IMA Journal of Management Mathematics (2010) **21**, 111–129
doi:10.1093/imaman/dpp009
Advance Access publication on June 11, 2009

Hidden Markov models for financial optimization problems

DIANA ROMAN† AND GAUTAM MITRA‡
Centre for the Analysis of Risk and Optimisation Modelling Applications, School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, UK and Optirisk Systems, Uxbridge, Middlesex UB9 4DA, UK

AND

NICOLA SPAGNOLO§
Department of Economics and Finance, School of Social Sciences, Brunel University, Uxbridge, Middlesex, UB8 3PH UK and Centre for Applied Macroeconomic Analysis, Canberra, Australia

[Received on 2 July 2007; accepted on 14 July 2008]

Many financial decision problems require scenarios for multivariate financial time series that capture their sequentially changing behaviour, including their extreme movements. We consider modelling financial time series by hidden Markov models (HMMs), which are regime-switching-type models. Estimating the parameters of an HMM is a difficult task and the multivariate case can pose serious implementation issues. After the parameter estimation, the calibrated model can be used as a scenario generator to describe the future realizations of asset prices. The scenario generator is tested in a single-period mean–conditional value-at-risk optimization problem for portfolio selection.

Keywords: scenario generation; asset pricing; hidden Markov models; extreme events; stability; conditional value at risk.

1. Introduction

A stochastic programming problem is an optimization problem in which some of the parameters are not known with certainty, but described by random variables (for single-period problems) or stochastic processes (for multiperiod problems). In most cases, it is not possible to construct computable optimization models with continuous distributions. The uncertain model parameters have to be approximated by discrete distributions with a finite number of outcomes (scenarios), or, in other words, scenarios have to be generated.

In recent years, financial optimization problems have been of great interest, due to their applicability in a range of areas, including portfolio selection and asset liability management. Portfolio choice continues to be a leading problem in finance and in recent times considerable attention has been given to the downside risk control. In order to take 'hedged' decisions, the downside risk has to be considered firstly in the scenario generation of asset price evolution and secondly in the stochastic optimization problem.

There are several optimization models that take into account various types of downside risk (see, e.g. Fishburn, 1977; Rockafeller & Uryasev, 2000; Mansini *et al.*, 2007). However, scenario generation methods based on classical financial time series models sometimes fail to capture extreme price movements.

†Corresponding author. Email: diana.roman@brunel.ac.uk
‡Email: gautam.mitra@brunel.ac.uk
§Email: nicola.spagnolo@brunel.ac.uk

Traditional models for stock prices assume that they follow a geometric Brownian motion: stock prices, which are stochastic processes in discrete time, are approximated by continuous time stochastic processes. This approximation is accepted by practitioners for short time periods and has become a standard in finance. However, empirical studies show that these models fail to capture extreme price movements (Hardy, 2001) and thus scenario generators based on such methods may not be appropriate for certain optimization problems, especially if downside risk is explicitly considered in the decision process. Among the most popular models for financial time series are the autoregressive conditional heteroskedasticity (ARCH)–generalized autoregressive conditional heteroskedasticity (GARCH) models (Engle, 1982; Bollerslev, 1986), which were designed for modelling time-dependent variance.

In this paper, we use a regime-switching approach to model asset prices. This approach is underpinned by two motivating aspects. Firstly, financial time series have a sequentially changing behaviour. The market may switch from time to time from, let us say, a state with low volatility to a state with high volatility (although the states do not necessarily have such a 'semantic' interpretation); each state is characterized by different model parameters. In this case, one single model is valid only for short periods; regime-switching models offer a better way of modelling. Secondly, extreme price movements can be captured by considering an appropriate 'state' or 'regime' (possibly, occurring with low probability) that generates such movements.

A hidden Markov model (HMM) is a particular regime-switching model, in which there are two stochastic processes involved: one related to the time series of interest and the other an underlying stochastic process describing the system's state over time, that is not observable (hidden).

