nom du fichier et après avoir formaté la série on crée un MLP initial avec 1 unités cachée (si on utilise la recherche automatique le nombre d'unités cachées du MLP initial n'importe pas). On sauvera ce MLP dans le fichier "*mlp_ init*" (par exemple). La figure A.19 montre comment on remplit les champs pour ce nouveau modèle.

Après avoir cliqué sur la touche "créer MLP", cliquez sur "cancel" puis sur "charger un modèle".

FIG. A.19 – Le nouveau modèle



A.3.3 Paramétrer l'apprentissage

Après avoir spécifié le nom générique des fichiers de sauvegarde des MLP et des fichiers résultats, on peut charger une configuration d'apprentissage toute faite en chargeant le fichier "config_app" grâce au champ "*charger les paramètres d'apprentissage*". On peut vérifier alors comment sont configurées les différentes composantes de l'apprentissage en cliquant sur les touches "type d'apprentissage", "identification", "fonction de coût" et "stratégie d'apprentissage". Ici l'estimation se fait grâce à l'algorithme BFGS avec 50 itérations au maximum, la fonction de coût est la vraisemblance, il y aura 50 itérations au maximum pour estimer le MLP entre 2 élagages, pour chaque architecture le programme fera 10 estimations avec des initialisations différentes et il élaguera les 3 meilleurs MLP des architectures dominantes. Pour commencer l'apprentissage cliquer sur la touche "GO".

