

Nonparametric estimation in Markov switching autoregressive-ARCH models

Joseph Tadjuidje Kamgaing · Jean-Pierre Stockis ·
Jürgen Franke

Received: date / Accepted: date

Abstract We consider a time series switching between different states which all are characterized as nonparametric AR-ARCH models. The switching is controlled by a hidden Markov chain with finitely many states. We approximate the autoregressive and volatility functions by neural networks and provide an EM algorithm to calculate quasi maximum likelihood estimates of the parameters. A Viterbi algorithm allows to reconstruct the hidden state sequence from the observations. We illustrate the applicability of this approach with a simple portfolio management problem. Finally, we show a consistency result for the network parameters.

Keywords nonparametric estimation · neural networks · autoregression · ARCH · mixture · Markov switching · hidden variables · EM algorithm, · Viterbi algorithm

1 Introduction

Time series data often show locally a time-homogeneous pattern, but are not stationary on the whole. Frequently, changes in the structure of the data generating process are more or less sudden, and between those changepoints, the time series follows a stationary regime. Two examples among many are EEG records from sleeping persons switching between different sleep states, see Müller et al.(1995) or financial time series showing different local trends and volatilities depending on the state of the market. The latter is our main example and will be discussed in detail in section 3.2.

We model such data by a stochastic process which is driven by K different dynamics where one and only one is active at each instant. Not all of them have to be necessarily stationary; explosive states are admissible if they do not occur too frequently. The points where a change of the dynamics takes place are the changepoints of the model. Switching

J. Tadjuidje Kamgaing
Department of Mathematics, University of Kaiserslautern
Erwin-Schrödinger-Str., 67663 Kaiserslautern
Tel.: +49-631-2054707
Fax: +49-631-2052748
E-mail: tadjuidj@mathematik.uni-kl.de
J. P. Stockis and J. Franke
University of Kaiserslautern, Department of Mathematics

between the states is controlled by a hidden Markov chain Q_t with values in $\{1, \dots, K\}$. In this paper, we assume that the time series data between changepoints are generated by nonparametric autoregressive-ARCH models, i.e. the whole process is a mixture of K such models

$$X_t = \sum_{k=1}^K S_{tk} \left(m_k(X_{t-1}, \dots, X_{t-M}) + \sigma_k(X_{t-1}, \dots, X_{t-M}) \varepsilon_t \right) \quad (1)$$

with

$$S_{tk} = \begin{cases} 1 & \text{for } Q_t = k \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where we consider the same order M of all the autoregressive and volatility functions m_k, σ_k involved for sake of convenience only. We call such a switching regime a CHARME (conditional heteroskedastic autoregressive mixture of experts) model. Depending on the values of $\{Q_t\}$, the process $\{X_t\}$ passes through different dynamics characterized by the autoregressive functions m_k and the volatility functions $\sigma_k > 0, k = 1, \dots, K$. The random errors ε_t are assumed to be independent and identically distributed (i.i.d.) with mean 0 and variance 1. For sake of simplicity, we only consider the case where ε_t has a density $p_\varepsilon(u) > 0$ for all $u \in \mathbb{R}$.

The distribution of the hidden state process is given by the $K \times K$ transition probability matrix A , i.e.

$$A_{jk} = \mathbb{P}(Q_t = k \mid Q_{t-1} = j),$$

and we denote the weights of the corresponding stationary distribution by $\pi = (\pi_1, \dots, \pi_K)$, i.e. in the stationary state we have $\pi_k = \text{pr}(Q_t = k)$. Mark that the latter are determined by A via $\pi A = \pi$.

The use of switching processes like (1) as models for economic time series goes back to Hamilton (1989), who applied a parametric switching autoregressive model, where the m_k are linear and the σ_k constant, to the long-term US real GNP. Rydén et al. (1998) have shown that even rather simple models of that form can reproduce the stylized facts of stock price return series. The addition of time varying volatilities of parametric ARCH-form has been considered by Hamilton and Susmel (1994) for modeling stock price data. The same kind of models has been investigated by Wong and Li (2000, 2001) with a focus on algorithms for computing the estimates. Francq et al. (1997, 1998, 2001) have developed stability properties as well as asymptotic theory for parameter estimators of such parametric switching autoregressions with or without heteroskedasticity.

To allow for more flexibility, we consider a nonparametric approach in this paper, allowing the autoregressive and volatility functions $m_k, \sigma_k, k = 1, \dots, K$, to be of arbitrary form. Such nonparametric autoregressive-ARCH models without switching, i.e. $K = 1$ are now well established, compare e.g. Härdle and Tsybakov (1997), Hafner (1998) or Franke et al. (2004), where estimates of m_1 and σ_1 based on local smoothing have been thoroughly investigated. If the order M of the AR- and ARCH-models involved is larger than say 1 or 2, local smoothers suffer from the scarcity of data in local neighborhoods. In that case, sieve estimates, compare Grenander (1981), are more convenient. A particular class of such estimates is based on fitting the output functions of feedforward neural networks to the data and has been applied in particular to financial data. We consider this type of nonparametric procedure in this paper to estimate the functions m_k, σ_k in the switching model (1).

In the next section, we derive estimates of the model parameters, using a conditional quasi-maximum-likelihood approach, and describe a numerical algorithm for calculating

those estimates. Additionally, we provide an algorithm which allows to reconstruct the hidden states Q_t from the data. In section 3, we apply the model and the estimation algorithms to some simulated data and to a simple portfolio management problem. Finally, we show a consistency result for the parameters of the autoregressive and volatility functions of the CHARME model in section 4 which follows from a theorem of Pötscher and Prucha (1997).

2 The estimation algorithm

For estimating the unknown functions m_k, σ_k in (1) nonparametrically, we approximate them by neural networks. In the presence of noise, such estimates have turned out to possess nice asymptotic properties under rather general assumptions (compare, e.g. White (1984) or Franke and Diagne (2001) and some of the references therein), and they are easy to implement. In fitting the mixture model (1) to data, we therefore approximate $m_k(x)$ and $\sigma_k(x)$ by output functions of feedforward neural networks with one hidden layer

$$f_k(x) = v_{0k} + \sum_{h=1}^H v_{hk} \psi(\langle \alpha_{hk}, x \rangle + b_{hk}), k = 1, \dots, K, \quad (3)$$

and

$$f_k^*(x) = v_{0k}^* + \sum_{h=1}^H v_{hk}^* \psi(\langle \alpha_{hk}^*, x \rangle + b_{hk}^*), k = 1, \dots, K, \quad (4)$$

where ψ is a sigmoid activation function, e.g. the logistic function. $\langle \alpha_{hk}, x \rangle$ denotes the scalar product of weight vector $\alpha_{hk} \in \mathbb{R}^M$ and input vector $x \in \mathbb{R}^M$. We denote the vector of the combined parameters, i.e. of $v_{0k}, \dots, v_{Hk}, \alpha_{hk}, b_{hk}, h = 1, \dots, H$, by θ_k for the weights of $f_k(x)$ and correspondingly by θ_k^* for the weights of $f_k^*(x)$. Also, we use $\theta = (\theta_1, \dots, \theta_K), \theta^* = (\theta_1^*, \dots, \theta_K^*)$ to denote the combined weights of all the autoregressive functions resp. volatility functions, and $\vartheta = (\theta, \theta^*)$ for all the network parameters. To keep notation simple, we restrict ourselves to the case where all the $2K$ networks involved have the same number H of neurons in the hidden layer. A generalization to individual network sizes is straightforward.

For the purpose of estimating m_k, σ_k , we replace model (1) by the misspecified parametric model

$$X_t = \sum_{k=1}^K S_{tk} \left(f_k(X_{t-1}, \dots, X_{t-M}) + f_k^*(X_{t-1}, \dots, X_{t-M}) \varepsilon_t \right), \quad (5)$$

which, however, approximates the true data-generating process (1) arbitrarily well by the universal approximation properties of neural networks provided m_k, σ_k satisfy some weak regularity condition.

