Modelling and Forecasting financial time series of «tick data» by functional analysis and neural networks

S. DABLEMONT, S. VAN BELLEGEM, M. 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
E-mail : {dablemont, verleysen}@dice.ucl.ac.be
vanbellegem@stat.ucl.ac.be

Subjects
Finances , Neural Networks, Forecasting, Nonlinear Time Series Models, Tick Data

Abstract
The analysis of financial time series is of primary importance in the economic world. This paper deals with a data-driven empirical analysis of financial time series. The goal is to obtain insights into the dynamics of series and out-of-sample forecasting.

In this paper we present a forecasting method based on an empirical functional analysis of the past of series.

An originality of this method is that it does not make the assumption that a single model is able to capture the dynamics of the whole series. On the contrary, it splits the past of the series into clusters, and generates a specific local neural model for each of them. The local models are then combined in a probabilistic way, according to the distribution of the series in the past.

This forecasting method can be applied to any time series forecasting problem, but is particularly suited for data showing nonlinear dependencies, cluster effects and observed at irregularly and randomly spaced times like high-frequency financial time series do. One way to overcome the irregular and random sampling of "tick-data" is to resample them at low-frequency, as it is done with "Intraday". However, even with optimal resampling using say five minute returns when transactions are recorded every second, a vast amount of data is discarded, in contradiction to basic statistical principles. Thus modelling the noise and using all the data is a better solution, even if one misspecifies the noise distribution.

The method is applied to the forecasting of financial time series of «tick data» of assets on a short horizon in order to be useful for speculators

1 Introduction

The analysis of financial time series is of primary importance in the economic world. This paper deals with a data-driven empirical analysis of financial time series, the goal is to obtain insights into the dynamics of series and out-of-sample forecasting.

Forecasting future returns on assets is of obvious interest in empirical finance. If one were able to forecast tomorrow's returns on an asset with some degree of precision, one could use this information in an investment today. Unfortunately, we are seldom able to generate a very accurate prediction for asset returns.

Financial time series display typical nonlinear characteristics, it exists clusters within which returns and volatility display specific dynamic behavior. For this reason, we consider here nonlinear forecasting models, based on local analysis into clusters. Although financial theory does not provide many motivations for nonlinear models, analyzing data by nonlinear tools seems to be appropriate, and is at least as much informative as an analysis by more restrictive linear methods.

Time series of asset returns can be characterized as serial dependent. This is revealed by the presence of positive autocorrelation in squared returns, and sometimes in the returns too. The increased importance played by risk and uncertainty considerations in modern economic theory, has necessitated the development of new econometric time series techniques that allow for modelling of time varying means, variances and covariances.

Given the apparent lack of any structural dynamic economic theory explaining the variation in the second moment, econometricians have thus extended traditional time series tools such as AutoRegressive Moving Average (ARMA) models (Box and Jenkins, 1970) for the conditional means and equivalent models for the conditional variance. Indeed, the dynamics observed in the dispersion is clearly the dominating feature in the data. The most widespread modelling approach to capture these properties is to specify a dynamic model for the conditional means and the conditional variance, such as an ARMA-GARCH model or one of its various extensions (Engle, 1982), (Hamilton, 1994).

The Gaussian random walk paradigm - under the form of the diffusion geometric Wiener process - is the core of modelling of financial time series. Its robustness mostly suffices to keep it as the best foundation for any development in financial modelling, in addition to the fact that, in the long run, and with enough spaced out data, it is almost verified by the facts. Failures in its application are however well admitted on the (very) short term (market microstructure) (Fama, 1991), (Olsen and Dacorogna, 1992), (Franke et al., 2002). We claim that, to some extent, such failures are actually caused by the uniqueness of the modelling process.

The first breach in such a unique process has appeared with two-regime or switching processes (Diebold et al., 1994), which recognize that a return process could be originated by two different stochastic differential equations. But in such a case, the switch is governed by an exogenous cause (for example in the case of exchange rates, the occurrence of a central bank decision to modify its leading interest rate or to organize a huge buying or selling of its currency through major banks) .

Market practitioners (Engle, 1982), however, have always observed that financial markets can follow different behaviors over time, such as overreaction, mean reversion, etc, which look like succeeding each other with the passing of the time. Such observations would justify a rather fundamental divergence from the classic modelling foundations. That is, financial markets should not be modeled by a single process, but rather by a succession of different processes, even in absence of the exogenous causes retained by existing switching process. Such a multiple switching process should imply, first, the determination of a limited number of competitive sub-processes, and secondly, the identification of the factor(s) causing the switch from one to another sub-processes. The resulting model should not be Markovian, and, without doubt, would be hard to determine.

The aim of this paper is, as a first step, to at least empirically verify, with the help of functional clustering and neural networks, that a multiple switching process leads to better short term forecasting.

In this paper we present a forecasting method based on an empirical functional analysis of the past of series. An originality of this method is that it does not make the assumption that a single model is able to capture the dynamics of the whole series. On the contrary, it splits the past of the series into clusters, and generates a specific local neural model for each of them. The local models are then combined in a probabilistic way, according to the distribution of the series in the past.

This forecasting method can be applied to any time series forecasting problem, but is particularly suited for data showing nonlinear dependencies, cluster effects and observed at irregularly and randomly spaced times like high-frequency financial time series do.

One way to overcome the irregular and random sampling of "tick-data" is to resample them at low frequency, as it is done with "Intraday". However, even with optimal resampling using say five minute returns when transactions are recorded every second, a vast amount of data is discarded, in contradiction to basic statistical principles. Thus modelling the noise and using all the data is a better solution, even if one misspecifies the noise distribution (Ait-Sahalia and Myland, 2003). And, one way to get to this goal is by using *Functional Analysis* as done in this paper.

Further in this paper, we first describe how *Functional Analysis* can be applied to time series data (section 2), and briefly introduce the *Radial-Basis Functions Networks* we use as nonlinear models (section 3). Then, we describe the forecasting method itself (section 4), and illustrate its results on the IBM series of "tick data" (section 5).

2 Functional Modelling and Clustering

Our purpose is to realize the clustering of the observations into classes having homogenous properties in order to build nonlinear local forecasting models in each class.

When the observations are sparse, irregularly spaced, or occur at different time points for each subject, as with high-frequency financial series, standards statistical tools can not be used because we do not have the same number of observations for each series, and the observations are not at the same time-point. In this case it will be necessary to

represent these data by a fixed number of features. One way to get to this purpose is to smooth the rough data by projecting them onto a *functional basis*, for example cubic splines. Then, the coefficients of this projection may be used in a more standard way for clustering purposes.

In (section 2.1) we introduce the generic problem of *clustering*, in (section 2.2) we explain why we have to use this tool for high-frequency financial series, in (section 2.3) we define the *functional model* we use, in (section 2.4) we build the *likelihood function*, and in (section 2.5) we have the procedures to optimize the parameters .

2.1 Clustering

Cluster analysis consists in identifying groups in data; it is the dual form of discriminant analysis but in cluster analysis the group labels are not known a priori. It is an unsupervised process.