A.3.4 Les fichiers de sauvegarde des résultats

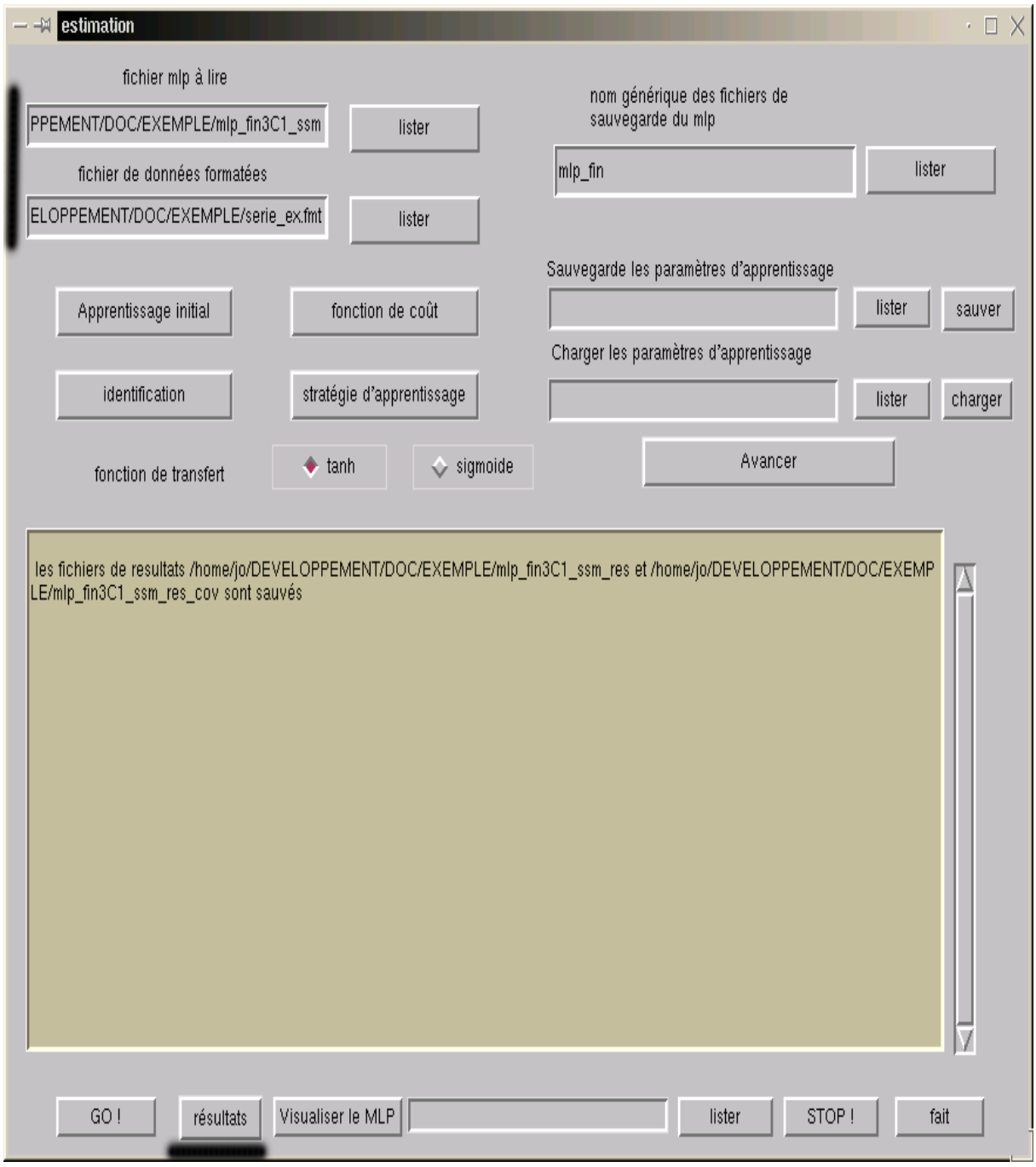
A.3.4.1 Les MLP estimés

Si on veut visualiser une architecture, par exemple celle de "mlp_fin3C1_ssm", il faut rentrer le nom "mlp_fin3C1_ssm" dans le champ après "*visualiser le mlp*" et cliquer sur cette touche.

A.3.4.2 Les performances des différents MLP

Le fichier regroupant les meilleurs MLP dominants obtenus sera sauvé sous le nom de sauvegarde général avec l'extension "_ssm". Si on veut voir les prévisions et les erreurs d'un MLP, il faudra entrer son nom dans le champ "*fichier mlp* à lire", entrer le nom de la base sur laquelle le MLP doit être évalué dans le champs "*fichier de données formatées*" (donc ne rien changer s'il s'agit de la base d'apprentissage, ou modifier le nom s'il s'agit d'une base de validation) et cliquer sur la touche "*résultats*", comme indiqué sur la figure A.20. Ici le fichier "mlp_fin3c1_ssm_res" contiendra les prévisions du mlp, les données désirées et les erreurs, tandis que "mlp_fin3c1_ssm_res_cov" contiendra la matrice de covariance de l'erreur, sa trace, son déterminant et les différents critères d'information.

FIG. A.20 – Obtenir les résultats d'un MLP



A.4 Programmes de simulations

Il est important de pouvoir simuler des séries temporelles pour tester un modèle statistique ou probabiliste. Pour cela on dispose de deux programmes :

- "simulation_mlp"
- "simulation_hyb"

le programme "simulation_mlp" permet de simuler un modèle autorégressif, dont la fonction de régression est un perceptron multicouches. Le programme "simulation_hyb" permet de simuler un modèle autorégressif à changements de régime markoviens

Ces programmes n'ont pas d'interface graphique, il faut donc utiliser une ligne de commande.

A.4.1 Simulation d'une série par perceptron multicouches

Le programme "simulation_mlp" prend en argument un nom de fichier, dans lequel figurent tous les noms de fichiers utiles à la simulation. Nous l'appellerons par exemple "config_simu_mlp", Pour lancer le programme, on tape donc dans une émulation de terminal X, "simulation_mlp config_simu_mlp".