The most common way to learn network weights is based on a nonlinear least-squares approach which corresponds to Gaussian maximum likelihood. This method frequently works well in practice, and even in case that the innovations ε_t are not normally distributed, consistency and asymptotic normality can be proven in an appropriate sense taking into account that the neural networks are usually misspecified. In case of a mixture model like (1), however, the reason for using nonlinear least-squares is unsustainable. We, therefore, translate the Gaussian maximum likelihood idea to the generalized type of process which switches between finitely many states. For the moment, we pretend that the innovations ε_t are standard normal random variables and that the chosen neural networks provide a correct

specification of the data generating process, i.e. $m_k \equiv f_k, \sigma_k \equiv f_k^*, k = 1, \dots, K$, for suitably chosen network weights.

To illustrate the difficulties, let us first consider the simple special case where the Q_t are i.i.d. with $\pi_k = \text{pr}(Q_t = k), k = 1, \dots, K$. With the state variables Q_t not observable, the conditional probability density of a single observation X_t at $z \in \mathbb{R}$ given $\mathbf{X}_{t-1} = (X_{t-1}, \dots, X_{t-M}) = \mathbf{x} \in \mathbb{R}^M$ is

$$g_{\vartheta, \pi}(z|\mathbf{x}) = \sum_{k=1}^K \pi_k \varphi(z; f_k(\mathbf{x}), f_k^*(\mathbf{x})),$$

where $\pi = (\pi_1, \dots, \pi_K)$ and $\varphi(z; \mu, \sigma)$ denotes the normal density with mean μ and standard deviation σ . As usual for stationary time series with an autoregressive dependence Brockwell and Davis (1991), we consider the conditional log likelihood given an initial piece X_0, \dots, X_{-M+1} as the target function. In the case of i.i.d. state variables Q_t , it is given by

$$\ell(\vartheta, \pi|X^{(N)}) = \sum_{t=1}^N \log g_{\vartheta, \pi}(X_t|X_{t-1}, \dots, X_{t-M}),$$

where $X^{(N)} = (X_{-M+1}, \dots, X_N)$. This is already of a rather complicated form and hard to maximize numerically. A popular statistically motivated algorithm which is able to handle such situations is the EM algorithm (compare, e.g., Dempster et al. (1977), Wu (1983)). It is based on the observation that the complete log likelihood which we would have if the hidden state variables Q_1, \dots, Q_N or, equivalently, $S^{(N)} = (S_1, \dots, S_N)$ would be known is simpler. Using that the S_{tk} assume the values 0 and 1 only, we get

$$\begin{aligned} \ell_c(\vartheta, \pi|X^{(N)}, S^{(N)}) &= \sum_{t=1}^N \sum_{k=1}^K S_{tk} \log \left(\pi_k \varphi(X_t; f_k(\mathbf{X}_{t-1}), f_k^*(\mathbf{X}_{t-1})) \right) \\ &= \ell_1(\pi|S^{(N)}) + \ell_2(\vartheta|X^{(N)}, S^{(N)}) \end{aligned}$$

separating into two components depending on the parameter π of the state variables and the parameter ϑ of the time series involved only.

$$\ell_1(\pi|S^{(N)}) = \sum_{t=1}^N \sum_{k=1}^K S_{tk} \log \pi_k$$

and up to a constant term

$$\ell_2(\vartheta|X^{(N)}, S^{(N)}) = - \sum_{t=1}^N \sum_{k=1}^K S_{tk} \left(\log f_k^*(\mathbf{X}_{t-1}) + \frac{1}{2} \left(\frac{X_t - f_k(\mathbf{X}_{t-1})}{f_k^*(\mathbf{X}_{t-1})} \right)^2 \right). \quad (6)$$

The EM algorithm iterates between approximating the hidden variables S_{tk} by their conditional expectations \hat{S}_{tk} given the observed data and using a preliminary estimate of the parameters on the one hand, and by maximizing $\ell_c(\vartheta, \pi|X^{(N)}, \hat{S}^{(N)})$ to get an update of estimates of ϑ, π on the other hand. For pure autoregressions, where the conditional variance of X_t given the past is constant, Franke et al. (2009) have shown convergence of this algorithm and consistency of the resulting estimates in a nonparametric setting using kernel estimates for estimating the autoregressive function.

In the general case, we assume that the hidden state variables forms a stationary Markov chain with finite state space $\mathcal{K} = \{1, \dots, K\}$. To get a concise representation of the conditional log-likelihood in that case and, later on, of the estimation and filtering algorithm, we first introduce some notation. First, we denote by

$$X^{(N)} = \{X_{-M+1}, \dots, X_N\}, Q^{(N)} = \{Q_0, \dots, Q_N\}, N \geq 1,$$

the observed sample up to time n resp. the corresponding hidden sample of the state process. We always condition on the initial piece $\{X_{-M+1}, \dots, X_0\}$ of the observed time series. We denote by

$$x^{(N)} = \{x_{-M+1}, \dots, x_N\}, q^{(N)} = \{q_0, \dots, q_N\}$$

possible values of $X^{(N)}, Q^{(N)}$. We assume that the observations X_t have a density w.r.t. Lebesgue measure λ whereas the hidden variables have a discrete distribution. We characterize the joint distribution of a part X of $X^{(n)}$ and a part Z of $Q^{(n)}$ by the density $p(x, z)$ of (X, Z) w.r.t. the product measure $\lambda \otimes \nu$ where ν denotes the counting measure of the set of possible values of Z , i.e.

$$\mathbb{P}(X \in B, Z \in A) = \sum_{z \in A} \int_B p(x, z) dx.$$

We now state our main assumption on the dependence structure of the observed and the hidden process.

A. 1 for any $N \geq 1, q_N \in \mathcal{H}, q^{(N-1)} \in \mathcal{H}^N, x^{(N-1)} \in \mathbb{R}^{N-1+M}$,

$$\mathbb{P}(Q_N = q_N | X^{(N-1)} = x^{(N-1)}, Q^{(N-1)} = q^{(N-1)}) = \mathbb{P}(Q_N = q_N | Q_{N-1} = q_{N-1}),$$

i.e. $\{Q_t\}$ forms a Markov chain, and its conditional distribution given the past depends only on its own past and not on the past observations $X_j, j < t$.

A. 2 for any $N \geq 1, x \in \mathbb{R}, q^{(N)} \in \mathcal{H}^{N+1}, x^{(N-1)} \in \mathbb{R}^{N-1+M}$ with $\mathbf{x} = (x_{N-1}, \dots, x_{N-M})$, the conditional density of X_N given $X^{(N-1)}, Q^{(N)}$ depends only on the current state Q_N and the last M observations $\mathbf{X}_{N-1} = (X_{N-1}, \dots, X_{N-M})$:

$$p(x | X^{(N-1)} = x^{(N-1)}, Q^{(N)} = q^{(N)}) = p(x | \mathbf{X}_{t-1} = \mathbf{x}, Q_N = q_N)$$