We assume that the observations $\{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$ are generated according to a mixture distribution with G clusters. Let $f_k(\mathbf{y}|\theta_k)$ be the density distribution function corresponding to cluster k, with parameters θ_k, and let $1_{\{k\}}(i)$ be the cluster membership (indicator function of cluster k) for the observation i where $1_{\{k\}}(i) = 1$ if y_i is a member of cluster k and 0 otherwise. The indicators are unknown and $1_{\{k\}}(i)$ is multinomial with parameters $[\pi_1, \cdots, \pi_G]$ and π_k is the probability that an observation belongs to cluster k.

We can estimate the parameters by maximizing the likelihood

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

The maximum likelihood corresponds to the most probable model, given the observations $\{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$.

Such model can be used in finite dimensional problems, but it is not appropriated to infinite dimensional data such as curves (Hastie et al., 2001). We could get around by discretizing the time interval, but generally the resulting data vectors are highly correlated and high-dimensional, and by resampling at low frequency we loose much information (Ait-Sahalia and Myland, 2003).

Another approach is to project each curve onto a finite-dimensional *basis* $\phi(x)$, and find the best projection of each curve onto this basis. The resulting basis coefficients can than be used as a finite-dimensional representation making it possible to use classical clustering methods on the basis coefficients (Ramsay, and Silverman, 1997).

These approaches can work well when every curve has been observed over the same fine grid of points, but they break down if the individual curves are sparsely sampled. The variance of the estimated basis coefficients is different for each individual because the curves are measured at different time points. And for sparse data sets many of the basis coefficients would have infinite variance, making it impossible to produce reasonable estimates.

In this case, we convert the original infinite dimensional problem into a finite dimensional one using basic functions and we use a *random effects model* for the coefficients (Rice 2001).

2.2 Modelling functional data

Modern financial data sets may contain tens of thousands of transactions per day stamped to the nearest second. The analysis of these data are complicated due to stochastic temporal spacing, diurnal patterns, prices discreteness, and complex temporal dependence.

Let data coming from an interval $[t_0 \cdots t_N]$. If we use very liquid series, like stocks, with observations onto the whole interval of time, we could realize the smoothing by splines and afterwards the clustering from the spline coefficients, in two separate steps. If we don't have enough data near the limits t_0 or t_N of the interval, the smoothing by splines will not be fine at these limits, we will have many outliers and the clustering will be very poor. If we use poorly liquid data with large fragment of curve without observations, like options, the smoothing could be very chaotic. In those cases we have to realize the smoothing of the observations and the clustering in the same iterative steps.

Thus, when the observations are sparse, irregularly spaced, or occur at different time points for each subject and moreover when only fragments of the function are available, with the Linear Discriminant Analysis (LDA), many of the basis coefficients would have infinite variance, making it impossible to produce reasonable estimates (Gareth et al., 2000), and similar problem arise with the clustering methods. In this case, we will use a random effects model for the coefficients and we will realize, in the same step, the estimation of the splines coefficients and the clustering (Gareth and Sugar, 2003).

2.3 Functional Clustering

We will use basis functions in order to convert the original infinite dimensional problem into a finite dimensional one, but instead of treating the basis coefficients as parameters and fitting a separate spline for each individual, we will use a random effects model for the coefficients. This procedure borrows 'information' across curves and produces far better results no matter how sparsely or irregularly the individual curves are sampled, provided that the total number of observations is large enough. Moreover, it automatically weights the estimated spline coefficients according to their variances, which is highly efficient because it requires fitting few parameters, and it can be used to produce estimates of individual curves that are optimal in terms of mean square errors.

Let $g_i(t)$ the true value for the curve i at time t and $\mathbf{g}_i$, $\mathbf{y}_i$ and $\boldsymbol{\epsilon}_i$ the vectors of true values, measurements and errors at times $t_{i1}, t_{i2}, \cdots, t_{in_i}$ We have got :

$$\mathbf{y}_i = \mathbf{g}_i + \boldsymbol{\epsilon}_i, \qquad i = 1, \cdots, N \ , \tag{2}$$

where N is the number of curves. The errors are assumed to have mean zero and uncorrelated with each other and with $\mathbf{g}_i$.
Let :

$$\boldsymbol{g}_i = \Big(g_i(t_1), \cdots, g_i(t_j), \cdots, g_i(t_{n_i})\Big)^T,$$

$$\boldsymbol{y}_i = \Big(y_i(t_1), \cdots, y_i(t_j), \cdots, y_i(t_{n_i})\Big)^T,$$

$$\boldsymbol{\epsilon}_i = \Big(\epsilon_i(t_1), \cdots, \epsilon_i(t_j), \cdots, \epsilon_i(t_{n_i})\Big)^T,$$

where t_j is the time-point for the observation j and n_i is the number of observations for the curve i.
For the true values $\mathbf{g}_i$, we use a functional basis for their representation, and we have :

$$g_i(t) = \mathbf{s}^T(t)\ \boldsymbol{\eta}_i\ , \tag{3}$$

where $\mathbf{s}(t)$ is the spline basis vector with q dimensions, and $\boldsymbol{\eta}_i$ is a vector of spline coefficients, ($\boldsymbol{\eta}_i$ is a Gaussian random variable). Let :

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

$$\boldsymbol{\eta}_i = \Big(\eta_{i1}, \cdots, \eta_{iq}\Big)^T,$$

If we used a "power cubic spline" with 1 knot, then $q = 5$ and we would have :

$$\mathbf{s}(t) = \Big(s_1(t), \cdots, s_5(t)\Big)^T,$$

with : $s_1(t) = 1$, $s_2(t) = t$, $s_3(t) = t^2$, $s_4(t) = t^3$, $s_5(t) = (t-\tau)^3_+$, where τ is the knot.

For the Gaussian coefficients $\boldsymbol{\eta}_i$ we have :

$$\boldsymbol{\eta}_i = \boldsymbol{\mu}_{\mathbf{z}_i} + \boldsymbol{\gamma}_i\ , \qquad \boldsymbol{\gamma} \sim N(\mathbf{0}, \boldsymbol{\Gamma})\ , \tag{4}$$

where $\mathbf{z}_i$ denotes the unknown cluster membership for the curve i, and it will be treated as missing data,

$$z_{ki} = \begin{cases} 1 & \text{if curve } i \text{ belongs to cluster } k, \\ 0 & \text{otherwise,} \end{cases}$$

then we have :

$$P\big(z_{ki} = 1\big) = \pi_{k|i}\ ,$$

$$\boldsymbol{\mu}_{\mathbf{z}_i} = \Big(\mu_{i1_{\mathbf{z}_i}}, \cdots, \mu_{iq_{\mathbf{z}_i}}\Big)^T,$$

$$\boldsymbol{\mu}_{\mathbf{z}_i} = \Big\{(\mu_{(1)})^{z_{i1}} \cdots (\mu_{(k)})^{z_{ik}} \cdots (\mu_{(G)})^{z_{iG}}\Big\}\ ,$$

$$\boldsymbol{\gamma}_i = \Big(\gamma_{i1}, \cdots, \gamma_{iq}\Big)^T.$$

We have split $\boldsymbol{\eta}_i$ into two terms, $\boldsymbol{\mu}_{z_i}$ represents the centroid of the cluster and $\boldsymbol{\gamma}_i$ represents the curve in its cluster. Also in the same way, we can represent the centroid of the cluster from the global mean of the population by :

$$\boldsymbol{\mu}_k = \boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k\ , \tag{5}$$

where $\boldsymbol{\lambda}_0$ is a $(q, 1)$ vector, and $\boldsymbol{\alpha}_k$ $(h, 1)$, $\boldsymbol{\Lambda}$ is (q, h) matrix, with $h \leq \min(q, G-1)$.