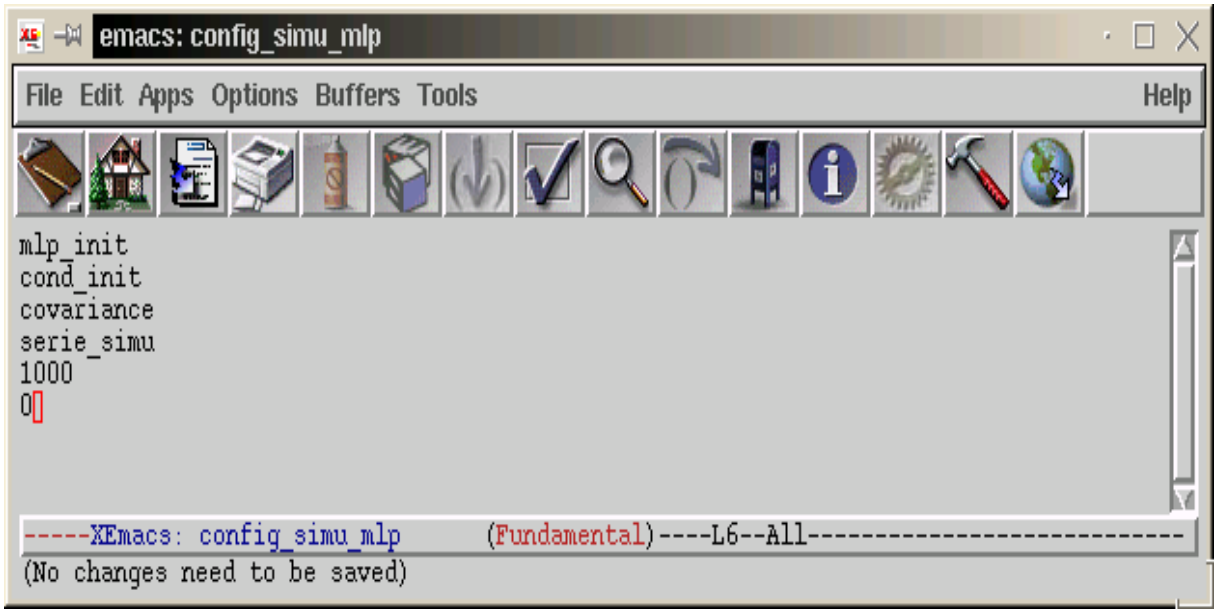
A.4.1.1 Le format du fichier en argument

Il est composé des lignes suivantes :

1. Le fichier où lire le MLP
2. Le fichier où lire les conditions initiales
3. Le fichier où lire l'écart type de l'erreur
4. Le nom du fichier où sauvegarder la série
5. La longueur de la série simulée
6. La graine du générateur aléatoire

La figure A.21 montre le fichier “config_simu_mlp”.

FIG. A.21 – Le fichier “config_simu_mlp”



A.4.1.2 Fichier où lire le MLP

La première ligne donne :

- le nombre de couches (nombre de couches cachées +1) (NC)
- La dimension de l'entrée (NE)
- Les dimensions des couches cachées. (C_i, $1 \leq i \leq NC-1$)
- La dimension de la sortie (NS)
- La colonne qui suit donne les poids du MLP

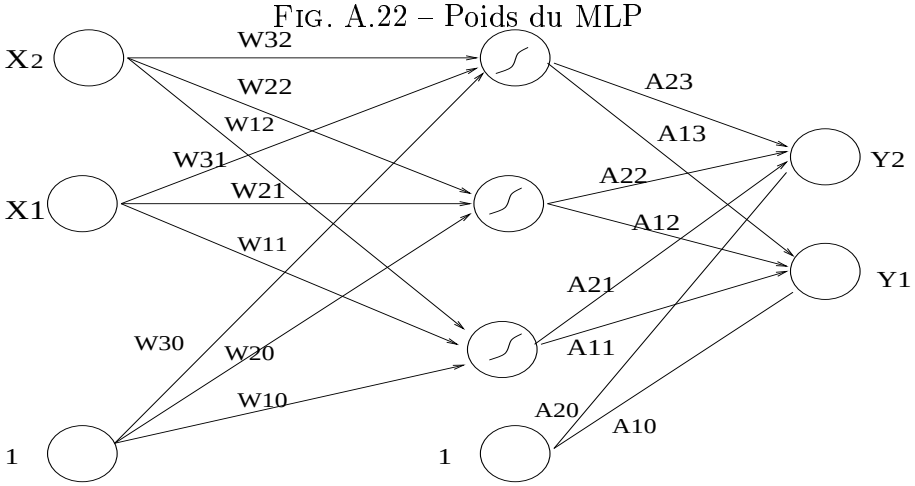
Les poids sont classés de la façon suivante :

Les poids associés au biais de chaque couche (cachée et sortie), ce sont les ($\sum_{i=1}^{NC-1} C_i + NS$) premiers poids.

Du poids 1 (connexion partant du biais sur la première couche cachée reliée à la première unité cachée) au poids $\sum_{i=1}^{NC-1} C_i + NS$ partant du biais sur la sortie reliée à la dernière unité de sortie.

Les poids sont notés couches après couches comme dans l'exemple suivant.