HMMs were applied to a variety of fields, in particular to speech recognition (Rabiner, 1989); a review of HMMs and their applications is given in Ephraim & Merhav (2002). However, HMMs have not been much applied for financial time series modelling and forecasting; see Zhang (2004) for a review of the work undertaken in this regard. In Hamilton & Susmel (1994), weekly time series were modelled using regime switching with ARCH-type models within regimes. Hardy (2001) proposed a simpler regime-switching model for monthly returns with log-normal distributions. Shi & Weigend (1997), Azzouzi & Nabney (1999), Bengio *et al*. (2001) and Zhang (2004) consider financial time series modelling using HMM. Messina & Toscani (2008) use autoregressive HMM for modelling univariate financial time series and for generating scenarios to describe their possible future evolution. In order to validate the HMM-based scenario generation method, they show that the resulting Monte Carlo-sampled distributions can replicate with good approximation the empirical distribution of the observed data.

Our purpose is to model multivariate financial time series using HMM in order to generate scenarios to be integrated in financial decision problems. As already stated, we want to capture the extreme downside movements of asset prices. The resulting scenarios are used in a mean–conditional value-at-risk (CVaR) optimization problem (Rockafeller & Uryasev, 2000, 2002); thus, we consider downside risk explicitly in the decision model (CVaR is, roughly speaking, the mean of the losses in a prespecified number of worst cases).

It is known that the purpose of a scenario generator is not to approximate well the distributions involved (from a statistical point of view) but to give good approximations of the optimal solutions of the optimization problem. Methods of assessing the quality of a scenario generator are discussed later in the paper; a full treatment is given in Kaut & Wallace (2007).

The rest of the paper is organized as follows. In Section 2, HMMs are presented. The parameters of the HMM and the current state of the system are estimated using past data; subsequently, the calibrated model can be used to generate future scenarios. Section 3 describes the portfolio selection problem posed as a mean–CVaR model—which is the optimization problem in which the resulting scenarios are

integrated. Section 4 describes the numerical experiments; we test the stability of our scenario. Section 5 presents the conclusions.

2. Hidden Markov models

The rationale behind HMM is as follows:

- Real-world processes produce observable outputs (called signals, observations) at equally spaced discrete time points $t = 1, 2, \ldots$.
- At each time point $t = 1, 2, \ldots$, the system can be in one of N distinct states $S_1, \ldots, S_N$. Each state produces observations according to a probability density function (discrete or continuous). Given a time point, we do not know what state the system is in—the state is 'hidden'; we can only observe the output at that time point. Denote the underlying stochastic process by $\{q_t\}_{t=1,2,\ldots}$ (q_t = the system's state at time t). Thus, at each time t, an observation o_t is generated by a state $q_t = S_j$ according to a probability density function $b_j(o_t)$.
- At the initial time point ($t = 1$), the system can be in one of the N states according to some 'initial' probabilities; π_i = the probability that the system is in state S_i at time 1 $\left(\sum_{i=1}^{N} \pi_i = 1\right)$.
- At each subsequent time point, the system moves into another state or stays into the previous state according to some transition probabilities. For example, if the system is in state S_i at time $(t-1)$, we denote by a_{ij}^t the probability that the system transits from state S_i to state S_j at time t $\left(\sum_{j=1}^{N} a_{ij}^t = 1\right)$.

In addition, for ease of modelling, a number of other assumptions are usually made. The model described in this paper is based on the following additional assumptions:

- The Markov assumption. The underlying stochastic process is a first-order Markov chain: $P(q_t = S_i | q_{t-1} = S_j, q_{t-2} = S_k, \ldots) = P(q_t = S_i | q_{t-1} = S_j)$. In other words, the system's state at any time period depends only on the system's state at the previous time period, not on the entire history.
- The time independence assumption: a_{ij}^t (the probability that the system transits from state S_i to state S_j at time t) is the same at any time period t (thus, the time superscript can be dropped). The vector of initial probabilities $\pi = (\pi_1, \ldots, \pi_N)$, together with the matrix $A = (a_{ij})_{i,j=1,\ldots,N}$, fully characterize the evolution of the system's states.
- The output independence assumption: the output (observation) generated at time t depends only on the system's state at time t (meaning, the probability distribution that generates it) and not on the previous observations.