$$\boldsymbol{\lambda}_0 = \Big(\lambda_{01}, \cdots, \lambda_{0q}\Big)^T,$$

$$\boldsymbol{\alpha}_k = \Big(\alpha_{k1}, \cdots, \alpha_{kh}\Big)^T,$$

$$\mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Gamma}_{11} & \cdots & \mathbf{\Gamma}_{1h} \\ \vdots & \ddots & \vdots \\ \mathbf{\Gamma}_{p1} & \cdots & \mathbf{\Gamma}_{qh} \end{pmatrix},$$

With this formulation, the functional clustering model can be written as :

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\lambda}_0 + \mathbf{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \ , \qquad i = 1, \cdots, N \ , \tag{6}$$

$$\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \mathbf{R}) \ , \qquad \boldsymbol{\gamma}_i, \sim N(\mathbf{0}, \mathbf{\Gamma}) \ ,$$

where $\mathbf{S}_i = \Big[\mathbf{s}(t_{i1}), \cdots, \mathbf{s}(t_{in_i})\Big]^T$ is the splines basis matrix for the individual i.

$$\mathbf{S}_i = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix},$$

$$\boldsymbol{\alpha}_{\mathbf{z}_i} = \left\{ (\alpha_{(1)})^{z_{i1}} \cdots (\alpha_{(k)})^{z_{ik}} \cdots (\alpha_{(G)})^{z_{iG}} \right\},$$

but $\boldsymbol{\lambda}_0$, $\boldsymbol{\alpha}_k$, and $\mathbf{\Lambda}$ could be confounded if no constraints were imposed : Hastie et al. (2001), Therefore we require that :

$$\sum_k \boldsymbol{\alpha}_k = \mathbf{0} \ , \tag{7}$$

that means that $\mathbf{s}(t)^T\boldsymbol{\lambda_0}$ may be interpreted as the overall mean curve, and

$$\mathbf{\Lambda}^T\mathbf{S}^T\mathbf{\Sigma}^{-1}\mathbf{S}\mathbf{\Lambda} = \mathbf{I} \ , \tag{8}$$

with :

$$\mathbf{\Sigma} = \sigma^2\mathbf{I} + \mathbf{S}\mathbf{\Gamma}\mathbf{S}^T,$$

where $\mathbf{S}$ is the splines basis matrix on a fine grid of time points over the full range of the data, and we will put $\mathbf{R} = \sigma^2\mathbf{I}$, and with $\mathbf{\Gamma}$ the same for every cluster.

Then we have :

- $\mathbf{s}(t)^T\boldsymbol{\lambda}_0$ the representation of the global mean curve,
- $\mathbf{s}(t)^T(\boldsymbol{\lambda}_0 + \mathbf{\Lambda}\boldsymbol{\alpha}_k)$ the global representation of the centroid of cluster k,
- $\mathbf{s}(t)^T\mathbf{\Lambda}\boldsymbol{\alpha}_k$ the local representation of the centroid of cluster k in connection with the global mean curve,
- $\mathbf{s}(t)^T\boldsymbol{\gamma}_i$ the local representation of the curve i in connection with the centroid of its cluster k.

2.3.1 Example : Equations for curve i

Let the curve i with n_i observations at times $\{t_{i1}, t_{i2}, \cdots, t_{in_i}\}$, with $g_i(t)$ the unknown true value, $y_i(t)$ the measurement, and $\epsilon_i(t)$ the measurement error, at time t, and G the number of clusters. We have :

$$\begin{pmatrix} y_i(t_1) \\ \vdots \\ y_i(t_{n_i}) \end{pmatrix} = \begin{pmatrix} g_i(t_1) \\ \vdots \\ g_i(t_{n_i}) \end{pmatrix} + \begin{pmatrix} \epsilon_i(t_1) \\ \vdots \\ \epsilon_i(t_{n_i}) \end{pmatrix} . \tag{9}$$

We represent the value $g_i(t)$ at time t on a spline basis vector $\mathbf{s}(t)$, with q dimensions, and $\boldsymbol{\eta}_i$ the random spline coefficients vector. Then we have :

$$g_i(t) = \begin{pmatrix} s_1(t) & \cdots & s_q(t) \end{pmatrix} \begin{pmatrix} \eta_{i1} \\ \vdots \\ \eta_{iq} \end{pmatrix} , \tag{10}$$

and an exhaustive description :

$$\begin{pmatrix} g_i(t_1) \\ \vdots \\ g_i(t_{n_i}) \end{pmatrix} = \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix} \begin{pmatrix} \eta_{i1} \\ \vdots \\ \eta_{iq} \end{pmatrix} . \tag{11}$$

We split the random spline coefficients vector $\boldsymbol{\eta_i}$ into a deterministic coefficients vector $\boldsymbol{\mu}_{\mathbf{z}_i}$ if the curve i is part of the cluster k defined by the cluster membership $z_{ik} = 1$ and $z_{il} = 0$ for $l \neq k$, $l = 1 \cdots G$, and a random coefficients vector $\boldsymbol{\gamma_i}$:

$$\begin{pmatrix} \eta_{i1} \\ \vdots \\ \eta_{iq} \end{pmatrix} = \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_q \end{pmatrix}_{\mathbf{z}_i} + \begin{pmatrix} \gamma_{i1} \\ \vdots \\ \gamma_{iq} \end{pmatrix} . \tag{12}$$

We also split the deterministic spline coefficients vector $\boldsymbol{\mu_i}$ into a deterministic coefficients vector $\boldsymbol{\lambda_0}$ which represents the coefficients of the global mean curve of all curves, and $\boldsymbol{\Lambda\alpha}_k$ which represents the centroid coefficients of the cluster k from the coefficients of the global mean curve of the population :

$$\begin{pmatrix} \mu_{k1} \\ \vdots \\ \mu_{kq} \end{pmatrix} = \begin{pmatrix} \lambda_{01} \\ \vdots \\ \lambda_{0q} \end{pmatrix} + \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1h} \\ \vdots & \ddots & \vdots \\ \lambda_{q1} & \cdots & \lambda_{qh} \end{pmatrix} \begin{pmatrix} \alpha_{k1} \\ \vdots \\ \alpha_{kh} \end{pmatrix} , \tag{13}$$

