

Estimation de la dimension intrinsèque d'une série temporelle et prédiction par une méthode de projection *Application au SBF 250 sur la période 1992-1997*

A. Lendasse¹, E. de Bodt², M. Verleysen¹

¹Université Catholique de Louvain, Electricity Dept., 3 pl. du Levant, 1348 Louvain-la-Neuve, Belgique.

²Université Catholique de Louvain, IAG-FIN, 1 pl. des Doyens, 1348 Louvain-la-Neuve, Belgique.

Key words: Intrinsic Dimension, Time series, Prediction, SBF 250, Dimension Reduction, CCA.

Abstract: Une étape fondamentale dans la prédiction d'une série temporelle est la détermination de son ordre auto-régressif et/ou d'une matrice auto-régressive. Nous proposons une méthode de prédiction basée sur la détermination de la dimension fractale de la série temporelle et sur une réduction de dimension par Analyse en Composantes Curvilignes (CCA). Les performances de la méthode sont illustrées par deux exemples : une série générée artificiellement et l'indice SBF 250.

1. INTRODUCTION

La prédiction d'une série temporelle est un problème très courant, que ce soit dans le domaine industriel (prédiction de la consommation électrique d'un pays, débit d'une rivière...) ou dans le domaine financier (prédiction de taux de changes, cours boursier, ...) [2,5,8,9].

Malheureusement, la prédiction doit être effectuée en l'absence d'indications sur le processus sous-jacent à ces séries temporelles. Le cas de la prédiction de la consommation électrique est rarissime : on devine bien dans ce cas que la valeur à prédire de la série dépend du jour de la semaine (composante hebdomadaire dans le vecteur auto-régressif), du mois (composante mensuelle dans le vecteur auto-régressif), ...

Souvent, les indications que l'on a sur la série sont inexistantes ou insuffisantes. Plusieurs méthodes peuvent alors être utilisées :

- on construit des modèles linéaires de prédiction (par ex. des modèles ARX) en faisant varier l'ordre auto-régressif. On choisit ensuite l'ordre optimal pour ce type de modèle linéaire ;
- on travaille par essais et erreurs avec des modèles non-linéaires (par ex. des Perceptrons Multicouches), en parcourant toute une gamme de vecteurs auto-régressifs possibles.

Ces méthodes sont souvent les seules possibles mais elles ont de gros défauts. La première surestime l'ordre auto-régressif nécessaire (parce qu'elle ne tient pas compte de dépendances non-linéaires entre les données passées) et favorise un surapprentissage, et donc une mauvaise généralisation lors de la prédiction proprement dite. La deuxième est très lourde à implémenter et souvent peu fiable ; en effet, les différents apprentissages peuvent être entachés d'erreurs qui sont provoquées par la méthode de prédiction choisie, comme par exemple la présence de minima locaux dans l'optimisation des Perceptrons Multicouches.

La méthode proposée tente de remédier à ces inconvénients. Elle sera exposée dans la deuxième partie et ensuite appliquée à un exemple artificiel simple. Dans la quatrième partie, nous tenterons de prédire les fluctuations successives de l'indice SBF 250.

2. LA MÉTHODE DE PRÉDICTION

2.1 Ordre et vecteurs auto-régressifs

L'ordre auto-régressif non-linéaire peut être défini comme le nombre optimum de valeurs passées à utiliser dans une série temporelle pour une bonne prédiction (notion à définir selon les cas). Le vecteur auto-régressif regroupe l'ensemble des valeurs à utiliser pour une bonne prédiction. L'illustration d'une méthode non-linéaire pour estimer l'ordre auto-régressif doit permettre de prendre en compte les dépendances non-linéaires entre les valeurs passées, ce que les méthodes classiques linéaires ne permettent pas.

Il y a deux façons d'interpréter ces définitions. La première, qui est la plus courante, consiste à considérer que les éléments du vecteur auto-régressif sont des éléments bruts qui proviennent directement de la série

temporelle. Il faut donc rechercher quels sont ces éléments et leur nombre optimal [5,9].