FIG. A.22 – Poids du MLP



TAB. A.1 – Le format du fichier correspondant au MLP

2	2	3	2
W_{10}			
W_{20}			
W_{30}			
A_{10}			
A_{20}			
W_{11}			
W_{21}			
W_{31}			
W_{12}			
W_{22}			
W_{32}			
A_{11}			
A_{21}			
A_{12}			
A_{22}			
A_{13}			
A_{23}			

A.4.1.3 Fichier conditions initiales

Comme le programme est conçu pour simuler des séries multidimensionnelles, il faut, sur la première ligne du fichier, indiquer la dimension de la série, les retards pris en compte sur chaque dimension ; puis sur la ligne suivante donner toutes les valeurs initiales. Pour simuler une série tridimensionnelle, avec un ordre de régression de 1 sur la première dimension, de 3 sur la deuxième, de 2 sur la troisième et toutes les valeurs initiales à zéros, on devra donc écrire dans le fichier des conditions initiales :

3	1	3	2		
0	0	0	0	0	0

A.4.1.4 Fichier écart-type

Il s'agit de la matrice M, tel que $M \times M^T = \Sigma$, est la covariance de l'erreur, on indiquera d'abord dans le fichier ses dimensions, puis les coordonnées ligne par ligne.

Ainsi la matrice

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

sera notée dans le fichier :

3	3	
1	2	3
4	5	6
7	8	9

A.4.2 Simulation d'une série à changements de régime markoviens (modèles hybrides)

Le programme "simulation_hyb" prend en argument un nom de fichier, dans lequel figurent tous les noms de fichiers utiles à la simulation. Nous l'appellerons par exemple "config_simu_hyb", Pour lancer le programme, on tape donc dans une fenêtre d'émulation de terminal X : "simulation_hyb config_simu_hyb".

A.4.2.1 Le format du fichier en argument

Il est composé des lignes suivantes :

1. Le fichier où lire le modèle autorégressif à changements de régime markoviens (ici "modele_hyb")

2. Le fichier où lire les conditions initiales
3. Le nom du fichier où sauvegarder la série
4. La longueur de la série simulée
5. La graine du générateur aléatoire

Le seul fichier nouveau par rapport à la section précédente est “modèle_hyb”.

FIG. A.23 – Format du fichier “config_simu_hyb”

A.4.2.2 Le fichier du modèle à changement de régime markovien

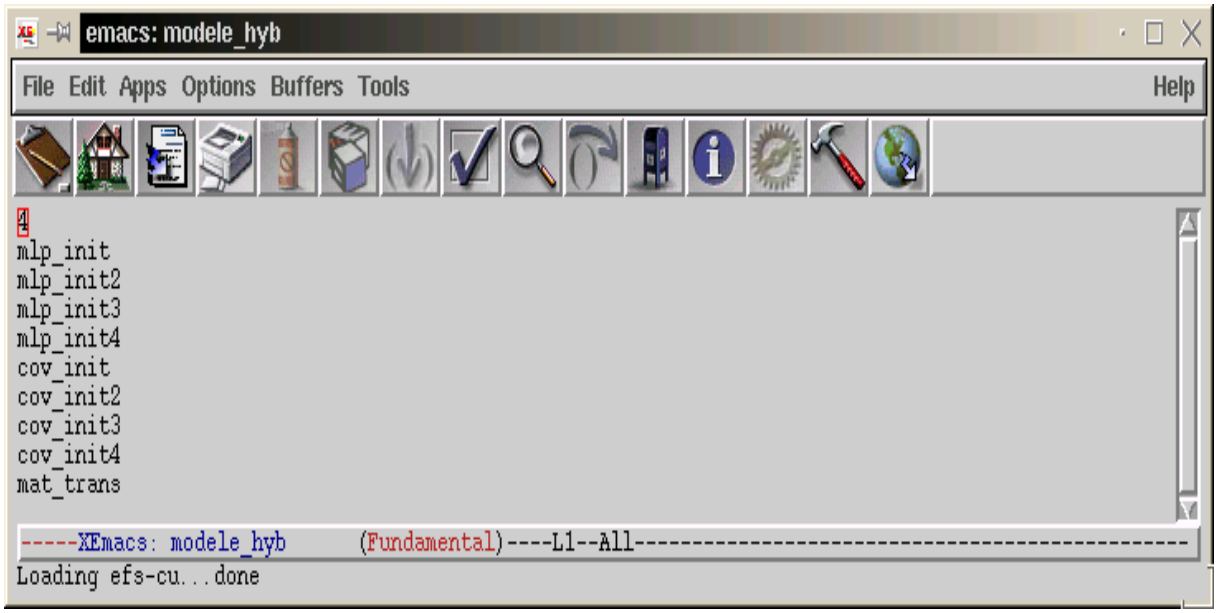
Le fichier général Ce fichier est composé de