With all these representations, we have got :

$$\begin{aligned} \begin{pmatrix} y_i(t_1) \\ \vdots \\ y_i(t_{n_i}) \end{pmatrix} = & \begin{pmatrix} s_1(t_1) & \cdots & s_q(t_1) \\ \vdots & \ddots & \vdots \\ s_1(t_{n_i}) & \cdots & s_q(t_{n_i}) \end{pmatrix} \\ & \left\{ \begin{pmatrix} \lambda_{01} \\ \vdots \\ \lambda_{0q} \end{pmatrix} + \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1h} \\ \vdots & \ddots & \vdots \\ \lambda_{q1} & \cdots & \lambda_{qh} \end{pmatrix} \begin{pmatrix} \alpha_{k1} \\ \vdots \\ \alpha_{kh} \end{pmatrix} + \begin{pmatrix} \gamma_{i1} \\ \vdots \\ \gamma_{iq} \end{pmatrix} \right\} + \begin{pmatrix} \epsilon_i(t_1) \\ \vdots \\ \epsilon_i(t_{n_i}) \end{pmatrix} \end{aligned} \tag{14}$$

2.4 Parametric Identification

Now, we have to estimate the parameters $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Gamma}$, σ^2 et π_k by maximization of a likelihood function.
For $\mathbf{y}_i$ we have a conditional distribution :

$$\mathbf{y}_i \sim N\Big(\mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i}), \boldsymbol{\Sigma}_i\Big) ,$$

where

$$\boldsymbol{\Sigma}_i = \sigma^2\mathbf{I} + \mathbf{S}_i\boldsymbol{\Gamma}\mathbf{S}_i^T .$$

Since the observations of the different curves are independent, the joint distribution of $\mathbf{y}$, and $\mathbf{z}$ is given by :

$$f(\mathbf{y},\mathbf{z}) \;=\; \sum_{k=1}^{G} \pi_k \frac{1}{(2\pi)^{\frac{n}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\Big[-\frac{1}{2}\big(\mathbf{y}-\mathbf{S}(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}})\big)^T\boldsymbol{\Sigma}^{-1}\big(\mathbf{y}-\mathbf{S}(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}})\big)\Big] , \tag{15}$$

and the likelihood for the parameters $\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2$, given the observations $\mathbf{y}_i, \mathbf{z}_i$ is :

$$L(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_i, \mathbf{z}_i) = \prod_{i=1}^{N}\sum_{k=1}^{G} \pi_k \frac{1}{(2\pi)^{\frac{n_i}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}} \exp\Big[-\frac{1}{2}\big(\mathbf{y}_i-\mathbf{S}_i(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i})\big)^T\boldsymbol{\Sigma}_i^{-1}\big(\mathbf{y}_i-\mathbf{S}_i(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i})\big)\Big] . \tag{16}$$

Maximizing this likelihood would give us the parameters $\boldsymbol{\lambda}_0$, $\boldsymbol{\Lambda}$, $\boldsymbol{\alpha}_k$, π_k , $\boldsymbol{\Gamma}$ and σ^2 but unfortunately, a direct maximization of this likelihood is a difficult non-convex optimization problem. If the $\boldsymbol{\gamma}_i$ had been observed, then the joint likelihood of $\mathbf{y}_i$, $\mathbf{z}_i$ and $\boldsymbol{\gamma}_i$ would simplify, and like $\mathbf{z}_i$ and $\boldsymbol{\gamma}_i$ are independent, the joint distribution can be written as :

$$f(\mathbf{y},\mathbf{z},\boldsymbol{\gamma}) = f(\mathbf{y}|\mathbf{z},\boldsymbol{\gamma})f(\mathbf{z})f(\boldsymbol{\gamma}) ,$$

where $\mathbf{z}_i$ are multinomial (π_k), $\boldsymbol{\gamma}_i$ are $N(0,\boldsymbol{\Gamma})$, and $\mathbf{y}_i$ are conditional $N\Big(\mathbf{S}_i(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_k+\gamma_i);\sigma^2\mathbf{I}\Big)$.
The joint distribution is now written as :

$$f(\mathbf{y},\mathbf{z},\boldsymbol{\gamma}) = \frac{1}{(2\pi)^{\frac{n+q}{2}}|\boldsymbol{\Gamma}|^{\frac{1}{2}}} \exp\big(-\frac{1}{2}\boldsymbol{\gamma}^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}\big) \prod_{k=1}^{G}\Big\{\pi_k \exp\{-\frac{1}{2}n\log(\sigma^2)\} \exp\Big[-\frac{1}{2\sigma^2}\big(\mathbf{y}-\mathbf{S}(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_k+\boldsymbol{\gamma})\big)^T\big(\mathbf{y}-\mathbf{S}(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_k+\boldsymbol{\gamma})\big)\Big]\Big\}^{z_k}, \tag{17}$$

and the likelihood of the parameters is given by :

$$L(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_i, \mathbf{z}_i, \boldsymbol{\gamma}_i) = \prod_{i=1}^{N} \frac{1}{(2\pi)^{\frac{n_i+q}{2}}|\boldsymbol{\Gamma}|^{\frac{1}{2}}} \exp\big(-\frac{1}{2}\boldsymbol{\gamma}_i^T\boldsymbol{\Gamma}^{-1}\boldsymbol{\gamma}_i\big) \prod_{k=1}^{G}\Big\{\pi_k \exp\{-\frac{1}{2}n_i\log(\sigma^2)\} \exp\Big[-\frac{1}{2\sigma^2}\big(\mathbf{y}_i-\mathbf{S}_i(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_k+\boldsymbol{\gamma}_i)\big)^T\big(\mathbf{y}_i-\mathbf{S}_i(\boldsymbol{\lambda}_0+\boldsymbol{\Lambda}\boldsymbol{\alpha}_k+\boldsymbol{\gamma}_i)\big)\Big]\Big\}^{z_{ik}}. \tag{18}$$

In the paper, we will use the log likelihood :

$$
\begin{aligned}
l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_i, \mathbf{z}_i, \boldsymbol{\gamma}_i) = \\
& -\frac{1}{2} \sum_{i=1}^{N} (n_i + q) \log(2\pi) \\
& + \sum_{i=1}^{N} \sum_{k=1}^{G} z_{ik} \log(\pi_k) \qquad (19) \\
& -\frac{1}{2} \sum_{i=1}^{N} \left[\log(|\boldsymbol{\Gamma}|) + \boldsymbol{\gamma}_i^T \boldsymbol{\Gamma}^{-1} \boldsymbol{\gamma}_i \right] \qquad (20) \\
& -\frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{G} z_{ik} \Big[n_i \log(\sigma^2) + \frac{1}{\sigma^2} \big\| \mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i) \big\|^2 \Big] . \qquad (21)
\end{aligned}
$$

2.5 EM algorithm

The EM algorithm consists of iteratively maximizing the expected values of (19), (20) and (21) given $\mathbf{y}_i$ and the current parameters estimates. As these three parts involve separate parameters, we can optimize them separately :