Une seconde interprétation de ces définitions consiste à supposer qu'il existe un vecteur auto-régressif idéal (c'est-à-dire le plus petit possible), mais que celui-ci est en fait composé de combinaisons (non-linéaires) de valeurs de la série brute, et non des valeurs elles-mêmes.

Dans la suite, nous utiliserons cette seconde interprétation et nous chercherons à déterminer la taille de ce vecteur de variables d'état. Nous étudierons ensuite une méthode pour le construire.

2.2 Dimension intrinsèque

Afin de déterminer l'ordre auto-régressif non-linéaire, nous allons construire un premier vecteur auto-régressif de taille n à partir de la série temporelle. Celui-ci sera constitué par les valeurs successives de la série brute et devra être suffisamment grand pour contenir toute l'information nécessaire à une bonne prédiction. Une solution sera de prendre le vecteur auto-régressif optimum pour un modèle ARX [5]; on suppose en effet que celui-ci contiendra toute l'information désirée, et même probablement trop. Des vecteurs plus grands peuvent être pris pour plus de sécurité, mais ils rendraient plus difficile la suite du travail. Les vecteurs auto-régressifs successifs ainsi construits sont disposés dans une matrice appelée matrice auto-régressive.

Comme il y a un excès d'information, nous allons rechercher la dimension fractale d de cette matrice. Cette valeur peut être interprétée comme le nombre de colonnes « non-linéairement indépendantes » de cette matrice : il existe une transformation non-linéaire qui permet de reconstruire entièrement la matrice à partir de d colonnes. Il s'agit donc bien de la dimension intrinsèque recherchée.

Pour calculer cette dimension fractale, nous utiliserons la méthode de Grassberger et Procaccia [4], mais il en existe de nombreuses autres dans la littérature [1,6,7]. Le résultat de cette méthode pouvant être une valeur non entière, nous prendrons pour la suite la valeur entière la plus proche et nous l'appellerons d (n étant la dimension d'un vecteur initial).

2.3 Vecteur auto-régressif non-linéaire

L'étape suivante consiste à construire un vecteur auto-régressif non-linéaire de taille d comme expliqué précédemment. A cette fin, nous partirons de la matrice auto-régressive que nous avons déjà utilisée.

Comme d est sa dimension intrinsèque, l'ensemble de points formé par les lignes de cette matrice forme une d -surface dans un espace de dimension n . Si nous pouvions déplier cette d -surface par une projection de l'ensemble de dimension n initial vers un ensemble de dimension d , tout en préservant la topologie de l'ensemble initial, nous aurions à notre disposition des vecteurs auto-régressifs ou plutôt des vecteurs de variables d'état de bonne qualité.

De telles méthodes de dépliage existent ; nous utiliserons celle qui a été développé sous le nom de méthode VQP (Vector Quantization Projection) et qui est connue aujourd'hui sous le nom de CCA (Curvilinear Component Analysis) [3].

Après avoir réalisé cette projection, nous obtenons le vecteur auto-régressif non-linéaire recherché. Sa dimension est optimale pour une bonne projection et il contient, grâce à sa construction, une grande partie de l'information disponible dans la série. Toutefois, comme la projection n'est pas parfaite, nous ajouterons au vecteur trouvé précédemment le dernier élément de la série. Cet artifice n'est pas du tout un inconvénient, car on a généralement $d \ll n$.

3. APPLICATION À UNE SÉRIE TEMPORELLE ARTIFICIELLE

Afin de tester la méthode générale, nous avons construit une série temporelle artificielle qui est générée par une équation non-linéaire de la forme suivante :

$$x_{t+1} = a x_t^2 + b x_{t-2} + \varepsilon \quad (1)$$

L'ordre auto-régressif non-linéaire de cette série temporelle est 2. Notons bien l'absence d'un terme x_{t-1} , ainsi que la présence d'un bruit ε (de l'ordre de 10% de la valeur maximale de la série).

Cette série est représentée à la Figure 1.