Under these assumptions, the parameters of an HMM are the following:

- The number of states N;
- The observation probability distribution functions b_i, for each state $i \in \{1, \ldots, N\}$;
- The initial probabilities vector $\pi = (\pi_i)_{i=1,\ldots,N}$;
- The transition probability matrix $A = (a_{ij})_{i,j=1,\ldots,N}$.

Our interest is in financial time series, e.g. the prices or returns of stocks. These returns depend on the 'state of economy', which is the underlying, unobservable stochastic process. We could give these states a 'semantic interpretation' (like, e.g. 'bad' or 'good', if $N = 2$), but the main point is that they generate observations according to probability distributions with different parameters.

We choose these probability distributions to be mixtures of normal distributions (Gaussian mixtures). By definition, a Gaussian mixture has a density function that can be written as the weighted sum of normal density functions:

$$f(x) = \sum_{j=1}^{M} c_j N(x; \mu_j, \sigma_j^2), \tag{1}$$

where $\sum_{j=1}^{M} c_j = 1$ and $N(x; \mu_j, \sigma_j^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$ is the (univariate[1]) density of the normal distribution with mean μ_j and standard deviation σ_j.

Thus, in the HMM, the density function corresponding to state $i \in \{1, \ldots, N\}$ is

$$b_i(x) = \sum_{j=1}^{M} c_{ij} N(x; \mu_{ij}, \sigma_{ij}^2).$$

The reason for choosing Gaussian mixtures is that they can approximate arbitrarily closely any finite, continuous density function (see Rabiner, 1989). It has been often argued that financial returns are not normally distributed: they have fatter tails, left skewed, etc. By choosing mixture of Gaussians instead of a single normal distribution, some of these drawbacks could be overcome.

Thus, in this case, in addition to the number of states N, the initial probabilities vector $\pi = (\pi_i)_{i=1,\ldots,N}$ and the transition probabilities matrix $A = (a_{ij})_{i,j=1,\ldots,N}$, the parameters of the HMM are

- the number of mixtures M;
- the matrix of mixture coefficients $C = (c_{ij})_{i=1,\ldots,N, j=1,\ldots,M}$ (the ith line of this matrix is composed of the mixture coefficients of state i);
- the matrix of means of the Gaussian mixtures $\Gamma = (\mu_{ij})_{i=1,\ldots,N, j=1,\ldots,M}$ (the ith line of this matrix is composed of the means of the Gaussian mixture corresponding to state i, i.e. $\mu_{ij} =$ the mean of the jth mixture in the ith state);
- the matrix of variances of the component Gaussian distributions $\Sigma = (\sigma_{ij}^2)_{i=1,\ldots,N, j=1,\ldots,M}$ ($\sigma_{ij}^2 =$ the variance of the jth mixture in the ith state).

We can estimate these parameters using past data. This process is called 'training' or 'learning' the HMM: parameters are estimated such that the likelihood that the past observations were produced by the model is maximized. Once the parameters of an HMM are estimated, the model can be used for scenario generation.

2.1 *Estimation of the parameters of an HMM*

The estimation of the parameters of an HMM is done using a sequence of past observations $O = (O_1, \ldots, O_T)$.

The number of states N and the number of mixtures M are estimated separately from the rest of the parameters.

Determining the number of Markov states (or regimes) required for a switching model to be an adequate characterization of the observed data is a key problem. In practice, the state dimension of the hidden Markov chain that drives regime changes is sometimes dictated by the actual application or is determined in an empirical manner by visual inspection of plots of the data (Messina & Toscani, 2008;

[1]We consider for the moment only the univariate case and treat the multivariate case in Section 2.3.

Geyer & Ziemba, 2008). A more formal and statistically sound procedure for determining the state dimension can be based on likelihood ratio (LR) tests. Such tests are problematic, however, because the usual regularity conditions are not fulfilled under the null hypothesis.[2] Instead of relying on hypothesis testing, one can consider methods based on complexity penalized likelihood criteria such as the Akaike information criterion (AIC) (Akaike, 1974). Using Monte Carlo analysis, Psaradakis & Spagnolo (2003) show that model selection procedures based on complexity penalized likelihood criteria are generally successful in choosing the correct state dimension.