2.5.1 E step

The E step is realized from :

$$\widehat{\boldsymbol{\gamma}}_i = E\Big\{ \boldsymbol{\gamma}_i | \mathbf{y}_i, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}, \boldsymbol{\Gamma}, \sigma^2, z_{ik} \Big\} .$$

For the curve i we have the model :

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i .$$

Let :

$$\mathbf{u}_i = \mathbf{y}_i - \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_k) ,$$

then, the joint distribution of $\mathbf{u}_i$ and of $\boldsymbol{\gamma_i}$ is written as :

$$\begin{pmatrix} \mathbf{u}_i \\ \boldsymbol{\gamma}_i \end{pmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{S}\boldsymbol{\Gamma}\mathbf{S}^T + \sigma^2\mathbf{I} & \mathbf{S}\boldsymbol{\Gamma} \\ \Gamma\mathbf{S}^T & \boldsymbol{\Gamma} \end{bmatrix} \right),$$

and the conditional distribution of $\boldsymbol{\gamma_i}$ given $\mathbf{u}_i$ is :

$$\boldsymbol{\gamma_i} | \mathbf{u}_i = N\Big(\widetilde{\boldsymbol{\gamma}}_{\boldsymbol{i}}; \boldsymbol{\Sigma}_{\gamma_i} \Big) ,$$

where :

$$\widetilde{\boldsymbol{\gamma}}_{\boldsymbol{i}} = \big(\mathbf{S}_i^T \mathbf{S}_i + \boldsymbol{\sigma}^2 \Gamma^{-1} \big)^{-1} \mathbf{S}_i^T \mathbf{u}_i ,$$

and :

$$\boldsymbol{\Sigma}_{\boldsymbol{\gamma_i}} = \sigma^2 \big(\mathbf{S}_i^T \mathbf{S}_i + \sigma^2 \boldsymbol{\Gamma}^{-1} \big)^{-1} .$$

Then, we have got the conditional distribution for $(\widehat{\boldsymbol{\gamma}}_i | \mathbf{y}_i, z_{ik} = 1)$:

$$(\widehat{\boldsymbol{\gamma}}_i | \mathbf{y}_i, z_{ik} = 1) \sim N\big(\widetilde{\boldsymbol{\gamma}}_{\boldsymbol{i}}; \boldsymbol{\Sigma}_{\widehat{\boldsymbol{\gamma}}_{\boldsymbol{i}}} \big) , \qquad (22)$$

with :

$$
\begin{aligned}
\widetilde{\boldsymbol{\gamma}}_{\boldsymbol{i}} &= \big(\mathbf{S}_i^T \mathbf{S}_i + \sigma^2 \boldsymbol{\Gamma}^{-1} \big)^{-1} \mathbf{S}_i^T \big(\mathbf{y}_i - \mathbf{S}_i \boldsymbol{\lambda}_0 - \mathbf{S}_i \boldsymbol{\Lambda} \boldsymbol{\alpha}_k \big) , \qquad (23) \\
\boldsymbol{\Sigma}_{\widehat{\boldsymbol{\gamma}}_{\boldsymbol{i}}} &= \sigma^2 \big(\mathbf{S}_i^T \mathbf{S}_i + \sigma^2 \boldsymbol{\Gamma}^{-1} \big)^{-1} . \qquad (24)
\end{aligned}
$$

2.5.2 M step

The M step involve maximizing :

$$Q = E\Big\{ l(\pi_k, \boldsymbol{\lambda}_0, \boldsymbol{\Lambda}, \boldsymbol{\alpha}_k, \boldsymbol{\Gamma}, \sigma^2 | \mathbf{y}_i, \mathbf{z}_i, \boldsymbol{\gamma}_i) \Big\} \ ,$$

holding $\boldsymbol{\gamma}_i$ fixed

2.5.3 Estimation of $\widehat{\pi}_k$

The expected value of (19) is maximized by setting :

$$\widehat{\pi_k} = \frac{1}{N} \sum_{i=1}^{N} \pi_{k|i} \ , \tag{25}$$

with :

$$\begin{aligned} \pi_{k|i} &= P\big(z_{ik} = 1 | \mathbf{y}_i\big) \ , \\ &= \frac{f(y|z_{ik}=1)\pi_k}{\sum_{j=1}^{G} f(y|z_{ij}=1)\pi_j} \ , \end{aligned}$$

with : $f(y|z_{ik} = 1)$ given by :

$$\mathbf{y}_i \sim N(\mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i}), \boldsymbol{\Sigma}_i) \ ,$$

where

$$\boldsymbol{\Sigma}_i = \sigma^2 \mathbf{I} + \mathbf{S}_i \boldsymbol{\Gamma} \mathbf{S}_i^T \ .$$

2.5.4 Estimation of $\widehat{\boldsymbol{\Gamma}}$

The expected value of (20) is maximized by setting :

$$\begin{aligned} \widehat{\boldsymbol{\Gamma}} &= \frac{1}{N} \sum_{i=1}^{N} E\Big[\widehat{\boldsymbol{\gamma}}_i \widehat{\boldsymbol{\gamma}}_i^T | \mathbf{Y}_i \Big] \ , \\ &= \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} E\Big[\widehat{\boldsymbol{\gamma}}_i \widehat{\boldsymbol{\gamma}}_i^T | \mathbf{y}_i, z_{ik} = 1 \Big] \ , \end{aligned} \tag{26}$$

with $(\widehat{\boldsymbol{\gamma}}_i | \mathbf{y}_i, z_{ik} = 1)$ given by the E step.

2.5.5 Estimation of $\boldsymbol{\lambda}_0$, $\boldsymbol{\alpha}_k$, $\boldsymbol{\Lambda}$

To maximize (21), we need an iterative procedure where $\boldsymbol{\lambda}_0$, $\boldsymbol{\alpha}_k$, and the columns of $\boldsymbol{\Lambda}$ are repeatedly optimized while holding all other parameters fixed.

2.5.6 Estimation of $\boldsymbol{\lambda}_0$

From the functional model :

$$\mathbf{y}_i = \mathbf{S}_i(\boldsymbol{\lambda}_0 + \boldsymbol{\Lambda}\boldsymbol{\alpha}_{\mathbf{z}_i} + \boldsymbol{\gamma}_i) + \boldsymbol{\epsilon}_i \ , \qquad i = 1, \cdots, N \ ,$$

we have got, by Generalized Least Squares (GLS) :

$$\widehat{\boldsymbol{\lambda}}_0 \ = \ \Big(\sum_{i=1}^{N} \mathbf{S}_i^T \mathbf{S}_i\Big)^{-1} \sum_{i=1}^{N} \mathbf{S}_i^T \Big[\mathbf{y}_i - \sum_{k=1}^{G} \pi_{k|i} \mathbf{S}_i(\boldsymbol{\Lambda}\boldsymbol{\alpha}_k + \widehat{\boldsymbol{\gamma}}_{ik})\Big] \ , \tag{27}$$

with $\widehat{\boldsymbol{\gamma}}_{ik} = E\Big\{\boldsymbol{\gamma}_{ik} | z_{ik} = 1, \mathbf{y}_i\Big\}$ given by the E step.