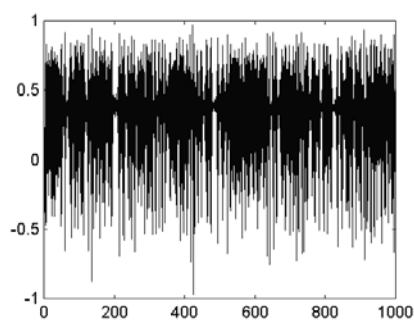


Figure 1. Série temporelle artificielle

La première étape de notre méthode est la recherche de la matrice auto-régressive optimale pour un modèle ARX.

La figure 2 représente la somme des erreurs quadratiques qui sont obtenues si on utilise des modèles ARX de taille croissante. En abscisse, nous avons l'ordre auto-régressif du modèle linéaire.

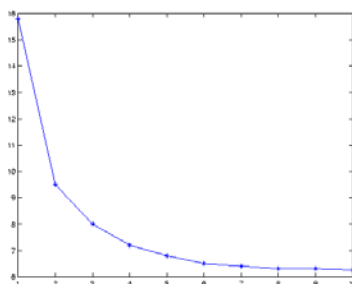


Figure 2. Somme des erreurs quadratiques pour un modèle d'ordre croissant

Pour être certain de capter toute la dynamique de la série, nous allons construire une matrice auto-régressive initiale d'ordre 6. Le calcul de la dimension fractale de cette matrice donne 2.12, ce qui est très proche de la réalité.

L'étape suivante est donc la projection de l'ensemble des points de cette matrice, de \mathbb{R}^6 vers \mathbb{R}^2 . Nous rajoutons ensuite x_t pour obtenir le vecteur auto-régressif final (de dimension 3).

La méthode a été testée en construisant un modèle non-linéaire à partir de ce vecteur auto-régressif. Le modèle utilisé est un Perceptron Multicouches à

une couche cachée. La somme des erreurs quadratiques entre les valeurs réelles et les valeurs prédites est égale à 5.

Des modèles basés sur des vecteurs d'ordres auto-régressifs croissants (construits à l'aide de valeurs successives de la série brute) sont également élaborés à l'aide de Perceptrons. Les résultats sont comparés à la figure 3. Les erreurs des modèles classiques sont représentées par des astérisques tandis que le niveau d'erreur obtenu par notre méthode est indiqué par une droite horizontale.

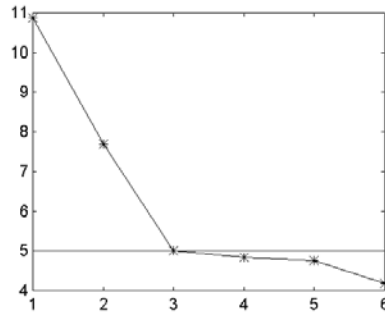


Figure 3. Comparaison entre notre méthode et la méthode classique

Les résultats obtenus par la méthode proposée sont donc comparables à ceux obtenus par des modèles construits par essais et erreurs, ce qui était le but recherché.

4. APPLICATION A L'INDICE SBF 250

Un exemple intéressant de série temporelle dans le domaine de la finance est l'indice SBF 250. D'un point de vue théorique, il n'est pas du tout sûr qu'une prédiction soit possible. En effet, une série temporelle peut être décomposée en une partie déterministe et une partie aléatoire. Dans le cas de l'indice SBF 250, la partie aléatoire est probablement déterminante.

Pour réussir à déterminer les variations de cet indice choisi comme objectif de recherche, d'autres variables pouvant influencer ses fluctuations doivent être sélectionnées. Nous avons choisi trois indices boursiers internationaux : le S&P500, le Topix et le FTSE100, respectivement américain, japonais et anglais ; deux taux de change : Dollar/Mark et Dollar/Yen ; et enfin, deux taux d'intérêt américains : le T-Bills 3 mois et le US Treasury Constant Maturity 10 years.

*Estimation de la dimension intrinsèque d'une série temporelle et
 prédiction par une méthode de projection* 43

Toutes les données collectées sont journalières sur 5 ans (du 01/06/92 au 01/12/97), ce qui permet d'avoir un historique significatif.

Une question se pose : connaissant toutes les données au jour t , comment prévoir le SBF en $t+1$?