- Le nombre de régimes
- Les différents fichiers où lire les MLP qui sont les fonctions de régression du modèle
- Les différents fichiers où lire les covariances associées à chaque modèle de régression
- La matrice de transition de la chaîne de Markov cachée.

Les fichiers MLP ont le format indiqué dans la section A.4.1.2.

Nous donnons un exemple d'un modèle avec 4 régimes, le cas général s'en déduit aisément.

FIG. A.24 – Format du fichier "model_hyb" pour 4 régimes



Les fichiers covariances Il s'agit de la matrice Σ, symétrique tel que la covariance de l'erreur pour la simulation sera $Cov_{erreur} = \Sigma \times \Sigma^T$. On indiquera d'abord dans le fichier sa dimension, puis les coordonnées au dessous de la diagonale ligne par ligne.

Ainsi la matrice

$$M = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 3 & 6 & 9 \end{pmatrix}$$

sera notée dans le fichier :

3		
1		
2	5	
3	6	9

On pourra trouver un exemple de ces fichiers dans le répertoire d'exemples.

Le fichier de la matrice de transition Il s'agit de la matrice A stochastique en colonne. On indiquera d'abord dans le fichier ses dimensions, puis les coordonnées ligne par ligne.

Ainsi la matrice

$$A = \begin{pmatrix} 0.95 & 0.05 & 0.1 & 0.01 \\ 0.01 & 0.9 & 0.02 & 0.01 \\ 0.02 & 0.02 & 0.8 & 0.02 \\ 0.02 & 0.03 & 0.08 & 0.96 \end{pmatrix}$$

sera notée dans le fichier :

3	3		
0.95	0.05	0.1	0.01
0.01	0.9	0.02	0.01
0.02	0.02	0.8	0.02
0.02	0.03	0.08	0.96

A.5 Appel des programmes par commande shell

Pour automatiser les estimations, il est pratique de pouvoir appeler les programmes grâce à une commande shell. C'est même la seule façon de pouvoir utiliser le programme d'estimation de modèle hybride.

A.5.1 "Regress", en ligne de commande.

"Regress" utilise le programme "estimation_mlp" pour faire les calculs. Le programme "estimation_mlp" prend en argument un nom de fichier, dans lequel figurent tous les noms de fichiers utiles qui sont saisis habituellement dans les champs de l'interface graphique de "regress".

A.5.1.1 Création d'un modèle

Il suffit de fournir en argument au programme "estimation_mlp" :

- Le nom du fichier où sauvegarder le MLP créé
- La borne inférieure pour l'initialisation aléatoire des poids
- La borne supérieure pour l'initialisation aléatoire des poids
- La graine du générateur aléatoire
- Le nombre de couches
- Le nombre d'unités dans les différentes couches

A.5.1.2 Identification d'un modèle

Supposons que les paramètres de l'apprentissage sont écris dans le fichier "config_app". Pour lancer le programme, on tape dans une fenêtre d'émulation de terminal X : "estimation_mlp config_app". Le ficher "config_app" est composé des champs suivants tous

séparés par une ligne de commentaire (elle doit obligatoirement exister). Les champs correspondent aux paramètres de "Regress" (cf A.2)