2.5.7 Estimation of $\boldsymbol{\alpha}_k$

The $\widehat{\boldsymbol{\alpha}}_k$ are estimated from :

$$\widehat{\boldsymbol{\alpha}}_k = \Big(\sum_{i=1}^{N} \pi_{k|i} \boldsymbol{\Lambda}^T \mathbf{S}_i^T \mathbf{S}_i \boldsymbol{\Lambda}\Big)^{-1} \sum_{i=1}^{N} \pi_{k|i} \boldsymbol{\Lambda}^T \mathbf{S}_i^T \Big[\mathbf{y}_i - \mathbf{S}_i \widehat{\boldsymbol{\lambda}_0} - \mathbf{S}_i \widehat{\boldsymbol{\gamma}}_{ik}\Big] \ . \tag{28}$$

2.5.8 Estimation of $\boldsymbol{\Lambda}$

By GLS, we only have the possibility of estimating vectors and no matrix, thus we will have to optimize each column of $\boldsymbol{\Lambda}$ separately, holding all other fixed using :

$$\begin{aligned} \boldsymbol{\Lambda}_m \ = \ & \Big(\sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{\alpha}_{km}^2 \mathbf{S}_i^T \mathbf{S}_i\Big)^{-1} \\ & \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_{k|i} \widehat{\alpha}_{km} \mathbf{S}_i^T \Big(\overline{\mathbf{y}}_i - \sum_{l \neq m}^{G} \widehat{\alpha}_{km} \mathbf{S}_i \widehat{\boldsymbol{\Lambda}}_l - \mathbf{S}_i \widehat{\boldsymbol{\gamma}}_{ik}\Big) \ , \end{aligned} \tag{29}$$

where :

- $\boldsymbol{\Lambda}_m$ is the column m of $\boldsymbol{\Lambda}$
- $\widehat{\alpha}_{km}$ is the component m of $\widehat{\boldsymbol{\alpha}}_k$
- $\overline{\mathbf{y}}_i = \mathbf{y}_i - \mathbf{S}_i \widehat{\boldsymbol{\lambda}}_0$

We iterate through (27) (28) (29) until all parameters have converged , then we can optimize σ^2 .

2.5.9 Estimation of σ^2

We have got :

$$\widehat{\sigma^2} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_k E\Big[\big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda}\boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i\big)^T \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda}\boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i\big) | \mathbf{y}_i, z_{ik} = 1\Big] \ . \tag{30}$$

Let :

$$\begin{aligned} \widehat{\sigma^2} \ = \ & \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{G} \pi_k \Big\{ \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda}\boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i\big)^T \big(\overline{\mathbf{y}}_i - \mathbf{S}_i \boldsymbol{\Lambda}\boldsymbol{\alpha}_k - \mathbf{S}_i \boldsymbol{\gamma}_i\big) \\ & + \mathbf{S}_i \ \mathrm{Cov}[\boldsymbol{\gamma}_i | \mathbf{y}_i, z_{ik} = 1] \mathbf{S}_i^T \Big\} \ . \end{aligned} \tag{31}$$

The algorithm iterates until all the parameters have converged.

3 Radial Basis Function Networks

Radial Basis Function Networks (RBFN) are neural networks used in approximation and classification tasks. They share with Multi-Layer Perceptrons the universal approximation property (Haykin, 1999). Classical RBF networks have their inputs fully connected to non-linear units in a single hidden layer. The output of a RBFN is a linear combination of the hidden units outputs. More precisely, the output is a weighted sum of Gaussian functions or kernels (i.e. the nonlinearities) applied to the inputs :

$$y \;=\; \sum_{i=1}^{I} \lambda_i \exp\Big\{ - \frac{|| \; x - c_i \; ||^2}{\sigma_i} \Big\} \,, \tag{32}$$

where x is the input vector, y is the scalar output of the RBFN, $c_i,\ 1 \leq i \leq I$, are the centers of the I Gaussian kernels, $\sigma_i,\ 1 \leq i \leq I$, are their widths, and $\lambda_i,\ 1 \leq i \leq I$, their weights. Intuitively those last λ_i parameters represent the relative importance of each kernel in the output y. As shown in equation (32), the RBF network has three sets of parameters $c_i,\ \sigma_i, \lambda_i, 1 \leq i \leq I$. One advantage of RBFN networks compared to other approximation models is that these three sets can be learned separately with suitable performances. Moreover the learning of the λ_i weights results from a linear system. A description of learning algorithms for RBF networks can be found in (Benoudjit and Verleysen, 2003).

4 The Forecasting Method

In this section we present a detailed model-based approach for clustering functional data and a time series forecasting method. This method will first be sketched to give an intuition of how the forecasting is performed. Then each step of the method will be detailed.

4.1 Method Description

The forecasting method is based on the "looking in the past" principle.

Let's the observations on the time interval $[t_0, T]$. To perform a functional prediction of the curve for the time interval $[t, t + \triangle t_{out}]$, we create two functional spaces.
A first functional space IN is built with past observations for the time interval $[t - \triangle t_{in}, t]$, the "regressors" and a similar second functional space OUT is built with observations for the time interval $[t - \triangle t_{in}, t + \triangle t_{out}]$. These two spaces are built with all data corresponding to times $t \in [t_0 + \triangle t_{in}, T - \triangle t_{in} - \triangle t_{out}]$.

These functional spaces are combined into a probabilistic way to build the functional prediction for the time interval $[t, t + \triangle t_{out}]$ and are quantized using the functional clustering algorithm. The relationship between the first and the second functional spaces issued from the clustering algorithms is encoded into a probability transition table constructed empirically on the datasets.

In each of the clusters determined by the second clustering OUT, a local RBFN model is built to approximate the relationship between the functional output (the local predic-

tion) and the functional input (the regressor).

Finally, the global functional prediction at time t for the interval $[t, t + \triangle t_{out}]$ is performed by combining the local models results associated to clusters OUT, according to their frequencies with respect to the class considered in the cluster IN.

4.2 Quantizing the « inputs »

Consider a scalar time series X, where $x(t)$ is the value at time t, $t \in [t_0, T]$. This original series is transformed into an array of observations X_{in} for the time intervals $[t, t + \triangle t_{in}]$, for all $t \in [t_0, t_0 + \triangle t_{in}, t_0 + 2\triangle t_{in}, \cdots, T - \triangle t_{in} - \triangle t_{out}]$.
Then the clustering algorithm is applied to the *input* array X_{in}; after convergence it gives an IN map of K_{in} codewords and the spline coefficients for the curves of each cluster in this IN map .

4.3 Quantizing the « outputs »

At each input vector of the matrix X_{in} we aggregate the next observations to get a new array Y_{out} for the time interval $[t, t + \triangle t_{in} + \triangle t_{out}]$ for all $t \in [t_0, t_0 + \triangle t_{in}, t_0 + 2\triangle t_{in}, \cdots, T - \triangle t_{in} - \triangle t_{out}]$.
The clustering algorithm is applied to the new array Y_{out}; after convergence it gives an OUT map of K_{out} codewords and the spline coefficients for the curves of each cluster in this OUT map.
Note that, by construction, there is a one-to-one relationship between each *input* and each *output* vector of spline coefficients.

