ACSEG'2004 proceedings – Connectionist Approaches in Economics and Management Sciences
Lille (France), 18-19 November 2004, pp. 30-38

# CLASSIFICATION ET PREDICTION FONCTIONNELLES D'ACTIFS BOURSIERS EN "TICK DATA"

SIMON DABLEMONT, MICHEL VERLEYSEN *

Université catholique de Louvain, Machine Learning Group, DICE
3, Place du Levant, B-1348 Louvain-la-Neuve - BELGIUM
Tel : +32 10 47 25 51 - Fax : +32 10 47 25 98
{dablemon, verleysen}@dice.ucl.ac.be

**Abstract**— *A functional method for time series forecasting is presented. Based on the splitting of the past dynamics into clusters, local models are built to capture the possible evolution of the series given the last known values. A probabilistic model is used to combine the local predictions. The method can be applied to any time series prediction problem, but is particularly suited to data showing non-linear dependencies and cluster effects, as many financial series do. The method is applied to the prediction of "tick data".*

**Keywords**— *functional, "tick data", prediction, cluster, neuronal*

**Résumé**— *Nous présentons une méthode d'analyse fonctionnelle pour la prédiction de séries temporelles. A partir de la décomposition des dynamiques en clusters, nous construisons des modèles locaux pour la prédiction de l'évolution des séries à partir des données du passé. Un modèle probabiliste est utilisé pour la combinaison des prédictions locales. Cette méthode peut être appliquée à tout problème de prédiction de séries temporelles mais elle est particulièrement adaptée aux données avec des dépendances non linéaires et des clusters, tels que les séries financières. La méthode a été appliquée à la prédiction des séries boursières de données en "tick par tick".*

**Mots clefs**— *fonctionnel, "tick par tick", prédiction, cluster, neuronal*

## 1 Introduction

Les "tick data" ou données en "tick par tick" sont les valeurs instantanées des transactions sur un actif boursier. Pour ces "tick data" en horizon très court, les modèles financiers théoriques construits à partir de l'hypothèse d'absence d'opportunité d'arbitrage et des marchés efficients ne sont pas vérifiés. Il devrait donc être possible de construire d'autres modèles non linéaires, directement à partir des données hautes fréquences, sans hypothèses initiales, qui seraient susceptibles, dans la limite des possibilités, de fournir une

*Michel Verleysen est Maitre de Recherches du Fond National de la Recherche Scientifique (FNRS)

prédiction des valeurs futures de ces actifs. Par exemple, il serait intéressant d'avoir une prédiction des "high" et "low" ainsi que des moments d'apparition de ces extrema. Pour les "tick data" le futur c'est quelques heures et le passé 1 à 2 jours ; au delà la bourse a intégré toutes les informations disponibles et l'hypothèse des marchés efficients redevient valable.

Une possibilité de modélisation consiste à séparer les observations passées (connues) en classes de dynamiques spécifiques et à construire des modèles dits "locaux" dans chacune des classes ainsi créées. La technique des modèles locaux est utilisée dans de nombreux domaines, car elle permet d'utiliser des modèles simples, y compris des modèles linéaires ou ARMA/GARCH, tout en permettant une modélisation globale non-linéaire performante. Cette technique a déjà été utilisée par exemple dans (S.Dablemont et al. (2003)) pour modéliser des valeurs journalières d'un indice boursier.

Le but de ce papier est de montrer comment des données "tick data" d'actifs boursiers peuvent être modélisées par des techniques fonctionnelles. En particulier, après avoir montré (section 2) comment modéliser des données fonctionnelles, la section 3 montrera comment créer des classes dans un ensemble de données connues ; la section 4 montrera comment classer des nouvelles données non-utilisées lors de la création des classes. Les sections 5 et 6 décriront brièvement respectivement la possibilité de créer des modèles locaux à l'intérieur des classes ainsi créées, et l'utilisation de ces modèles à des fins de prédiction. Enfin, la section 7 montrera un exemple d'application des techniques développées sur des "tick data" d'actifs boursiers.