Given Q_0 the distribution of Q_1, Q_2, \dots is determined by the transition probability matrix A of the Markov chain. The latter also uniquely determines the probability weights $\pi_i = \mathbb{P}(Q_0 = i), i = 1, \dots, K$, of the corresponding stationary distribution. We model the data generating process of the observation X_t by (5) which is characterized the parameter vector ϑ . As in the independent switching situation above, we want to determine the conditional log-likelihood $\ell(\vartheta, A | X^{(N)})$ given X_{-M}, \dots, X_0 and, as an auxiliary quantity, the complete log likelihood $\ell_c(\vartheta, A | X^{(N)}, S^{(N)})$ or equivalently $\ell_c(\vartheta, A, | X^{(N)}, Q^{(N)})$. We have for the joint density of $X^{(N)}, Q^{(N)}$

$$p_{\vartheta, A}(x^{(N)}, q^{(N)}) = p_{\vartheta, A}(x^{(N)} | Q^{(N)} = q^{(N)}) p_A(q^{(N)})$$

where $p_A(q^{(N)}) = \mathbb{P}(Q^{(N)} = q^{(N)})$ does not depend on the parameter vector ϑ of the observed process by A.1. By the Markov property of Q_t , we have for the latter

$$p_A(q^{(N)}) = \left(\prod_{t=1}^N p_A(q_t | Q_{t-1} = q_{t-1}) \right) \mathbb{P}(Q_0 = q_0) = \pi_{q_0} \left(\prod_{t=1}^N A_{q_{t-1}, q_t} \right).$$

we also have

$$\begin{aligned} p_{\vartheta, A}(x^{(N)}, q^{(N)}) &= p_{\vartheta, A}(X_N | X^{(N-1)} = x^{(N-1)}, Q^{(N)} = q^{(N)}) \times \\ &\quad p_{\vartheta, A}(q_N | X^{(N-1)} = x^{(N-1)}, Q^{(N-1)} = q^{(N-1)}) p_{\vartheta, A}(x^{(N-1)}, q^{(N-1)}) \quad (7) \\ &= p_{\vartheta, A}(X_N | \mathbf{X}_{N-1} = \mathbf{x}, Q_N = q_N) p_A(q_N | Q_{N-1} = q_{N-1}) p_{\vartheta, A}(x^{(N-1)}, q^{(N-1)}) \end{aligned}$$

by A.1, A.2. Iterating (7) we immediately get for the complete likelihood conditional on $(X_{-M+1}, \dots, X_0) = \mathbf{X}_0$

$$L_c(\vartheta, A | X^{(N)}, \mathcal{Q}^{(N)}) = \pi_{\mathcal{Q}_0} \prod_{t=1}^N A_{\mathcal{Q}_{t-1}, \mathcal{Q}_t} p_{\vartheta, A}(X_t | \mathbf{X}_{t-1}, \mathcal{Q}_t)$$

and for the corresponding log-likelihood

$$\begin{aligned} \ell_c(\vartheta, A | X^{(N)}, \mathcal{Q}^{(N)}) &= \log \pi_{\mathcal{Q}_0} + \sum_{t=1}^N \log A_{\mathcal{Q}_{t-1}, \mathcal{Q}_t} + \sum_{t=1}^N \log p_{\vartheta, A}(X_t | \mathbf{X}_{t-1}, \mathcal{Q}_t) \\ &= \ell_1(A | \mathcal{Q}^{(N)}) + \ell_2(\vartheta | X^{(N)}, \mathcal{Q}^{(N)}) \end{aligned} \quad (8)$$

where ℓ_2 is the same as in (6) and

$$\ell_1(A | \mathcal{Q}^{(N)}) = \log \pi_{\mathcal{Q}_0} + \sum_{t=1}^N \log A_{\mathcal{Q}_{t-1}, \mathcal{Q}_t}.$$

However, the state variables \mathcal{Q}_t are hidden, such that we have to consider the incomplete likelihood. Due to the dependence of \mathcal{Q}_t we have to sum over all possible path of the Markov chain starting at time $t = 0$.

$$L(\vartheta, A | X^{(N)}) = \sum_{q_0, \dots, q_N=1}^K \pi_{q_0} \prod_{t=1}^N A_{q_{t-1}, q_t} p_{\vartheta, A}(X_t | \mathbf{X}_{t-1}, \mathcal{Q}_t = q_t).$$

With $s_t = (s_{t1}, \dots, s_{tK}), s_{tk} = 1_k(q_t)$, we again have

$$p_{\vartheta, A}(z | \mathbf{X}_{t-1} = \mathbf{X}, \mathcal{Q}_t = q_t) = \sum_{k=1}^K s_{tk} \varphi(z; f_k(\mathbf{X}), f_k^*(\mathbf{X}))$$

Our goal now is to maximize

$$L(\vartheta, A | X^{(N)}) \quad \text{or equivalently,} \quad \ell(\vartheta, A | X^{(N)}) = \log L(\vartheta, A | X^{(N)})$$

to get the quasi maximum likelihood estimates $\hat{\vartheta}, \hat{A}$ for the parameter of our model. Additionally, we want to solve the filtering problem, i.e. to get estimates $\hat{\mathcal{Q}}_t$ of the hidden state variables \mathcal{Q}_t , or equivalently, estimates \hat{S}_t of the state vectors $S_t, t = 1, \dots, N$, from the observation $X^{(N)}$.

An elegant and efficient way of solving the problem is to make use of the EM algorithm. In our case, roughly speaking, in the E-step we assume that preliminary estimates of the model parameters are known and used to compute estimates of the conditional expectations of S_{tk} given the observations. In the M-step we use the latter estimates obtained in the E-step to replace S_{tk} in the complete log likelihood which we then maximize to get estimates of the parameters ϑ, A . The E-step and M-step are iteratively repeated until some stopping criterion are satisfied. At the end of our numerical procedure, we do not only get the quasi-maximum-likelihood estimates of ϑ but also estimates of the conditional expectations of S_{tk} given the data, i.e. which may be considered as first estimates for the state variables of the system themselves. We improve those estimates of S_{tk} by using the Viterbi algorithm which computes the optimal(most likely) state sequence for a Markov switching model given the sequence of observed outputs.

2.1 The EM algorithm

Baum et al. (1970) have proposed an elegant procedure to compute the likelihood $\mathbb{P}_{\vartheta}(X^{(N)})$ for Markov processes, and Dempster et al. (1977) introduced the so-called *Expectation Maximization* or EM algorithm to maximize it. This last proposal can be regarded as an extension of the Forward-Backward procedure. This kind of numerical procedure to maximize the likelihood in the presence of hidden variables is well-established, in particular in the context of hidden Markov chains. An extended discussion can be found, e.g., in Cappé et al. (2005). We, therefore, start immediately with the form of the algorithm for our problem at hand.

2.1.1 Forward-Backward Procedure or E-Step

In this subsection, the probabilities and densities are calculated given the parameters ϑ, A which for the moment are assumed to be known and which in the iterative scheme will be replaced by estimates. To simplify reading, we do not explicitly mark this dependence on ϑ, A in the notation. As above, p denotes the density of observed variables w.r.t Lebesgue measure or the joint density of observed and hidden variables w.r.t to the Lebesgue measure and the counting measure. To stress which of the hidden variables is considered, we write, e.g., $p(x^{(N)}, Q_t = i)$ for the joint density of the whole observed sample $X^{(N)}$ and the single hidden variable Q_t evaluated at $(x^{(N)}, i) \in \mathbb{R}^{N+M} \times \mathcal{K}$. Recall that we assume the Markov chain to be stationary, and, therefore, $\mathbb{P}(Q_t = i) = \pi_i, i = 1, \dots, K, t \geq 0$.