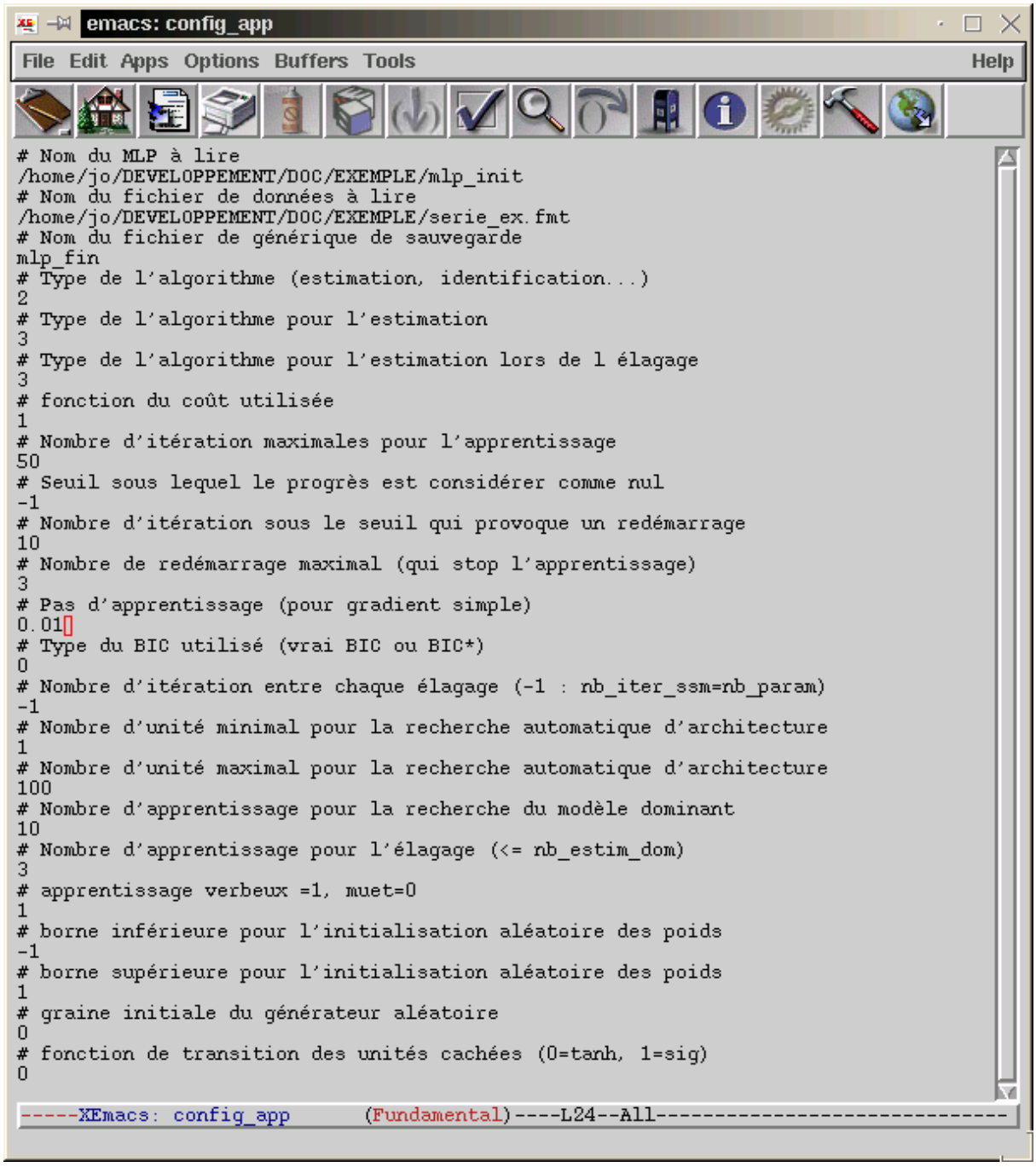
- Nom du MLP initial à lire
- Nom du fichier de données à lire
- Nom du fichier générique de sauvegarde
- Type de l'algorithme (cf A.2.3.2) :
  - 0 : estimation seule
  - 1 : identification seule
  - 2 : estimation et identification
- Type de l'algorithme pour l'estimation initiale (cf A.2.2.2)
  - 0 : gradient simple
  - 1 : gradient conjugué
  - 2 : BFGS "Brent"
  - 3 : BFGS "Fast"
- Type de l'algorithme pour l'estimation pendant l'élagage (cf A.2.3)
  - 0 : gradient simple
  - 1 : gradient conjugué
  - 2 : BFGS "brent"
  - 3 : BFGS "Fast"
- Fonction de coût (cf A.2.2.3)
  - 0 : moindres carrés
  - 1 : vraisemblance
- Nombre d'itérations maximal pour l'apprentissage (cf A.2.2.2)
- Seuil sous lequel les progrès sont considérés comme nuls (cf A.2.3.3)
- Nombre d'itérations sous le seuil qui provoque un redémarrage (cf A.2.3.3)
- Nombre de redémarrages maximal (cf A.2.3.3)
- Pas d'apprentissage
  (pour le gradient simple, cf A.2.2.2)
- Type du BIC (cf A.2.3)
  - 0 : BIC classique
  - 1 : BIC* (voir [49])
- Nombre d'itérations entre chaque élagage (cf A.2.3)
- Nombre minimal d'unités pour la recherche automatique d'architecture (cf A.2.3.2)
- Nombre maximal d'unités pour la recherche automatique d'architecture (cf A.2.3.2)
- Nombre d'apprentissages pour le recherche du modèle dominant (cf A.2.3.2)
- Nombre d'apprentissages pour l'élagage (cf A.2.3.2)
- Information lors de l'apprentissage (cf A.2.3.2)
  - 1 : verbeux
  - 0 : muet
- borne inférieure pour l'initialisation aléatoire des poids (cf A.2.3.2)
- borne supérieure pour l'initialisation aléatoire des poids (cf A.2.3.2)
- graine initiale du générateur aléatoire (cf A.2.3.2)
- fonction de transition des unités cachées