# 2 Représentation des "tick data"

Les "tick data" peuvent être considérés comme des observations d'un processus temporel continu. Soit $i = [1, \cdots, N]$ l'indice identifiant le jour d'observation d'un actif spécifique pendant l'ouverture de la bourse (par ex. de 09.30 hr à 16.00 hr). La représentation continue de la valeur de l'actif sera alors :

$$y_i(t) = g_i(t) + \epsilon_i(t), \tag{1}$$

où $y_i(t)$ représente la valeur observée de l'actif à l'instant $t$ du jour $i$, $g_i(t)$ sa vraie valeur et $\varepsilon_i(t)$ l'erreur de mesure supposée centrée et de variance fixe ("bid ask spread" et "outliers"). Les $y_i(t)$ ne sont connus qu'aux moments des transactions, notés $[t_{i1}, \cdots, t_{in_i}]$. Les instants d'observation, ainsi que leur nombre $n_i$, diffèrent de façon générale entre les différents jours d'observation $i$ de l'actifs. Les observations aux instants $t$ du jour $i$ sont alors regroupées en un vecteur défini par

$$\mathbf{y}_i = [y_i(t_{i1}), y_i(t_{i2}), \cdots y_i(t_{in_i})]^T. \tag{2}$$

Définissant de manière similaire les vecteurs $\mathbf{g}_i$ et $\epsilon_i$, les vecteurs d'observations $\mathbf{y}_i$ de tous les jours $i$ sont alors donnés par :

$$\mathbf{y}_i = \mathbf{g}_i + \epsilon_i, \qquad i = 1, \cdots, N. \tag{3}$$

Le principe de la modélisation fonctionnelle (J.O.Ramsay, B.W.Silverman (2002)) consiste alors à représenter les fonctions, dans ce cas $\mathbf{g}_i$, comme une combinaison de fonctions

splines prédéterminées. Par exemple, nous utiliserons des splines cubiques avec un noeud interne, définis par $s_1(t) = 1$, $s_2(t) = t$, $s_3(t) = t^2$, $s_4(t) = t^3$, $s_5(t) = (t - \tau)^3_+$, où $\tau$ est la position du noeud interne.
De manière plus générale, si

$$\mathbf{s}(t) = \Big(s_1(t), \cdots, s_p(t)\Big)^T \tag{4}$$

est la base de $p$ splines utilisées ($p = 5$ dans l'exemple ci-dessus), les fonctions $g_i(t)$ peuvent être exprimées sous la forme

$$g_i(t) = \mathbf{s}^T(t)\gamma_i \tag{5}$$

où $\gamma_i$ est le vecteur des coefficients des splines pour la fonction $g_i(t)$.
La modélisation fonctionnelle des "tick data" apporte un avantage par rapport à une technique qui pourrait paraître plus simple,le rééchantillonnage à intervalles réguliers de ces données afin d'obtenir des données "intraday". (Y.Ait-Sahalia, P.A.Myland (2003)). En effet, la variable fonctionnelle $y_i(t)$ pourrait être échantillonnée sur une fine grille de $p$ points temporels pour créer le vecteur $\mathbf{Y}_i$, en enlevant donc l'aspect fonctionnel du problème, mais cette approche pose de sérieux problèmes. D'abord, elle nécessite de modéliser un vecteur de coefficients de haute dimension, ce qui peut conduire à des estimations très instables. Ensuite, dans beaucoup d'applications, on ne dispose que d'un nombre limité de mesures pour chaque fonction. Dans le cas des séries boursières, les transactions journalières sont observées à des instants différents et en nombre variable. Pour de telles données il n'est pas possible de créer des prédicteurs de dimension finie par simple discrétisation. D'autre part, le rééchantillonnage devrait obligatoirement se faire à une fréquence assez basse, sous peine d'avoir à extrapoler (plutôt qu'interpoler) pendant les périodes où peu de données sont disponibles. L'ensemble des données n'est alors pas exploité. De plus, le fait qu'un intervalle de temps regroupe peu ou beaucoup de données est une information importante, perdue lors d'un rééchantillonnage.
Un modèle construit sur des données fonctionnelles, tel que le modèle de clustering détaillé dans le section suivante, permet de remédier à cet inconvénient.