Forward Procedure

Let α_j^t be the joint density of the observation from time $-M+1$ to t and of being in state j at time t , i.e.

$$\begin{aligned} \alpha_j^t &= p(X_{-M+1}, \dots, X_1, \dots, X_t, Q_t = j) \\ &= p(X_{-M+1}, \dots, X_1, \dots, X_t \mid Q_t = j) \pi_j, \quad 1 \leq t \leq N; \end{aligned} \quad (9)$$

where $p(x^{(t)} \mid Q_t = j)$ is the conditional density of $X^{(t)} = (X_{-M+1}, \dots, X_1, \dots, X_t)$ given $Q_t = j$. In particular, the density of the whole sequence of observations is given by the sum over all states at the end (N) of the sequence, i.e.

$$p(x^{(N)}) = \sum_{j=1}^K \alpha_j^N. \quad (10)$$

The surprising fact about this representation is its low computational complexity. Rather than being exponential in the sample size, it is only linear in N since $\alpha_i^N, i = 1, \dots, K$, can be computed recursively based on the assumptions A.1 and A.2 above.

$$\begin{aligned} \alpha_j^{t+1} &= p(X_{-M+1}, \dots, X_1, \dots, X_t, X_{t+1}, Q_{t+1} = j) \\ &= p(X_{t+1} \mid X^{(t)}, Q_{t+1} = j) \sum_{i=1}^K \mathbb{P}(Q_{t+1} = j \mid X^{(t)}, Q_t = i) p(X^{(t)}, Q_t = i) \\ &= p(X_{t+1} \mid \mathbf{X}_t, Q_{t+1} = j) \sum_{i=1}^K \mathbb{P}(Q_{t+1} = j \mid Q_t = i) p(X^{(t)}, Q_t = i) \\ &= b_j^{t+1} \left[\sum_{i=1}^K A_{ij} \alpha_i^t \right] \end{aligned} \quad (11)$$

with, using that we pretend the innovations ε_t to be standard normal random variables,

$$\begin{aligned} b_j^{t+1} &= p(X_{t+1} | \mathbf{X}_t, Q_{t+1} = j) = p(X_{t+1} | X_t, \dots, X_{t+1-M}, Q_{t+1} = j) \\ &= \varphi(X_{t+1}; f_j(\mathbf{X}_t), f_j^*(\mathbf{X}_t)). \end{aligned} \quad (12)$$

As we assume X_{-M+1}, \dots, X_0 to be given and Q_0 to follow the stationary distribution π of the Markov chain, this sequence can be initialized with

$$\alpha_j^1 = p(X_{-M+1}, \dots, X_0, X_1, Q_1 = j) = \pi_j b_j^1. \quad (13)$$

This step is called the forward procedure, given the initial values of π_j and b_j^1 .

Backward Procedure

In the same way as above we define β_i^t (the backward variable) as the conditional density of observing $X_s, s = t+1, \dots, N$, given the state i at time t and the past realizations of the process \mathbf{X}_t

$$\begin{aligned} \beta_i^t &= p(X_{t+1}, \dots, X_N | \mathbf{X}_t, Q_t = i) \\ &= \sum_{j=1}^K p(X_{t+1}, \dots, X_N, Q_{t+1} = j | \mathbf{X}_t, Q_t = i) \\ &= \sum_{j=1}^K p(X_{t+2}, \dots, X_N | \mathbf{X}_{t+1}, Q_{t+1} = j) p(X_{t+1} | \mathbf{X}_t, Q_{t+1} = j) \mathbb{P}(Q_{t+1} = j | Q_t = i) \\ &= \sum_{j=1}^K A_{ij} b_j^{t+1} \beta_j^{t+1}, \end{aligned} \quad (14)$$

for $t = N-1, N-2, \dots, 1$, and the recursion starts with $\beta_j^N = 1$.

Obviously, we derive

$$p(X^{(N)}, Q_t = j) = \alpha_j^t \beta_j^t. \quad (15)$$

Auxiliary Variables

Since the state variables $S_{t,k}$ are unknown, we replace them by their conditional expectations. To this end we compute the posterior probability of being in state i at time t given the entire sequence of observations and the parameters of the model.

$$\begin{aligned} \gamma_j^t &= \mathbb{P}(Q_t = j | X^{(N)}) = \frac{p(X^{(N)}, Q_t = j)}{p(X^{(N)})} \\ &= \frac{p(X^{(N)}, Q_t = j)}{\sum_{k=1}^K p(X^{(N)}, Q_t = k)} = \frac{\alpha_j^t \beta_j^t}{\sum_{k=1}^K \alpha_k^t \beta_k^t}. \end{aligned} \quad (16)$$

Mark that γ_j^t is the conditional expectation of $S_{t,j}$ given the whole data $X^{(N)}$ as the coordinates of the state vector S_t are 0-1-variables, i.e.

$$\mathbb{E}\{S_{t,j} | X^{(N)}\} = \mathbb{P}(S_{t,j} = 1 | X^{(N)}) = \gamma_j^t. \quad (17)$$

Finally, the joint conditional probability $\xi_{ij}^{t,t+1} = \mathbb{P}(Q_t = i, Q_{t+1} = j | X^{(N)})$ of Q_t and Q_{t+1} is given as follows

$$\begin{aligned} \xi_{ij}^{t,t+1} &= \mathbb{P}(Q_t = i, Q_{t+1} = j | X^{(N)}) \\ &= \frac{p(X^{(N)}, Q_t = i, Q_{t+1} = j)}{p(X^{(N)})} = \frac{A_{i,j} \alpha_i^t b_j^{t+1} \beta_j^{t+1}}{\sum_{k=1}^K \alpha_k^t \beta_k^t}, \end{aligned} \quad (18)$$

since

$$\begin{aligned} &p(X^{(N)}, Q_t = i, Q_{t+1} = j) \\ &= p(X_{t+2}, \dots, X_N | X^{(t+1)}, Q_t = i, Q_{t+1} = j) p(X^{(t+1)}, Q_t = i, Q_{t+1} = j) \\ &= p(X_{t+2}, \dots, X_N | \mathbf{X}_{t+1}, Q_{t+1} = j) p(X^{(t+1)}, Q_t = i, Q_{t+1} = j) \\ &= \beta_j^{t+1} p(X_{t+1} | X^{(t)}, Q_t = i, Q_{t+1} = j) p(X^{(t)}, Q_t = i, Q_{t+1} = j) \\ &= \beta_j^{t+1} p(X_{t+1} | \mathbf{X}_t, Q_{t+1} = j) \mathbb{P}(Q_{t+1} = j | Q_t = i, X^t) p(X^{(t)}, Q_t = i) \\ &= A_{i,j} \alpha_i^t b_j^{t+1} \beta_j^{t+1}. \end{aligned}$$

To estimate the conditional expectation of the state variables $S_{t,i}$ by estimating γ_i^t does not suit all purposes as we do not only use the past information up to time t but the entire training set. Therefore, those estimates are non causal or "offline". If one is interested, e.g., in forecasting, a causal or "online" version is more convenient, which may be obtained through a few computational stages, which we summarize as follows.

$$\begin{aligned} \mathbb{E}\{S_{t,k} | X^{(t-1)}\} &= \mathbb{P}(Q_t = k | X^{(t-1)}) = \frac{p(X^{(t-1)}, Q_t = k)}{p(X^{(t-1)})} \\ &= \frac{p(X^{(t-1)}, Q_t = k)}{\sum_{j=1}^K p(X^{(t-1)}, Q_t = j)} = \frac{\sum_{i=1}^K p(X^{(t-1)}, Q_{t-1} = i, Q_t = k)}{\sum_{j=1}^K \sum_{i=1}^K p(X^{(t-1)}, Q_{t-1} = i, Q_t = j)} \\ &= \frac{\sum_{i=1}^K \alpha_i^{t-1} A_{i,k}}{\sum_{j=1}^K \sum_{i=1}^K \alpha_i^{t-1} A_{i,j}}. \end{aligned}$$

Remark that from the Forward-Backward Procedure we can derive the estimates of the state variables and additionally obtain those of the transition probability matrix and of the initial distribution as well. Therefore we can say that we have a first step optimization in which the transition probability matrix and the initial distribution are the byproducts. Now we need to complete the estimation procedure in order to obtain a full set of parameters for the model.