Pour définir les relations qui peuvent exister entre l'indice français et les autres variables choisies, une co-intégration sur les prix de ces données semblent nécessaire. Cette co-intégration permet de calculer les résidus du SBF. Le résidu R_t au temps t est défini par l'écart entre la valeur théorique SBF_{t+1} et l'approximation \hat{SBF}_{t+1} faite de cette même valeur par le modèle :

$$R_t = SBF_{t+1} - \hat{SBF}_{t+1} = SBF_{t+1} - (\alpha_t + \sum_{i=1}^7 \beta_{t,i} \cdot I_{t,i}) \quad (2)$$

avec $I_{t,i}$ ($1 \leq i \leq 7$) les 7 variables initiales prises en t .

Par la suite, la recherche des comportements systématiques est focalisée sur la prévision de ces résidus, ou plus exactement sur la prévision des rendements journaliers de ces résidus. En effet, il est plus utile pour quelqu'un désireux de jouer sur le marché, de prévoir les fluctuations de la bourse plutôt que son niveau. Prévoir que le niveau du SBF demain est proche de celui d'aujourd'hui est trivial. Par contre, déterminer si la bourse va monter ou descendre est beaucoup plus complexe. Le rendement journalier ρ_t du résidu R_t à l'instant t est défini par :

$$\rho_t = \frac{R_t - R_{t-1}}{R_{t-1}} \quad (3)$$

Suivant en cela Refenes et al. [12], nous allons utiliser des indicateurs techniques directement issus des rendements des résidus :

- $\rho_t, \rho_{t-10}, \rho_{t-20}, \rho_{t-40}$: rendements ;
- $\rho_t - \rho_{t-5}, \rho_{t-5} - \rho_{t-10}, \rho_{t-10} - \rho_{t-15}, \rho_{t-15} - \rho_{t-20}$: différences de rendements ;
- K(20), K(40) : oscillateurs ;
- MM(10), MM(50) : moyennes mobiles ;
- MME(10), MME(50) : moyennes mobiles exp. ;
- ρ -MME(10), ρ -MME(50)
- MME(10)-MME(50).

Si nous effectuons une analyse en composantes principales sur ces 17 indicateurs, nous constatons que 99,72% de la variance totale se retrouve dans les onze premières composantes principales, permettant ainsi de supprimer 6 indicateurs techniques sans perte d'information.

La variable cible, que nous désirons prédire, est une variable de prévision sur 5 jours :

$$C_{t+6} = \sqrt[5]{\frac{R_{t+6}}{R_{t+1}}} \quad (4)$$

Cette série temporelle est représentée à la figure 4.

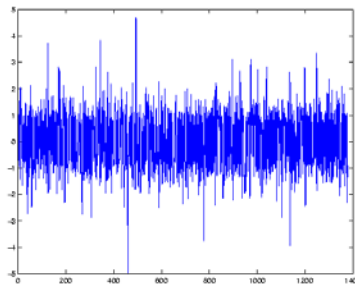


Figure 4. Variable de prévision sur 5 jours

Elle doit être prédite en utilisant les 11 indicateurs sélectionnés par l'analyse en composantes principales. Nous utiliserons, comme approximateur, un réseau à Fonctions Radiales de Base (RBF) qui permet d'approximer une fonction avec la précision voulue [13]. Ce réseau sera entraîné avec 1000 données d'apprentissage et ensuite testé sur 100 données de validation. Nous ne nous intéresserons qu'au signe de la prédiction, et nous le comparerons avec le signe exact de la variable de prédiction.

Les meilleurs résultats obtenus sont : 60,2% de bonnes approximations de la variation de la série pour l'ensemble d'apprentissage, et 48 % pour l'ensemble de test. Ce résultat est de fait très mauvais : nous faisons moins bien qu'un tirage aléatoire !

Si, par contre, nous calculons la dimension fractale de cet ensemble de données, nous obtenons 5. On peut donc appliquer l'algorithme VQP pour projeter les 11 dimensions de départ dans un espace de dimension 5. Ceci étant fait, nous utiliserons un nouveau réseau RBF pour approximer la

*Estimation de la dimension intrinsèque d'une série temporelle et
prédiction par une méthode de projection* 45

variable de prédiction. Les résultats obtenus sont bien meilleurs : 61% de bonnes approximations pour l'apprentissage et 57% pour l'ensemble de test.