# 3 Clustering des données fonctionnelles

La modélisation des "tick data", représentées sous forme fonctionnelle $y_i(t)$, peut se faire en créant des classes dans l'espace des $y_i(t)$. De cette façon, chaque classe représentera un ensemble homogène de données haute fréquence de même dynamique. On pourra alors songer, dans la suite, à construire des modèles de prédiction simples à l'intérieur de chaque classe. On parle alors de modèles locaux. Il faut noter que la classification envisagée ci-dessus est non-supervisée (il s'agit de "clustering") : a priori, on ne connaît pas les classes, il faut les créer à partir des données connues, à savoir les $\mathbf{y}_i$.

Trois méthodes peuvent être envisagées pour créer ces clusters. On pourrait songer à appliquer directement les méthodes classiques de clustering, telles que les k-means, le competitive learning ou encore les cartes auto-organisatrices de Kohonen, sur les données $\mathbf{y}_i$. Malheureusement, ces données sont des vecteurs de dimensions différentes, rendant impossible l'utilisation directe de ces méthodes. De plus, même si les dimensions correspondaient, les instants d'échantillonnage seraient différents entre les données, ce qui rend ces techniques inadéquates lorsqu'elles sont utilisées sur les données brutes.

Une solution est alors d'appliquer les méthodes classiques de clustering sur les coefficients $\gamma_i$ des splines. Les vecteurs $\gamma_i$ ont en effet tous la même dimension et leurs composantes respectives ont la même signification. Les coefficients $\gamma_i$ étant une image fidèle des données fonctionnelles $y_i(t)$, effectuer le clustering sur les coefficients ou sur les données fonctionnelles est donc indifférent. C'est cette méthode qui sera utilisée dans la suite de cet article. Néanmoins, il faut noter que l'estimation, de façon individuelle pour chaque jour $i$, des coefficients $\gamma_i$ des splines conduit à une qualité d'estimation très différente entre les coefficients. En effet, il est évident qu'une donnée $y_i(t)$ connue à travers un vecteur d'observations $\mathbf{y}_i$ de grande dimension donnera lieu à une estimation plus fiable que si le nombre d'observations est très faible. De même, chacun des coefficients du vecteur $\gamma_i$ peut lui-même être estimé de manière très différente, suivant que les observations connues formant le vecteur $\mathbf{y}_i$ sont espacées de façon régulière, concentrées dans le temps, etc. Or il se fait que les méthodes traditionnelles de clustering accordent "la même importance" à chacun des coefficients de chacune des observations à classer ; face à des données fonctionnelles échantillonnées de manière irrégulière comme les "tick data", elles vont donc avoir tendance à ne pas tenir compte du nombre, de la fréquence et de la fiabilité de chaque observation individuelle, ce qui est pourtant une information importante à la construction d'un modèle.
Une troisième façon de procéder, encore à l'état de développement, est alors de créer un modèle "semi-paramétrique". Faisons d'abord l'hypothèse que chaque observation fonctionnelle $y_i(t)$ est générée par un processus Gaussien $f(x|\theta)$ de paramètres $\theta$ inconnus (en l'occurrence moyenne et matrice de covariance du processus Gaussien). Une procédure bien connue en statistique, celle du maximum de vraisemblance, permet alors de trouver les paramètres inconnus qui maximisent la probabilité d'observer les $N$ observations à disposition, cette procédure est implémentée sous la forme d'un algorithme itératif d'optimisation, l'algorithme EM (Expectation-Maximisation). En pratique, faire l'hypothèse que chaque observation fonctionnelle $y_i(t)$ est générée par un seul processus Gaussien $f(x|\theta)$ est trop restrictive ; le modèle choisi est alors plus général, sous la forme d'un mélange de $G$ Gaussiennes $f_k(x|\theta_k) \quad k = 1, \cdots, G$ pondérés par $\pi_k$, les probabilités à priori d'appartenance d'une observation à chaque cluster. A nouveau, l'algorithme EM permet d'estimer, étant données les $N$ observations, l'ensemble des paramètres $\{\theta_1, \cdots, \theta_G, \pi_1, \cdots, \pi_G\}$ donnant lieu à la plus grande probabilité de ces observations ; la fonction de vraisemblance, maximisée par l'algorithme EM, s'écrit alors

$$L(\theta_1, \cdots, \theta_G; \pi_1, \cdots, \pi_G | \mathbf{y}_1, \cdots, \mathbf{y}_N) = \prod_{i=1}^{N} \sum_{k=1}^{G} \pi_k f_k(\mathbf{y}_i|\theta_k) \tag{6}$$