2.1.2 Maximization or M-step

In this section, we consider the state variables Q_t or, equivalently, $S_{t,k}, k = 1, \dots, K$, to be known; in the iteration scheme the latter will be replaced by preliminary estimates of their conditional expectations $\mathbb{E}\{S_{t,k} | X^{(N)}\}$ given the data, i.e. by γ_i^k , compare (17), calculated during the E-step. We use them to get estimates of the transition matrix A and the network parameters ϑ by maximizing the complete log likelihood

$$\ell_c(\vartheta, A | X^{(N)}, Q^{(N)}) = \ell_1(A | Q^{(N)}) + \ell_2(\vartheta | X^{(N)}, Q^{(N)})$$

as given above. The estimates of A_{ij} and of the stationary probabilities of the Markov chain can be calculated from the auxiliary quantities of the E-step, and we get

$$\begin{aligned}\hat{A}_{ij} &= \frac{\text{Expected number of transitions from state } i \text{ to state } j}{\text{Expected number of transitions from } i \text{ to anywhere}} \\ &= \frac{\sum_t \xi_{ij}^{t,t+1}}{\sum_t \gamma_i^t}\end{aligned}\quad (19)$$

and

$$\hat{\pi}_i = \frac{1}{N} \sum_t \gamma_i^t. \quad (20)$$

To maximize $\ell_2(\vartheta|X^{(N)}, Q^{(N)})$ w.r.t. $\vartheta = (\theta_1, \dots, \theta_K, \theta_1^*, \dots, \theta_K^*)$, we have to minimize, compare (6),

$$G(\vartheta) = \sum_{t=1}^N \sum_{k=1}^K S_{tk} \left(\log f_k^*(\mathbf{X}_{t-1}; \theta_k^*) + \frac{1}{2} \left(\frac{X_t - f_k(\mathbf{X}_{t-1}, \theta_k)}{f_k^*(\mathbf{X}_{t-1}; \theta_k^*)} \right)^2 \right) \quad (21)$$

where θ_k, θ_k^* are the network parameter vectors of the neural network output functions f_k, f_k^* resp. The first order derivatives w.r.t. ϑ can be written as it follows

$$\frac{\partial G(\vartheta)}{\partial \theta_{k,i}} = - \sum_{t=1}^n S_{t,k} \frac{\partial f_k(\mathbf{X}_{t-1}, \theta_k)}{\partial \theta_{k,i}} \frac{X_t - f_k(\mathbf{X}_{t-1}, \theta_k)}{(f_k^*(\mathbf{X}_{t-1}, \theta_k^*))^2}$$

and

$$\frac{\partial G(\vartheta)}{\partial \theta_{k,j}^*} = \sum_{t=1}^n S_{t,k} \frac{\partial f_k^*(\mathbf{X}_{t-1}, \theta_k^*)}{\partial \theta_{k,j}^*} \frac{1}{f_k^*(\mathbf{X}_{t-1}, \theta_k^*)} \left(1 - \frac{(X_t - f_k(\mathbf{X}_{t-1}, \theta_k))^2}{(f_k^*(\mathbf{X}_{t-1}, \theta_k^*))^2} \right).$$

Numerically, we can retrieve the network parameters by using a stochastic approximation algorithm as e.g. a stochastic gradient algorithm.

To conclude this discussion, we look at a special case where the volatility functions are constant but perhaps different from state to state, i.e. $\sigma_k^2(\mathbf{x}) = \sigma_k^2$. In that case, we do not need a neural network f_k^* , but may estimate the parameter σ_k^2 directly. From solving

$$\frac{\partial G(\vartheta)}{\partial \sigma_k^2} = 0$$

we derive

$$\hat{\sigma}_k^2 = \frac{\sum_{t=1}^N S_{t,k} (X_t - f_k(\mathbf{X}_{t-1}, \theta_k))^2}{\sum_{t=1}^N S_{t,k}}.$$

Intuitively, that is just the usual residual variance estimate of the k^{th} subsample in the mixture models. Similarly, solving

$$\frac{\partial G(\vartheta)}{\partial \theta_{k,i}} = 0$$

is equivalent to solving

$$\sum_{t=1}^N S_{t,k} \frac{\partial f_k(\mathbf{X}_{t-1}, \theta_k)}{\partial \theta_{k,i}} (X_t - f_k(\mathbf{X}_{t-1}, \theta_k)) = 0$$

For this special case, we observe that for σ_k^2 we have obtained an analytical formula; but this representation depends on the unknown autoregressive functions which under our considerations are parametric functions. Once more we can retrieve these parameters by using a stochastic gradient algorithm.

2.1.3 An Adaptation of the Expectation Maximization Algorithm

The procedures we presented in the Forward-Backward Procedure and the maximization steps can be summarized to the following version of the well-known EM-algorithm, which we will call *EM-Algorithm for GMAR-ARCH models*.

1. Set $m = 0$ and choose initial values $\hat{\vartheta}(0), \hat{A}(0)$ for the parameters ϑ, A . Get initial values of the stationary probabilities π from $\hat{\pi}(0)\hat{A}(0) = \hat{\pi}(0)$.
2. (Expectation or **E-Step**)
Assume that the parameters of the model are equal to $\vartheta = \hat{\vartheta}(m), A = \hat{A}(m)$. Compute (for each time instant t) the forward variables $\alpha_k^t(m)$ and the backward variables $\beta_k^t(m)$ from (11), (14). Calculate the auxiliary variables $\gamma_i^t(m)$ and $\xi_{ij}^{t,t+1}(m)$ from (16), (18).
3. (Maximization or **M-Step**)
Calculate the updated estimate $\hat{A}(m+1), \hat{\pi}(m+1)$ from (19), (20) using the current values $\gamma_i^t(m), \xi_{ij}^{t,t+1}(m)$ of the auxiliary variables. Calculate the updated estimate $\hat{\vartheta}(m+1)$ of the network parameter vector by maximizing $-G(\theta)$ of (21) where the $S_{s,k}$ are replaced by the current estimates $\gamma_k^t(m)$ of their conditional expectations given the data.
4. Replace m by $m+1$ and repeat the procedure starting from the **E-Step** until a stopping criterion is satisfied.

2.2 The Viterbi Algorithm

The EM algorithm provides estimates $(\gamma_1^t(m), \dots, \gamma_k^t(m))$ of the state vectors S_t which, however, are no unit vectors. If one is interested in an estimate of the whole sequence of hidden data, then, the Viterbi algorithm may be used. It computes the optimal (most likely) state sequence in a Hidden Markov model given the sequence of observed outputs. It is based on the maximization of the single best state sequence using ideas from dynamic programming. For further discussion, we refer to, e.g., Cappé et al. (2005), chapter 5.1.

In our case to find the single best state sequence $\{S_1, S_2, \dots, S_N\}$ corresponding to the observations $\{X_1, X_2, \dots, X_N\}$ we define

$$\delta_t(i) = \max_{q_1, \dots, q_{t-1}} \log p(X^{(t)}, q_1, q_2, \dots, q_{t-1}, Q_t = i),$$

i.e. $\delta_t(i)$ is the highest log-likelihood along a single path up to time t , which accounts for the first t observations and ends in state $Q_t = i$. By induction we have

$$\begin{aligned} \delta_{t+1}(j) &= \max_{q_1, \dots, q_t} \log p(X^{(t)}, q_1, q_2, \dots, q_t, Q_{t+1} = j) \\ &= \max_{q_1, \dots, q_t} \log \left\{ \mathbb{P}(Q_{t+1} = j \mid Q_t = q_t) p(X_{t+1} \mid \mathbf{X}_t, Q_{t+1} = j) p(X^{(t)}, q_1, \dots, q_t) \right\}, \end{aligned}$$

i.e., writing i instead of q_t ,

$$\delta_{t+1}(j) = \max_i (\delta_t(i) + \log A_{i,j}) + \log b_j^{t+1},$$

compare (12). To retrieve the state sequence, we need to follow the trajectory constructed successively from the argument that maximized the previous equation for each t and j . We will achieve this via an auxiliary variable $\psi_t(j)$ such that the complete procedure can be written as follows

1. Initialization:

$$\begin{aligned}\delta_1(j) &= \log \pi_j b_j^1, \quad 1 \leq j \leq K, \\ \psi_1(j) &= 0.\end{aligned}$$

2. Recursion:

$$\begin{aligned}\psi_t(j) &= \arg \max_{1 \leq i \leq K} (\delta_{t-1}(i) + \log A_{i,j}), \quad 2 \leq t \leq N, \quad 1 \leq j \leq K, \\ \delta_t(j) &= \max_{1 \leq i \leq K} (\delta_{t-1}(i) + \log A_{i,j}) + \log b_j^t, \\ &= \delta_{t-1}(\psi_t(j)) + \log A_{\psi_t(j),j} + \log b_j^t, \quad 2 \leq t \leq N, \quad 1 \leq j \leq K\end{aligned}$$

3. Termination:

$$q_N^* = \arg \max_{1 \leq i \leq K} (\delta_N(i))$$

4. Path (State Sequence) Backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = N-1, N-2, \dots, 1.$$

Replacing the unknown model parameters in this algorithm by the estimates which we have from the EM algorithm, q_1^*, \dots, q_N^* are the estimates of the states Q_1, \dots, Q_N .