There are several methods proposed for estimating the number of mixtures (see, e.g. Lee *et al.*, 2006). We consider here the k-means clustering algorithm, which is described in more detail in Section 2.1.3.

The rest of the parameters of an HMM are denoted by λ; in our case, $\lambda = (\pi, A, C, \Gamma, \Sigma)$.

The purpose of 'training' an HMM is to find the parameters λ such that $P(O|\lambda)$ (the likelihood that the sequence of observations was produced by the model) is maximized; the number of states and the number of mixtures are supposed to be known. The rest of Section 2.1 concerns this issue of estimating the parameters λ. In order to answer this question, additional quantities need to be defined and the so-called 'evaluation problem' be solved.

2.1.1 *The evaluation problem.* Given a model $\lambda = (\pi, A, C, \Gamma, \Sigma)$ and an observation sequence $O = (O_1, \ldots, O_T)$, how do we calculate $P(O|\lambda)$?
An efficient way for solving this problem is given by the 'forward–backward' procedure.
The 'forward variable' is defined as

$\alpha_t(i) = P(O_1, \ldots, O_t, q_t = i|\lambda)$, for all $t \in \{1, \ldots, T\}$ and $i \in \{1, \ldots, N\}$,

i.e. the probability of the partial observation sequence (until time t) and state i at time t, given the model λ.

$\alpha_t(i)$ is calculated inductively after time as follows:
Initialization:

$$\alpha_1(i) = \pi_i b_i(O_1) = \pi_i \sum_{k=1}^{M} c_{ik} N(O_1; \mu_{ik}, \sigma_{ik}^2), \quad \forall\, i \in \{1, \ldots, N\}.$$

Induction:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \quad \forall\, j \in \{1, \ldots, N\}, \ \ \forall\, t \in \{1, \ldots, T-1\}.$$

Then, $P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$.
$P(O|\lambda)$ may be calculated in a similar manner using the 'backward' variable

$\beta_t(i) = P(O_{t+1}, \ldots, O_T|q_t = i, \lambda)$, with $t \in \{1, \ldots, T-1\}$ and $i \in \{1, \ldots, N\}$.

The calculation of $\beta_t(i)$is also done inductively. We underline that this backward procedure is not necessary for calculating $P(O|\lambda)$ (it is just an alternative method to the forward procedure). However, this procedure is necessary for learning the HMM, so it will be described below.

[2]The null hypothesis of linearity (one state) against the alternative of a Markov regime-switching process (with two states or more) cannot be tested directly using a standard LR test. This is due to the fact that standard regularity conditions for likelihood-based inference are violated under the null hypothesis of linearity as the parameters of transition probabilities are unidentified and scores with respect to parameters of interest are identically zero. Under such circumstances, the information matrix is singular.

Initialization:

$$\beta_T(i) = 1, \quad \forall i \in \{1, \ldots, N\}.$$

Induction:

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad \forall j \in \{1, \ldots, N\}, \ \ t = T-1, T-2, \ldots, 1.$$

Then, $P(O|\lambda) = \sum_{i=1}^{N} \beta_1(i)$.

Thus, there are three ways of expressing $P(O|\lambda)$:

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \beta_1(i) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i), \quad \forall t \in \{1, \ldots, T\}.$$

2.1.2 *Training the HMM. Re-estimation formulas.* We return now to our original problem: given a sequence of T consecutive observations $O = (O_1, \ldots, O_T)$, how do we estimate the parameters of an HMM $\lambda = (\pi, A, C, \Gamma, \Sigma)$ such that $P(O|\lambda)$ is maximized?

There is no known way to analytically solve this problem. However, we can find λ such that $P(O|\lambda)$ is locally maximized using iterative procedures like Baum–Welch (Baum *et al.*, 1970) or, equivalently in this special case, expectation–maximization (EM) (Dempster *et al.*, 1977, see also Rabiner, 1989, and references within).