Une fois le modèle (6) estimé, nous sommes en présence de $G$ clusters Gaussiens, de paramètres connus. Il est donc facile d'estimer de quel cluster une nouvelle observation $\mathbf{y}_i$ provient le plus probablement : il faut trouver la classe dont la probabilité est la plus grande, étant donnée l'observation $\mathbf{y}_i$. Cette probabilité est donnée par

$$P(classe = k|\mathbf{y}_i) = \frac{f_k(\mathbf{y}_i|\theta_k)\pi_k}{\sum_{j=1}^{G} f_j(\mathbf{y}_i|\theta_j)\pi_j} \tag{7}$$

La classe $k$ qui maximise (7) est donc attribuée à l'observation $\mathbf{y}_i$.

# 4 Modélisation fonctionnelle par réseaux neuronaux

Les sections ci-dessus décrivaient comment réaliser un clustering de données fonctionnelles. En réalité, ceci n'est pas suffisant pour atteindre l'objectif, à savoir la prédiction des "tick data", ou plus précisément de leurs valeurs "high" et "low", des instants d'apparition de ces dernières, etc. Un modèle de prédiction fonctionnelle doit alors être utilisé. Les étapes ci-dessus remplaçant le clustering classique de données vectorielles par un clustering de fonctions, n'importe quelle méthode de prédiction utilisant le principe du clustering peut être utilisée ; seules les opérations de clustering doivent y être remplacées par les extensions décrites ci-dessus.
Le modèle de prédiction utilisé est largement inspiré de (S.Dablemont et al. (2003)). En résumé, il s'agit de créer deux espaces (ici fonctionnels), sur chacun desquels le clustering sera réalisé. De façon analogue à un modèle classique de prédiction, on crée tout d'abord des "régresseurs" composés, par exemple, des données fonctionnelles de deux jours consécutifs connus de transactions. L'ensemble de ces régresseurs (notés IN) est alors quantifié en un nombre prédéterminé de clusters suivant les méthodes décrites ci-dessus. De façon analogue, des données étendues (notées OUT) sont construites, composées chacune en agrégeant un régresseur tel que défini ci-dessus et la donnée fonctionnelle qui suit immédiatement. Ces fonctions étendues sont elles aussi quantifiées en un nombre prédéterminé de clusters.
Les relations de dépendance entre les classes dans chacun des deux espaces fonctionnels ainsi crées sont ensuite modélisées à travers une table de fréquences empiriques ; en d'autres mots, on crée une table dont la dimension (lignes/colonnes) vaut respectivement le nombre de clusters dans les espaces IN et OUT, et on mesure les fréquences d'apparition d'une des fonctions OUT soumise à la condition d'être en présence du régresseur IN correspondant.
Enfin, dans chacune des classes de l'espace des fonctions étendues OUT, un réseau RBFN (Radial-Basis Function Network) (J. Park, I. Sandberg, (1993)) est construit pour modéliser les valeurs à prédire en fonction des valeurs passées. Bien entendu, tout comme le clustering, la prédiction par RBFN ne peut pas, en pratique, être effectuée directement sur des données fonctionnelles ; ce sont donc les coefficients des splines, issus de (5) ou (7) appliqué à chacun des deux ensembles de fonctions IN et OUT, qui sont utilisés respectivement comme entrées et sorties des réseaux RBFN. Ces derniers sont donc utilisés de manière classique, pour prédire les coefficients des splines représentant les fonctions étendues OUT, dont une partie constitute la donnée fonctionnelle à prédire. La procédure utilisée pour l'apprentissage des modèles RBFN est décrite dans (N. Benoudjit, M. Verleysen, (2003)).