3 Applications

In this section we illustrate the algorithms developed in the previous section by applying them to simulated and real data. To this extent, we first consider computer generated data, where the underlying hidden Markov chain is known. Later, we apply the CHARME model to the daily stock price series of BASF and then derive for this case a first trading strategy that we compare with a classical Buy and Hold strategy.

3.1 Simulated data

In this section, we consider the following model with $K = 2$ states and autoregressive and volatility functions of order $M = 1$:

$$X_t = \begin{cases} m_1(X_{t-1}) + \sigma_1(X_{t-1})\varepsilon_t & \text{if } S_t = 1 \\ m_2(X_{t-1}) + \sigma_2(X_{t-1})\zeta_t & \text{if } S_t = 0, \end{cases} \quad (22)$$

where $S_t \in \{0, 1\}$ is a first order Markov Chain with transition probability matrix A , and the processes ε_t, ζ_t are independent sequences of i.i.d. $\mathcal{N}(0, 1)$ random variables. The transition probability matrix is given by

$$A = \begin{pmatrix} 0.985 & 0.015 \\ 0.015 & 0.985 \end{pmatrix}$$

and we choose a bump function and a decreasing logistic function

$$m_1(x) = \alpha x + \beta e^{-\gamma(x-\mu)^2} \quad m_2(x) = \frac{e^{v-x}}{1 + e^{v-x}}$$

for the autoregressive components of the model and ARCH(1) volatilities

$$\sigma_1(x) = \sqrt{\omega_1 + a_1 x^2} \quad \sigma_2(x) = \sqrt{\omega_2 + a_2 x^2}.$$

where $(\alpha, \beta, \gamma, \nu) \in \mathbb{R}^4$, $\omega_1 > 0, \omega_2 > 0$ and $a_1 \geq 0, a_2 \geq 0$. Making use of the Markov structure, we generate a sample of length $N = 3000$ of the hidden process S_t determining which dynamics is used at each time instant for generating the observed switching process.

To estimate $m_k, \sigma_k, k = 1, 2$, we approximate them by single layer feedforward networks. We also are interested in the estimation of the switching times where the hidden process S_t changes its value from 0 to 1 and vice versa. To get a decent comparison of the performance of our method, the usual technique from neural network estimation is applied where we the sample of size 3000 into two subsets, namely the training set and the validation set. The first 1100 observations are used as training set and the remainder are used as validation set. The results of the simulation are presented in the following picture.

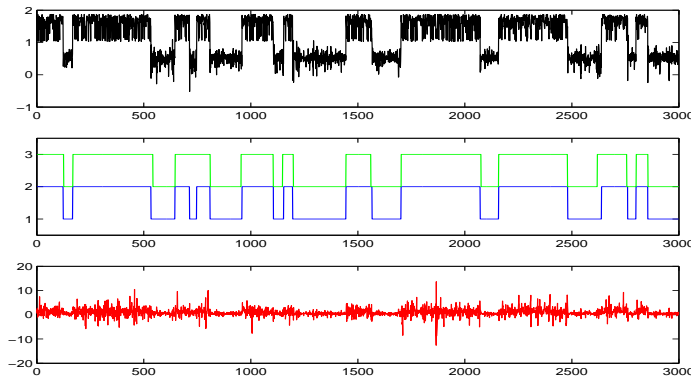


Fig. 1 Computer Generated Data and hidden process estimation

Figure 1 is composed of three subplots. The third below shows the generated process based on the model defined in equation. For this process we first generate a hidden Markov process that is represented by the tiny curve in the middle plots. Indeed, this curve represents the shifted hidden process, i.e. $S_t + 3$, for better visibility. The bold curve in this middle plot is the estimated hidden process retrieved by the Viterbi algorithm described above. The picture on top illustrates the estimated trend functions value at each time instant.

Up to a small perturbation (wrong alarm), that arises in a short period somewhere within the time span between 500 and 700, the estimated hidden process fits pretty well the true one. Furthermore, one can also observe with the estimated trend functions that at each time instant where the second state is detected, the estimated value of the trend function is positive, what is in line with the model definition.

3.2 BASF Daily Stock Data and a First Trading Strategy

For the practical application, we use BASF daily stock prices Y_t downloaded from <http://www.corporate.basf.com/en/investor/aktie/kurs.htm> for the period from July 1, 1996,

to October 12, 2004. We standardize them as

$$X_t = \log Y_t - \log Y_1,$$

and we fit the following model with $K = 3$ and $M = 1$:

$$X_t = \sum_{k=1}^3 S_{tk} (m_k(X_{t-1}) + \sigma_k(X_{t-1}) \varepsilon_t)$$

with

$$S_{tk} = \begin{cases} 1 & \text{if } Q_t = k \\ 0 & \text{otherwise} \end{cases}$$

where as before ε_t are i.i.d. (0,1) random variables and $\{Q_t\}$ is the non observable hidden Markov chain with values on $\{1, 2, 3\}$. Again, we use single layer feedforward neural networks functions with $H = 5$ hidden neurons each to estimate the autoregressive and volatility functions $m_k, \sigma_k, k = 1, 2, 3$, of the model. Again, we split the whole sample in a training set, consisting of the first 1000 data and used in estimation and in reconstructing the hidden states, and a validation set, consisting of the remaining data and used for evaluating the performance of the fitted model. The step (green) function illustrates the estimated hidden process, which in this case is completely unknown, the blue curve describes the transformed BASF stock values and the red curve, which is pretty much close to the blue one, illustrates the estimated trend function at each time instant.

As one can observe, the third regime appears only for a short period of time that corresponds to a big drop in the stock price. This period of occurrence of this third network corresponds to the period around September 11, 2001, and it is quite plausible that the stock market has been in a unique state at that time. Furthermore, we can observe that everywhere else, the model seems to be quite stable and spends most of the time in the first regime which corresponds to a low volatile market whereas the second state shows up in periods of high volatility.