4.4 Probability transition table

Both sets of codewords from maps IN and OUT only contain a static information. This information does not reflect completely the evolution of the time series.
The idea is thus to create a data structure that represents the dynamics of the time series, i.e. how each class of *output* vectors of spline coefficients (including the values for the time interval $[t, t + \triangle t_{out}]$) is associated to each class of *input* vectors of spline coefficients for the time interval $[t - \triangle t_{in}, t]$.
This structure is the probability transition table $T(i, j)$, with $1 \leq i \leq N_{in}$, $1 \leq j \leq N_{out}$.

Each element $T(i, j)$ of this table represents the proportion of *output* vectors that belongs to the j^{th} class of the OUT map while their corresponding *input* vectors belong to class i of the IN map. Those proportions are computed empirically for the given dataset and sum to one on each line of the table.
Intuitively the probability transition table represents all the possible evolutions at a given time t together with the probability that they effectively happen.

4.5 Local RBFN models

When applied to the « outputs », the functional clustering algorithm provides N_{out} classes and the spline coefficients of the curves for the intervals $[t, t + \triangle t_{out}]$. In each of these classes a RBFN model is learned.
Each RBFN model has p inputs (the spline's coefficients of the regressors) and q outputs

(the spline coefficients of the prediction curve).

These models represent the local evolution of the time series, restricted to a specific class of regressors. The local information provided by these models will be used when predicting the future evolution of the time series.

4.6 Forecasting

The relevant information has been extracted from the time series through both maps, the probability transition table and the local RBFN models detailed in the previous sections. Having this information, it is now possible to perform the forecasting itself.

At each time t, the goal is to estimate the functional curve for the time interval $[t, t + \triangle t_{out}]$ denoted $\hat{\mathbf{x}}([t, t + \triangle t_{out}])$.

First the *input* at time t is built, leading to $X(t)$. This vector is presented to the IN map, and the nearest codeword $X_{k(t)}$ is identified ($1 \leq k(t) \leq N_{in}$).

In the frequency table, in the $k(t)^{\text{th}}$ line, there are some columns corresponding to classes of the OUT map for which the proportions are non zero. This means that those columns represent possible evolutions for the considered data $X(t)$, since $X(t)$ has the same shape than data in the $k(t)^{\text{th}}$ class.

For each of those potential evolutions, the respective RBFN models are considered (one RBFN model has been built for each class in the OUT map).
For each of them, a local prediction $\hat{x}_j([t, t + \triangle t_{out}])$ is obtained ($1 \leq j \leq N_{out}$).

The final prediction is a weighted sum of the different local predictions, the weights being the proportions recorded in the probability transition table.

The final prediction is thus :

$$\hat{\mathbf{x}}([t, t + \triangle t_{out}]) = \sum_{j=1}^{N_{out}} T(k, j)\hat{\mathbf{x}}_j([t, t + \triangle t_{out}]). \tag{33}$$

5 Experimental Results

The examples presented here deal with the IBM stock time series of "tick data" for the period starting on January 02, 1997 and ending on may 08, 1997 with more than 3000 transactions per day, on the New York Stock Exchange (NYSE).

The pricing model will use as inputs, inhomogeneous and high-frequency time series of bid and ask prices and also implied volatility. Such volatility time series can be obtained from market data, derived from option market prices(from eight call and put near-the-money, nearby and secondary nearby option contracts on the underlying asset)or computed from diverse model assumptions.

On Fig. 1 we can see the evolution of the Prices (top) and Volumes (bottom) on one day.

On Fig. 2 we see the distribution of transactions for the same day. Each point is a transaction, with more transactions at the opening and closing of the NYSE.
The transactions are sampled discretely in time and like it is often the case with financial data the time separating successive observations is itself random.
On Fig. 3 we can see two successive days of the stock IBM with a fine smoothing of the "tick data" by splines.

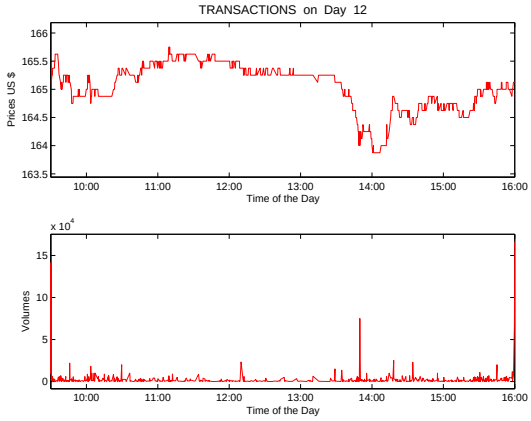

Figure 1: Prices (Top) et Volumes (Bottom)

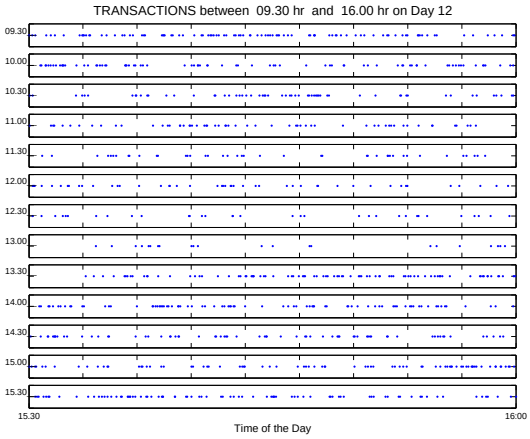

Figure 2: Distributions of transactions

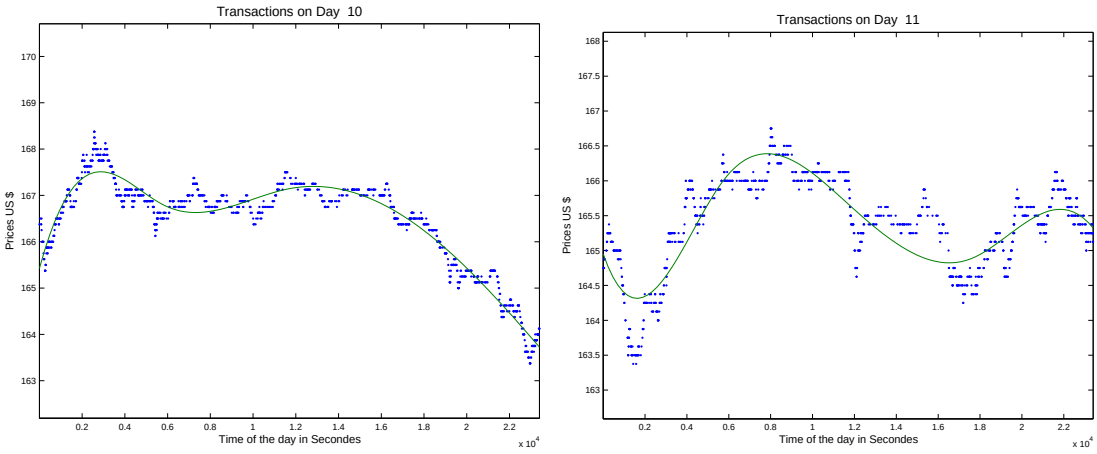

Figure 3: Two days of transactions for IBM (dashed curve),with smoothing splines (solid curve)

5.1 Prediction

We forecast the future transactions splines for three hours of day J between 10.30Hr. and 13.30 Hr. ($\triangle t_{out} = 3.0Hr$), from the past transactions (days $J-2$ and $J-1$ and half an hour of day J between 10.00 Hr. and 10.30 Hr.), in this case ($\triangle t_{in} = 11.30Hr$).
We have eliminated the transactions at the opening and closing of the NYSE, which are "outliers" without any correlation with the next hours.