De meilleurs résultats ont été obtenus en utilisant un Perceptron Multicouches (MLP) comme approximateur (+ de 62% de bonnes approximations pour la validation). Malheureusement, ces résultats sont sujets à caution, car la qualité de l'apprentissage de ces réseaux ne peut pas être assurée. Tantôt on obtient des minima locaux, tantôt un surapprentissage se manifeste.

5. CONCLUSION

La méthode que nous proposons pour la détermination du meilleur vecteur auto-régressif donne des résultats très satisfaisants. En effet, la qualité de la prédiction obtenue est soit équivalente soit très légèrement inférieure à celle obtenue par les autres méthodes. Mais en contrepartie, la procédure utilisée est plus systématique ; nous pouvons nous passer des différents essais et erreurs inhérents à la détermination des paramètres dans les méthodes neuronales, dont la fiabilité n'est pas assurée. De plus, la détermination du vecteur auto-régressif est tout à fait indépendante de la méthode de prédiction utilisée en aval.

Plusieurs voies peuvent être envisagées pour améliorer cette méthode. La première est une étude approfondie de la détermination de la dimension fractale d'une série, celle-ci n'étant pas toujours facile à calculer. Ensuite, la projection d'un espace de grande dimension vers un espace de dimension plus faible étant une opération délicate, une étude plus approfondie pourrait probablement améliorer la qualité de la prédiction.

La question de la prédictabilité d'une série comme l'indice SBF 250 reste posée ; comme nous l'avons vu, les résultats sont mitigés mais prometteurs. Nous ne sommes pas certains d'avoir capté toute la dynamique sous-jacente au processus complexe qui régit cette série ; néanmoins, la méthode présentée dans notre étude pourrait être utilisée comme un élément, parmi d'autres, à prendre en considération pour essayer de prédire une série telle que l'indice SBF 250.

6. RÉFÉRENCES

- [1] K. T. Alligood, T. D. Sauer and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems* (N-Y: Springer-Verlag, 1997), pp. 537-556.
- [2] G.E.P. Box and G. Jenkins, *Time Series analysis: Forecasting and Control* (Cambridge University Press, 1976).
- [3] Pierre Demartines, *Analyse de Données par Réseaux de Neurones Auto-Organisés* Institut Polytechnique de Grenoble, Thèse de doctorat (1992)
- [4] P. Grassberger and I. Procaccia, "Measuring the strangeness of strange attractors," *Physica Vol. D56*, 1983, pp. 189-208.
- [5] L. Ljung, *System Identification - Theory for User* (Prentice-Hall, 1987)
- [6] F. Tackens, "On the numerical Determination of the dimension of an attractor," *Lecture Notes in Mathematics No. 1125*, 1985, pp. 99-106.
- [7] J. Theiler, "Statistical precision of dimension estimators," *Phys. Rev. Vol A41*, 1990, pp. 3038-3051.
- [8] A. S. Weigend and N.A. Gershenfeld *Times Series Prediction: Forcasting the future and Understanding the Past*, (Addison-Wesley Publishing Company, 1994).
- [9] Xiangdong He and Haruhiko Asada, "A new method for Identifying Orders of Input-Output Models for Nonlinear Dynamic Systems," *Proc. of the American Control Conf., S.F., California*, 1993 pp. 2520-2523.
- [10] Burgess A.N., "Non-linear Model Identification and Statistical Significance Tests and their Application to Financial Modelling," in *Artificial Neural Networks, Inst. Elect. Eng. Conf., June 1995*.
- [11] Fama Eugène, "Efficient Capital Markets : A Review of Theory and Empirical Work" *Journal of Finance, XXV*, No 2 (March 1970).
- [12] Refenes A. N., Burgess A.N. and Bentz Y., "Neural Networks in Financial Engineering: A Study in Methodology", *IEEE Transactions on Neural Networks, vol. 8*, November 1997.
- [13] Verleysen M., Hlavačková K., "An optimized RBF network for approximation of functions", *European Symposium on Artificial Neural Networks, Bruxelles (Belgique)*, 20-22 avril 1994, D factio publications.