- 0 : tangente hyperbolique
- 1 : sigmoïde

Enfin, pour obtenir les résultats d'un modèle sur une série, Il suffit d'appeler le programme "estimation_mlp" avec 2 arguments :

- Le nom du fichier où lire le MLP (voir A.4.1.2)
- Le nom du fichier où lire les données formatées.

FIG. A.25 – Le fichier “config_app”



A.6 Estimation de modèles hybrides

Pour estimer ce type de modèle, il faut lancer le programme "estimation_mlp" avec 3 arguments :

- Le fichier de configuration de l'apprentissage
- Le nombre d'apprentissages, avec des initialisations aléatoires différentes.
- Le type d'estimation :
  - Algorithme E.M. : 1
  - estimation différentielle : 0

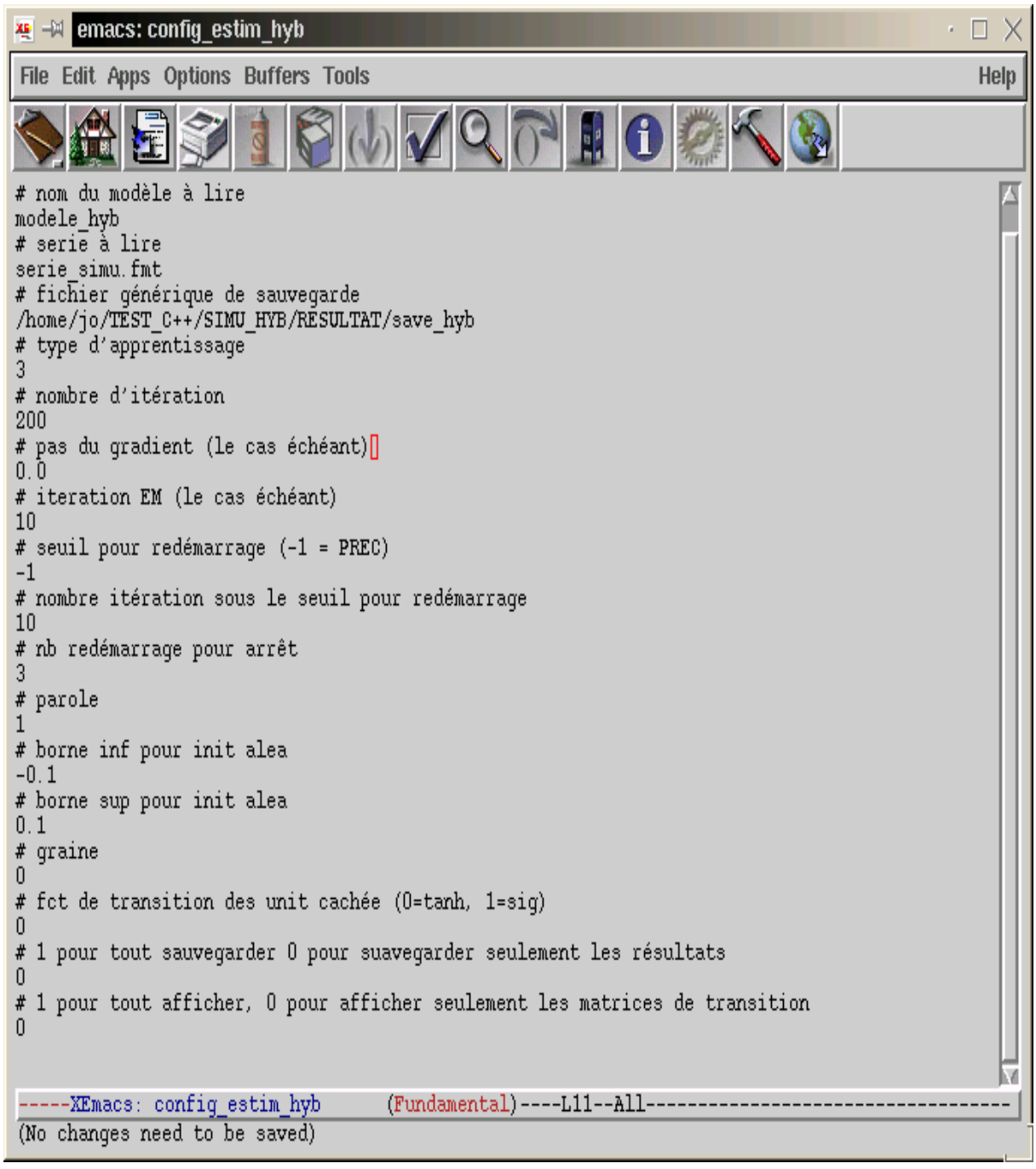
A.6.1 Le fichier de configuration de l'apprentissage