On Fig. 4 we can see four out-of-sample forecasting days superposed with the observations and smoothing splines (not known by the model). There is a good correlation

between the out-of-sample forecasting and the observations.

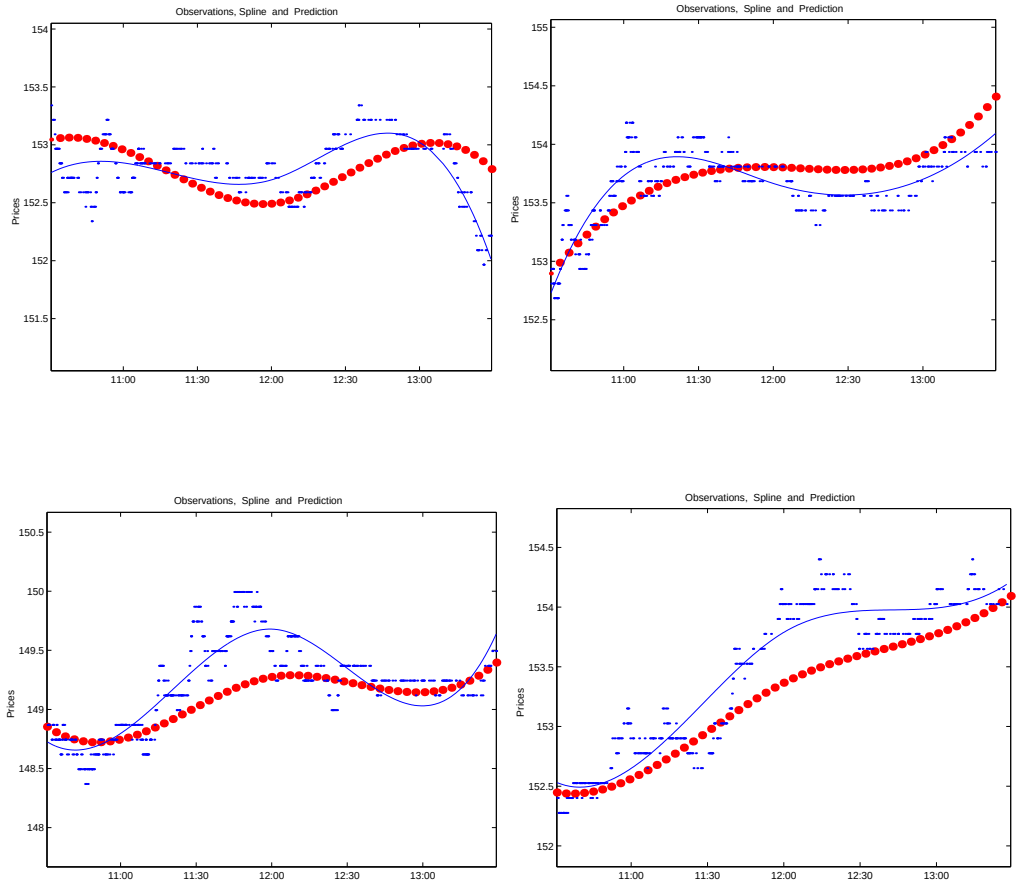

Figure 4: Four forecasting days for IBM stock price. Observations (Points); Smoothing splines (solid curve); Out-of-sample forecasting by the model (dashed curve)

6 Conclusion

We have presented a functional method for the clustering, modelling and forecasting of time series by functional analysis and neural networks. This method can be applied to any types of time series but is particularly effective when the observations are sparse, irregularly spaced, occur at different time points for each curve, or when only fragments of the curves are observed; standard methods completely fail in these circumstances.
By the functional clustering, we can also realize the forecasting of multiple dynamic processes.

References

Ait-Sahalia, Y. and P.A. Myland,(March, 2003),"The effects of random and discrete sampling when estimating continuous-time diffusions",*Econometrica*, Vol. 71, pp. 483-549.

Benoudjit, N. and 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.

Box, G. and G.M. Jenkins,(1970), " Time series Forecasting and Control ", *Holden-Day* ,

de Boor, C. (1978), " A Practical Guide to Splines ", *New York : Springer* ,

Dempster,A.P., N.M. Laird, and D.B. Rubin,(1977), "Maximum likelihood from incomplete data via the EM algorithm ", *Jounal of the Royal Statistical Society, Ser. B* , Vol. 39, pp. 1-22.

Diebold, F., J.H. Lee and G.Weinbach, "Regime switching with time-varying transition probabilities, in non stationary time series analysis and cointegration", ed. by C. Hargreaves, 1994, pp 283-302, *Oxford University Press.*

Engel, R.F., "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation",*Econometrica*, Vol. 50, pp. 987-1007, 1982.

Fama, E.F., "Efficient Capital Markets : II",*The Journal of Finance*, Vol. XLVI, N° 5, pp. 1515-1617, 1991.

Hamilton, J.D.,(1994), " Time Series Analysis ", *Princeton University Press* ,

Hastie, T., R. Tibshirani, and J. Friedman ,(2001), ' " The Elements of Statistical Learning. dta Mining, Inference, and prediction ", *Springer* ,

Haykin, S.,(1999), " Neural Networks - A comprehensive foundation ", *Prentice Hall*,

Franke, G., R. Olsen and W. Pohlmeier,(June 2002), "Overview of Forecasting Models",*University of Konstanz*, Seminar on High Frequency Finance, paper SS 2002.

James, G.M., T.H. Hastie and C.A. Sugar,(2000), "Principal component models for sparse functional data", *Biometrika*, Vol. 87, pp. 587-602.

James, G.M., C.A. Sugar,(2003), "Clustering for sparcely sampled functional data", *Journal of the Americal Association*, Vol. 98, pp. 397-408.

Ramsay, J.O., and B.W. Silverman,(1997), " Functional Data Analysis ", *Springer Series in Statistics*,

Rice, J.A. and C.O. Wu,(March 2001), "Nonpoarametric Mixed Effects models for Unequally Sampled Noisy Curves",*Biometrics*, Vol. 57, pp. 253-259.

Olsen, R.B. and M.M. Dacorogna,(December 1992), "going Back to the Basics - Rethinking Market Efficiency",*Olsen and Associates research Group*, Paper RBO.1992-09-07.