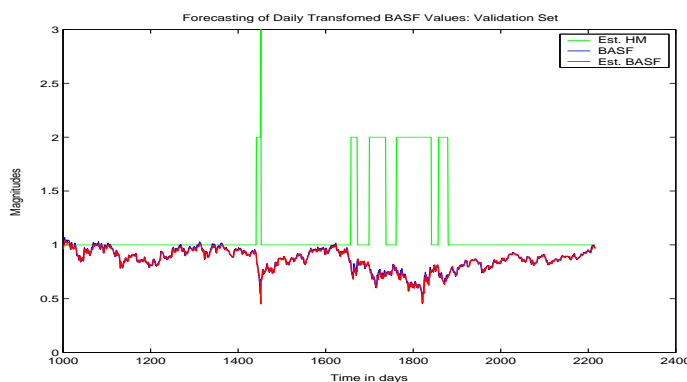


Fig. 2 Estimated Hidden Process and trend functions

Based on the fitted model, we now apply the estimated hidden process and the autoregressive functions to find a trading strategy which we compare to a common buy and hold

strategy. Every day, the investor has the choice between investing his whole capital either into BASF stock or into a zero bond. In comparing both strategies, we use as a simplifying assumption that there are no transaction costs and that the interest rate is 0, i.e. there is no return on investing into the bond. Let $\hat{f}_k(x)$ denote the neural network estimates of $m_k(x)$, $k = 1, 2, 3$, and let \hat{Q}_t be the value of the state sequence reconstructed by means of the Viterbi algorithm. We consider the difference $\Delta_t = \hat{f}_{\hat{Q}_t}(X_{t-1}) - X_{t-1}$ between the predicted log-price after one period and the actual log price, and if it is positive, we invest the whole wealth into the stock at day t whereas, otherwise, we invest it all into the bond. If we start with wealth G_0 at $t = 0$, then the wealth process evolves as

$$G_t = \begin{cases} \frac{Y_t}{Y_{t-1}} G_{t-1} & \text{if } \Delta_t > 0 \\ G_{t-1} & \text{else.} \end{cases}$$

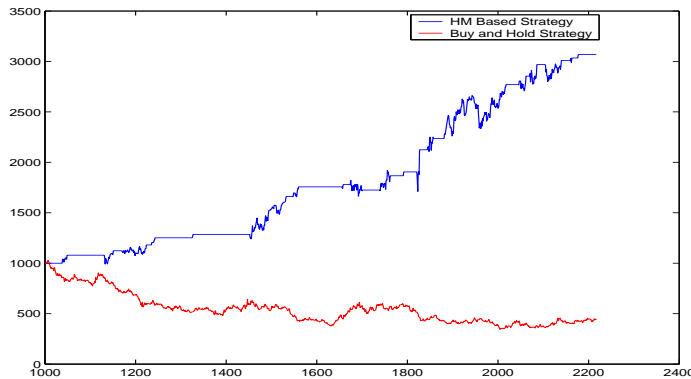


Fig. 3 Strategies for BASF Daily Stock Values

Given a starting capital of 1000 units of money, we apply this strategy on the validation set used previously and compare the results to a classical buy and hold strategy where all the wealth is invested into the stock which, then, is held until the end of the period. The results are illustrated in figure 3. Where, the continuous (blue) curve (below) represent the buy and hold strategy and the (red) curve on top represents the results using the CHARME based trading strategy. As one can observe, the difference is quite significant and indicates some promise for the use of portfolio management strategies based on nonparametric Markov switching models, even when transaction costs are taken into account.

4 Asymptotics of the network parameters

To illustrate how the asymptotics of the neural network based estimates for CHARME models can be investigated, we have a look at the estimate $\hat{\vartheta}$ of the parameters of both networks used in estimating the autoregressive and volatility functions. Its limit behavior can be derived from Theorem 14.1 of Pötscher and Prucha (1997) in a rather straightforward manner. Recall that $\hat{\vartheta}$ is calculated by minimizing $G(\vartheta)$ of (21) where, here, we assume the state vectors S_t to be given. We assume that the data X_t are generated by a CHARME model (1).

For the purpose of estimation, we pretend that the autoregressive and volatility functions are given by neural networks and that the innovations ε_t are i.i.d. standard normal, i.e. we fit model (5) to the data which, in general, will be misspecified.

Throughout this section we shall assume identifiability and uniqueness of the parameters. Sufficient identifiability conditions for the subclass of pure nonparametric autoregressive switching models are provided in Stockis, Tadjuidje and Franke (2008) and can be easily transferred to the case of nonconstant volatilities. There, the usual conditions of identifiability in the neural networks context are combined with some general conditions on the transition probability matrix of the hidden state process. We now formulate the additional assumptions needed for the consistency of $\hat{\vartheta}$.

A. 3 (Mixing Condition)

The Markov chain $\{Q_t\}$ is stationary, and the joint process $\{(X_t, Q_t)\}$ is α -mixing.

Mark that we do not necessarily assume that the observed process is in the stationary state. Stockis, Franke and Tadjuidje (2010) provide sufficient conditions for β -mixing of $\{(X_t, Q_t)\}$ with geometrically decreasing rate. In particular, it is not necessary that all the involved single AR-ARCH-processes

$$Y_t = m_k(Y_{t-1}, \dots, Y_{t-M}) + \sigma_k(Y_{t-1}, \dots, Y_{t-M})\varepsilon_t, \quad k = 1, \dots, K,$$

admit a stationary solution in order to ensure a stationarity solution of the switching process $\{X_t\}$. Some of them may even be explosive, provided they occur rarely enough.

A. 4 (Moment assumption)

$$\sup_n \frac{1}{n} \sum_{t=1}^n \mathbb{E}|X_t|^{2+\gamma} < \infty \text{ for some } \gamma > 0$$

A. 5 (Regularity Assumptions)

1. Assume that $\bar{\Theta} = \Theta \times \Theta^* \subseteq \mathbb{R}^{2K(H(M+2)+1)}$, the set of all admissible parameters $\vartheta = (\theta, \theta^*)$, is compact.
2. Assume that the activation functions ψ of the neural networks, used for approximating the autoregressive and volatility functions, are continuously differentiable on \mathbb{R} and bounded by 1 in absolute value.
3. There exists an $\delta > 0$ such that $f_k^*(u) \geq \delta$ for all $u \in \mathbb{R}^M$, $\theta_k^* \in \Theta^*$ and $k = 1, \dots, K$.

These assumptions are still quite standard for proving consistency of sieve estimates based on neural networks. At this point, we need to introduce some notation in line with Pötscher and Prucha (1997). Let

$$R_N(\vartheta) = -\frac{1}{N}G(\theta), \quad Z_t = (X_t, \mathbf{X}_{t-1}, S_t)$$

and define

$$q(Z_t, \vartheta) = -\sum_{k=1}^K S_{tk} \left(\log f_k^*(\mathbf{X}_{t-1} | \theta_k^*) + \frac{1}{2} \left(\frac{X_t - f_k(\mathbf{X}_{t-1} | \theta_k)}{f_k^*(\mathbf{X}_{t-1} | \theta_k^*)} \right)^2 \right),$$

such that the empirical risk is given by

$$R_N(\vartheta) = \frac{1}{N} \sum_{t=1}^N q(Z_t, \vartheta).$$

Let us also denote the expected risk as

$$\bar{R}_N = \frac{1}{N} \sum_{t=1}^N \mathbb{E}q(Z_t, \vartheta).$$

Let $\hat{\vartheta}_N$ be the sequence of network parameter estimates which we get by minimizing $G(\vartheta)$ resp. maximizing $R_N(\vartheta)$, and let $\bar{\vartheta}_N$ be any sequence of minimizers of \bar{R}_N . Then, we get the following consistency.

Theorem 1 *Assume A.1 to A.5 hold, then*

$$\sup_{\vartheta \in \bar{\Theta}} |R_N(\vartheta) - \bar{R}_N(\vartheta)| \rightarrow 0$$

in probability as $N \rightarrow \infty$, and $\{\bar{R}_N : N \in \mathbb{N}\}$ is equicontinuous on $\bar{\Theta}$. Additionally,

$$|\hat{\vartheta}_N - \bar{\vartheta}_N| \rightarrow 0$$

in probability as $n \rightarrow \infty$, i.e. $\hat{\vartheta}_N$ is consistent for $\bar{\vartheta}_N$.

Remark 1 Mark that $\bar{R}_N = \mathbb{E}q(Z_1, \vartheta)$ if the observed process $\{X_t\}$ is stationary. In that case, $\bar{\vartheta}_N = \bar{\vartheta}$ minimizes $\mathbb{E}q(Z_1, \vartheta)$ and does not depend on N .