# 5 Prédiction fonctionnelle

La section ci-dessus décrit comment construire un modèle fonctionnel de prédiction. Une fois le modèle construit, c'est à dire une fois les deux clusterings réalisés (données IN et OUT), le tableau de fréquences empiriques créé et l'apprentissage des modèles RBFN effectué (un par classe de l'espace des fonctions étendues OUT), le modèle global peut être utilisé. Pour ce faire, une nouvelle donnée (fonctionnelle) connue (par exemple les deux derniers jours connus de transactions) est tout d'abord classée dans un de clusters de l'espace des régresseurs IN. Ensuite, pour chacun des clusters dans l'espace des fonctions étendues OUT, le modèle RBFN correspondant est utilisé pour prédire une sortie

correspondant à l'entrée connue. Il y a donc à cette étape autant de prédictions qu'il y a de clusters dans l'espace OUT. Enfin, ces prédictions sont sommées après pondération par les fréquences empiriques d'apparition des clusters OUT correspondants, soumis à la condition d'avoir la donnée IN connue (en d'autres mots, en utilisant la ligne adéquate du tableau des fréquences empiriques).

# 6 Résultats

Cette section illustre l'application des méthodes fonctionnelles décrites ci-dessus à un actif particulier, à savoir le titre IBM coté à la bourse de N.Y. (NYSE), pendant la période de janvier 1997 à mai 1997. Le titre IBM étant très liquide, nous avions plus de 3.000 transactions par jour

La Fig. 1 montre, en haut, les cotations et en bas, les volumes des transactions correspondantes, sur une journée d'ouverture du NYSE. La Fig. 2 détaille les distributions des transactions par demi-heure pour un jour donné. Chaque point représente une transaction pendant la période de 09.30 hr à 16.00 hr. On y voit très clairement un nombre beaucoup plus élevé de transactions juste après l'ouverture (prise de position) et juste avant la fermeture de la bourse (cloture de position), par exemple en comparaison des transactions effectuées entre 12 :30 et 13 heures. .

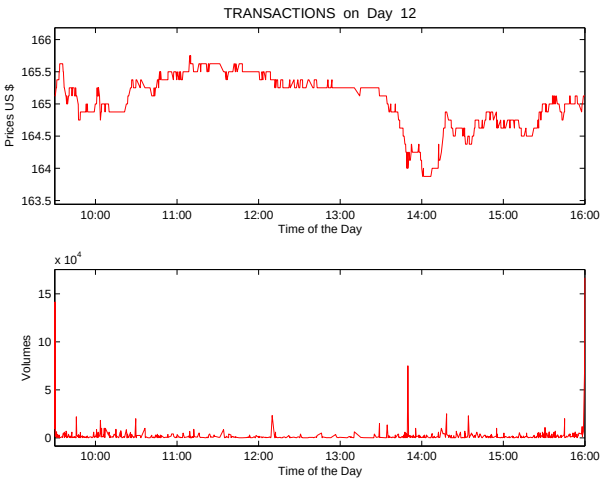


FIG. 1 – Cotations (haut) et Volumes (bas)

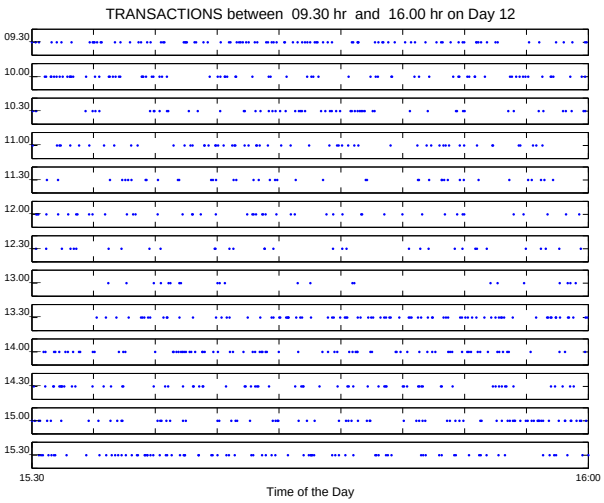


FIG. 2 – Distributions

La Fig. 3 montre quatre journées successives de cotation du titre IBM, ainsi que le résultat des splines de lissage construites sur base des "tick data" des transactions. On y voit un lissage adéquat, et une bonne représentation des valeurs des transactions grace à l'approximation par splines.