Le fichier de configuration de l'apprentissage est composé des champs suivants tous séparés par une ligne de commentaire (elle doit obligatoirement exister). Nombre de champs sont communs avec le fichier de la section A.5.1.2.

- Le nom du modèle initial à lire
- La série de données à lire
- Le fichier générique de sauvegarde
- Le type d'apprentissage
  - 0 : gradient simple
  - 1 : gradient conjugué
  - 2 : BFGS "brent"
  - 3 : BFGS "Fast"
  - 4 : gradient stochastique
- Le nombre d'itérations maximal pour l'estimation
- Le pas du gradient simple (obligatoire, mais utilisé uniquement si l'apprentissage est le gradient simple)
- Le nombre d'itérations E.M. (obligatoire, mais utilisé uniquement si l'apprentissage est l'algorithme E.M.)
- Seuil sous lequel les progrès sont considérés comme nuls
- Nombre d'itérations sous le seuil qui provoque un redémarrage
- Nombre de redémarrages maximal
- Information lors de l'apprentissage
  - 1 : verbeux
  - 0 : muet
- borne inférieure pour l'initialisation aléatoire des paramètres des MLP
- borne supérieure pour l'initialisation aléatoire des paramètres des MLP
- graine initiale du générateur aléatoire
- fonction de transition des unités cachées
  - 0 : tangente hyperbolique
  - 1 : sigmoïde

- Sauvegarde des résultats
  - 1 pour sauvegarder aussi les probabilités conditionnelles
  - 0 pour ne pas les sauvegarder
- Affichage des paramètres après chaque itération
  - 1 : tous les paramètres
  - 0 : juste la matrice de transition

FIG. A.26 – Fichier de configuration de l'apprentissage



A.6.2 Les Fichiers sauvegardés

Les noms des fichiers sauvegardés sont construits sur le nom du fichier de sauvegarde "générique". Si celui-ci est par exemple : "save_hyb", et qu'il y a 4 régimes pour le modèle, les fichiers sauvegardés seront :

- Les 4 modèles de régression
  - save_hyb_exp0
  - save_hyb_exp1
  - save_hyb_exp2
  - save_hyb_exp3
- Les 4 covariances associées au modèle
  - save_hyb_cov0
  - save_hyb_cov1
  - save_hyb_cov2
  - save_hyb_cov3
- La matrice de transition
  - save_hyb_trans
- Si la sauvegarde des probabilités conditionnelles des états a été demandée (voir section précédente) :
  - save_hyb_proba_state