Proof The result follows from Theorem 14.1 of Pötscher and Prucha (1997), and we only have to check the assumptions. First, let us rewrite our model in the following form

$$F_t(X_t, \mathbf{X}_{t-1}, S_t, \vartheta) = \varepsilon_t$$

with, writing $s = (s_1, \dots, s_K)'$,

$$F_t(u, \mathbf{x}, s, \vartheta) = \sum_k s_k \frac{u - f_k(\mathbf{x})}{f_k^*(\mathbf{x})}$$

and hence

$$\frac{\partial F_t}{\partial u}(u, \mathbf{x}, s, \vartheta) = \sum_k \frac{s_k}{f_k^*(\mathbf{x})}.$$

Considering assumption 14.2 of Pötscher and Prucha (1997), an immediate consequence of A.3 and A.4 ensures the existence of an identifiable unique sequence of minimizers of $\bar{R}_N(\vartheta)$. It remains to verify that their assumption 14.1 is satisfied.

Obviously, by our assumption of the activation function ψ in the network functions f_k and f_k^* , $F_t(u, \mathbf{x}, s, \vartheta)$ and $\frac{\partial F_t}{\partial u}(u, \mathbf{x}, s, \vartheta)$ are continuous functions that do not depend on t explicitly. Hence, $\{F_t : t \in \mathbb{N}\}$ and $\{\log \frac{\partial F_t}{\partial u} : t \in \mathbb{N}\}$ are equicontinuous on $V \times \bar{\Theta}$ where $V = \mathbb{R}^{M+1} \times \{1, \dots, K\}$. Also,

$$\sup_t |F_t(u, \mathbf{x}, s, \vartheta)| < \infty, \text{ for all } (u, \mathbf{x}, s, \vartheta) \in V \times \bar{\Theta}$$

Considering δ to be the lower volatility bound introduced in A.5, we have

$$|F_t(u, \mathbf{x}, s, \vartheta)| \leq \frac{1}{\delta} \sum_k |u - f_k(\mathbf{x})|$$

and since Θ is compact, it follows the existence of an $C > 0$ such that for all \mathbf{x}, k, θ

$$u - C \leq u - f_k(\mathbf{x}) \leq u + C.$$

Hence,

$$|X_t - f_k(\mathbf{X}_{t-1})| \leq |X_t - C| + |X_t + C| \text{ a.s.}$$

and therefore

$$\sum_k |X_t - f_k(\mathbf{X}_{t-1})| \leq K(|X_t - C| + |X_t + C|) \text{ a.s.}$$

Finally

$$\begin{aligned} & \sup_N \frac{1}{N} \sum_{t=1}^N \mathbb{E} \sup_{\theta \in \Theta} \left(\sum_k |X_t - f_k(\mathbf{X}_{t-1})| \right)^{2\gamma+2} \\ & \leq \sup_N \frac{1}{N} \sum_{t=1}^N \text{const} (\mathbb{E}|X_t - C|^{2\gamma+2} + \mathbb{E}|X_t + C|^{2\gamma+2}) < \infty \end{aligned}$$

A similar kind of argument leads to

$$\sup_N \frac{1}{N} \sum_{t=1}^N \mathbb{E} \sup_{\vartheta \in \Theta} \left(\left| \log \frac{\partial F_t}{\partial u}(X_t, \mathbf{X}_{t-1}, S_t, \vartheta) \right|^{1+\gamma} \right) < \infty$$

and therefore to the conclusion.

5 Conclusion

We have developed an algorithm for estimating the parameters of a Markov switching non-parametric AR-ARCH process where the autoregressive and volatility functions are approximated by feedforward neural networks. A Viterbi algorithm additionally allows to solve the filtering problem, i.e. to get an estimate of the hidden sequence of states. An application to a simple portfolio management problem shows some promise of this approach. Of course, there are some open questions to be solved, in particular the model selection problem, i.e. determining the number of different states K , the order of the autoregressive and ARCH schemes, here assumed to be all equal to M , and the number of neurons H of the neural networks. Those questions are addressed in future research. For practical purposes, it would also be of interest to add exogenous variables to the model, e.g., in the case of stock prices, to include indexes, interest or FX rates and other information into the argument of the functions m_k, σ_k additional to past prices or returns of the stock of interest (compare Franke and Diagne (2001) for such an approach in the nonswitching case $K = 1$).

References

1. L. E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Statist.*, 41, 164-171, (1970).
2. P. Brockwell and R. Davis. *Time Series: Theory and Methods*. Springer Verlag, Berlin, Heidelberg, New York, (1991).
3. O. Cappé, E. Moulines and T. Ryden. *Inference in Hidden Markov Models*. Springer Verlag, Berlin, Heidelberg, New York, (2005).

4. A. Dempster, N. Laird, and D. Rubin. Maximum likelihood for incomplete data via the em algorithm. *J. Royal Statist. Soc. B*, 39, 1-38, (1977).
5. C. Francq and M. Roussignol. On white noises driven by hidden Markov chains. *J. Time Series Anal.*, 18, 553-578, (1997).
6. C. Francq and M. Roussignol. Ergodicity of autoregressive processes with Markov switching and consistency of the MLE. *Statistics*, 32, 151-173, (1998).
7. C. Francq, M. Roussignol and J. M. Zakoian. Conditional heteroskedasticity driven by hidden Markov chains. *J. Time Series Anal.*, 22, 197-220, (2001).
8. J. Franke and M. Diagne. Estimating market risk with neural networks. *Statistics and Decisions*, 24, 233-253, (2006).
9. J. Franke, M. H. Neumann and J. P. Stockis. Bootstrapping nonparametric estimates of the volatility function. *Journal of Econometrics*, 118, 189-218 (2004).
10. J. Franke, J.-P. Stockis, J. T. Kamgaing, and W.K. Li Mixtures of nonparametric autoregressions. (2009) Tentatively accepted for publication in *J. Nonparametric Statistics*.
11. U. Grenander. *Abstract Inference*. Wiley, New York, (1981).
12. W. Härdle and A. B. Tsybakov. Local polynomial estimation of the volatility function. *J. Econometrics*, 81, 223-242, (1997).
13. Ch. Hafner. *Nonlinear Time Series Analysis with Applications to Foreign Exchange Rate Volatility*. Physica-Verlag, Heidelberg, (1998).
14. J. D. Hamilton. A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, 57, 357-384, (1989).
15. J. D. Hamilton and R. Susmel. Autoregressive conditional heteroskedasticity and changes in regime. *J. Econometrics*, 64, 307-333, (1994).
16. K.-R. Müller, J. Kohlmorgen, J. Rittweger, and K. Pawelzik. Analysing physiological data on wake-sleep state transition with competing predictors. *NOLTA 95: Las Vegas Symposium on Nonlinear Theory and its Applications, IEICE, Tokyo*, 223-226, (1995).
17. B. M. Pötscher and I. R. Prucha. *Dynamic Nonlinear Econometric Models: Asymptotic Theory*. Springer, Germany, (1997).
18. T. Rydén, T. Teräsvirta and S. Asbrink. Stylized facts of daily return series and the hidden Markov model. *J. Appl. Econ.*, 13, 217-244, (1998).
19. J.-P. Stockis, J. Franke and J. Tadjuidje Kamgaing. On geometric ergodicity of CHARME models.(2010) To appear in *J. Time Ser. Anal.*, 31, 141-152 (2010).
20. J.-P. Stockis, J. Tadjuidje Kamgaing and J. Franke. A note on the identifiability of the conditional expectation for the mixtures of neural networks. *Statistics and Probab. Letters*, 78, 739-742 (2008).
21. H. White. *Asymptotic Theory for Econometricians*. Academic Press, New York, (1984).
22. C.S. Wong and W.K. Li. On a mixture autoregressive model. *J. Royal Statist. Soc. B*, 62, 95-115 (2000).
23. C.S. Wong and W.K. Li. On a mixture autoregressive conditional heteroscedastic model. *J. American Statist. Assoc.*, 96, 982-995 (2001).
24. C.F.J. Wu. On the convergence properties of the EM algorithm. *Ann. Statist.*, 11, 95-103, (1983).