Les Fig. 4 et Fig. 5 montrent un exemple de clustering réalisé sur les splines de type "régresseur" formés de deux jours consécutifs de transactions. Les deux figures montrent l'ensemble des données fonctionnelles (courbes continues) affectées à chacun des deux clusters, ainsi que le centroide correspondant (courbe pointillée).

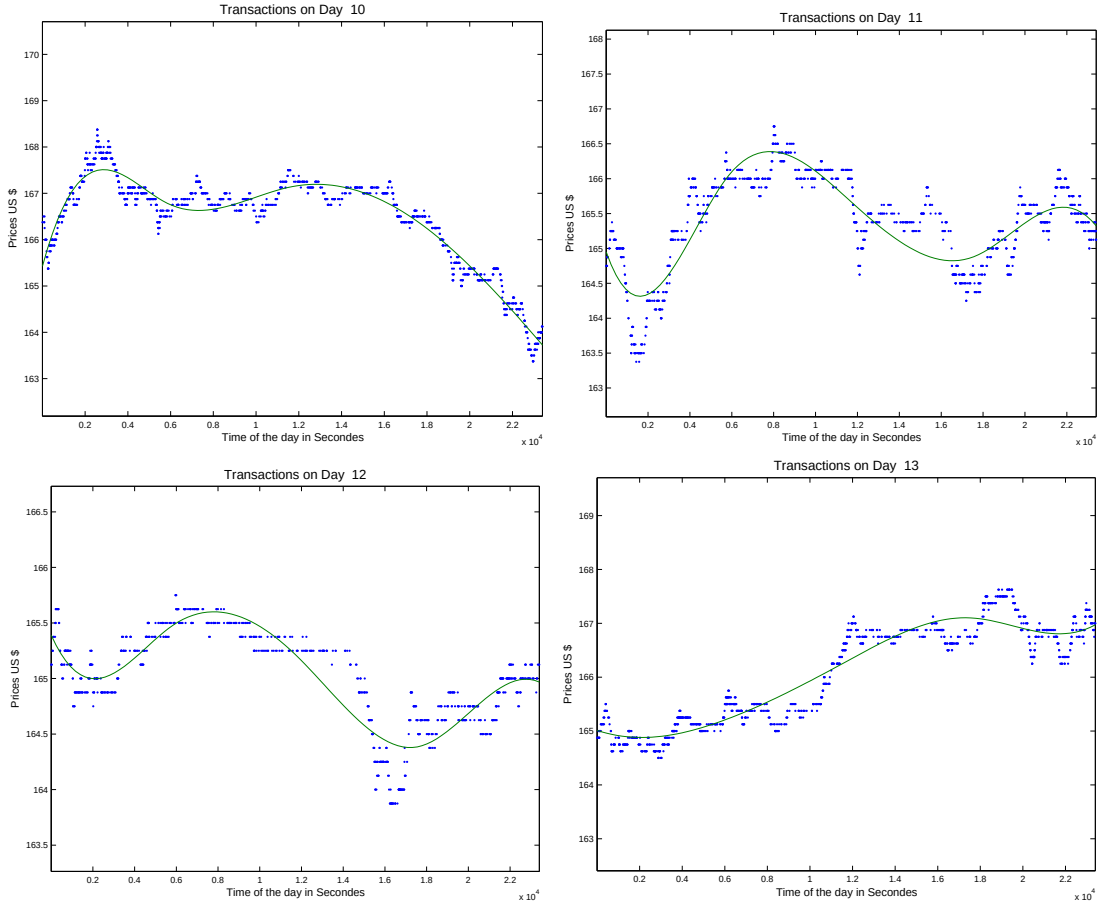


FIG. 3 – Quatre journées de transactions du titre IBM (traits pointillés), ainsi que les splines de lissage obtenus (trait pleins)

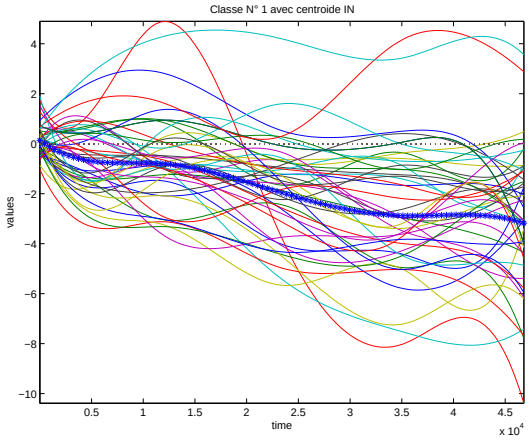


FIG. 4 – Cluster 1 des régresseurs IN

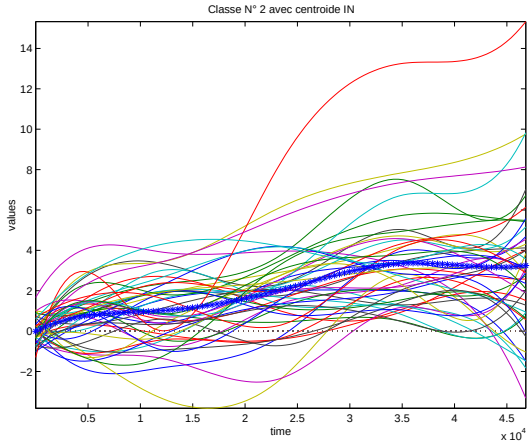


FIG. 5 – Cluster 2 des régresseurs IN

## 6.1 Prédiction

L'exemple de prédiction illustré ici consiste à prédire les splines des transactions pour le jour $J$ pour la période de 10 :30 hr à 13 :30 hr, à partir des transactions traitées les deux jours précédents ($J-2$ et $J-1$) et au début du jour $J$ depuis l'ouverture de la

bourse à 09 :30 hr jusqu'à 10 :30 hr, avec 2 clusters. La Fig. 6 montre quatre exemples de prédiction, superposées aux vraies valeurs et aux splines de lissage correspondants (les vraies valeurs et les splines étant bien entendu inconnus du modèle utilisé). On y voit qualitativement le bon comportement de la méthode fonctionnelle, qui permet de prédire une courbe correspondant relativement bien aux données réelles.

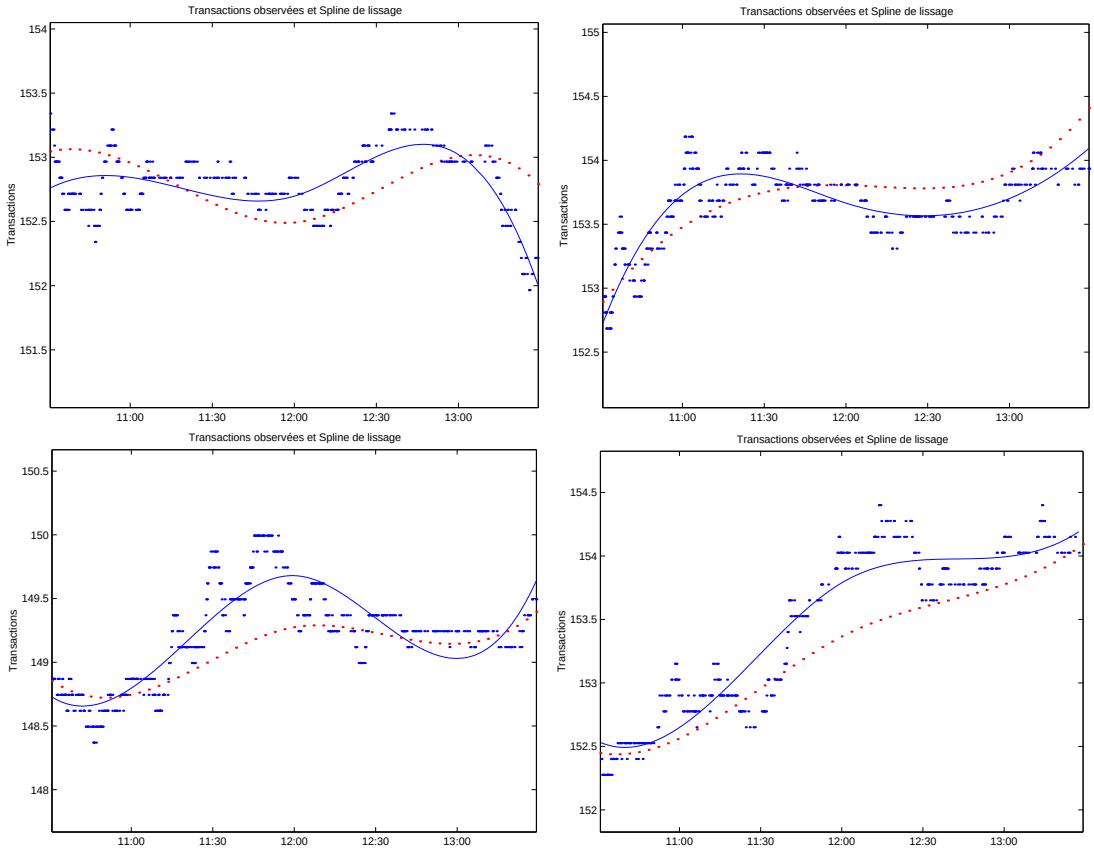


FIG. 6 – Quatre exemples de prediction des valeurs du titre IBM. Points : valeurs réelles ; trait continu : splines de lissage ; trait pointillé : prédiction obtenue par le modèle.

# 7 Conclusion

Les données "tick data" d'actifs boursiers présentent des particularités dont il faut tenir compte lors de leur analyse et prédiction. En particulier, leur échantillonnage dans le temps est souvent fort irrégulier et très différent d'un jour à l'autre. L'utilisation de méthodes traditionnelles d'analyse de données ne permet pas de prendre en compte ces particularités et donne donc des résultats médiocres. Ce papier montre comment utiliser des méthodes d'analyse fonctionnelle sur des données en "tick par tick". En particulier, il montre comment utiliser de façon traditionnelle l'approche fonctionnelle, en lissant les données connues par des splines. Il introduit également un modèle semi-paramétrique fonctionnel, sous la forme d'un mélange de Gaussiennes ; dans lequel les paramètres sont estimés par une procédure de maximum de vraisemblance des données observées. L'utilisation de ces modèles permet de créer des "clusters" dans l'espace des données haute fréquence. De cette manière, des modèles locaux de prédiction peuvent alors être construits

sur des ensembles homogènes de "tick data". La méthode de prédiction utilisée repose sur une modélisation des probabilités conditionnelles d'apparition d'une courbe (à prédire), en fonction des dernières données (fonctionnelles) connues, ainsi qu'une modélisation non-linéaire locale à l'intérieur de chaque cluster par un réseaux RBFN. La méthode est illustrée sur les données en "tick par tick" de l'action IBM.

# Références

[1] S. Dablemont, G. Simon, A. Lendasse, A. Ruttiens, M. Verleysen,"Prédiction de séries temporelles financières par double carte de Kohonen et modèles RBFN locaux : application à la prédiction de l'indice boursier DAX30", ACSEG 2003, Connectionist Approaches in Economics and Management Sciences, Nantes (France), 20-21 November 2003, pp. 153-164.

[2] J.O. Ramsay, B.W. Silverman,"Applied Functional Data Analysis : Methods and Case Studies". Springer-Verlag, New York, 2002.

[3] Y. Ait-Sahalia, P. A. Myland,"The effects of random and discrete sampling when estimating continuous-time diffusions", Econometrica, vol. 71, pp. 483-549, March 2003.

[4] J. Park, I. Sandberg,"Approximation and radial basis function networks", Neural Comput. vol.5 , pp. 305-316, 1993.

[5] N. Benoudjit, M. Verleysen, "On the kernel widths in Radial-Basis Function Networks", Neural Processing Letters, Kluwer academic pub., vol. 18, no. 2, pp. 139-154, October 2003.