The basic idea is to start with some arbitrary values for λ and then 're-estimate' the parameters, obtaining a new set of 'improved' parameters λ^* such that $P(O|\lambda^*) \geqslant P(O|\lambda)$. The process is repeated, thus a sequence of parameters $\lambda_0, \lambda_1, \lambda_2, \ldots$ is obtained such that $P(O|\lambda_{i+1}) \geqslant P(O|\lambda_i)$. The sequence $(P(O|\lambda_i))_i$ converges to a local maximum (Rabiner, 1989).

We will list below the re-estimation formulas for $\lambda = (\pi, A, C, \Gamma, \Sigma)$. As pointed out in Rabiner (1989), these formulas are the result of both the Baum–Welch algorithm and the EM (expectation modification) algorithm, in the sense both algorithms reach to the same formulas for this particular case.

Some additional quantities, that appear in the re-estimation formulas, need to be defined in advance:

1. The probability of being in state i at time t and in state j at time $(t + 1)$, given the sequence of observations and the current model λ:

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j|O, \lambda), \quad \forall i, j \in \{1, \ldots, N\}, \ \ t = 1, \ldots, T-1.$$

Computation:

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}, \text{ with } b_j(O_{t+1}) = \left[\sum_{k=1}^{M} c_{jk} N(O_{t+1}; \mu_{jk}, \sigma_{jk}^2) \right].$$

2. The probability of being in state i at time t, given the sequence of observations and the current model λ:

$$\gamma_t(i) = P(q_t = i|O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}, \quad \forall i \in \{1, \ldots, N\}, \ \ t = 1, \ldots, T.$$

3. The probability of being in state j at time t, with the kth mixture accounting for O_t:

$$\gamma_t(j,k) = \gamma_t(j)\frac{c_{jk}N(O_t;\mu_{jk},\sigma_{jk}^2)}{\left[\sum_{m=1}^{M} c_{jm}N(O_t;\mu_{jm},\sigma_{jm}^2)\right]},$$
$$\times\ \forall t \in \{1,\ldots,T\},\ \ \forall j \in \{1,\ldots,N\},\ \forall k \in \{1,\ldots,M\}.$$

REMARK 1 It is clear that

$$\sum_{t=1}^{T-1}\gamma_t(i) = \text{expected number of transitions from state } i;$$
$$\sum_{t=1}^{T-1}\xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j.$$

REMARK 2 $\gamma_t(i)$, $\gamma_t(i,k)$ and $\xi_t(i,j)$ are indeed probabilities in the sense that they are between 0 and 1, $\sum_{i=1}^{N}\gamma_t(i)=1$, $\sum_{j=1}^{N}\xi_t(i,j)=\gamma_t(i)$, etc. In contrast, $\alpha_t(i)$, $\beta_t(i)$ and $P(O|\lambda)$ are 'likelihood' (density) values, thus they can have any positive value. This aspect may create computational difficulties when these values are too small or too large; we go into more detail of this problem in Section 2.3.

Finally, the re-estimation formulas for the parameters of an HMM are as follows:

$$\overline{\pi_i} = \gamma_1(i)(\text{expected frequency of state } i \text{ at time } t=1),$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}\left(=\frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i}\right),$$

$$\overline{c_{jk}} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)}{\sum_{t=1}^{T}\sum_{m=1}^{M}\gamma_t(j,m)} = \frac{\text{expected number of times in state } j \text{ due to mixture } k}{\text{exp ected number of times in state } j},$$

$$\overline{\mu_{jk}} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)O_t}{\sum_{t=1}^{T}\gamma_t(j,k)};\quad \overline{\sigma_{jk}^2} = \frac{\sum_{t=1}^{T}\gamma_t(j,k)(O_t-\mu_{jk})^2}{\sum_{t=1}^{T}\gamma_t(j,k)}.$$

From the re-estimation formulas for π and A, it is easy to see that the 'stochastic' conditions $\sum_{i=1}^{N}\overline{\pi_i}=1$, $\sum_{i=1}^{N}\overline{a_{ij}}=1$ and $\sum_{k=1}^{M}\overline{c_{jk}}=1$, $\forall j \in \{1,\ldots,N\}$, are already fulfilled; thus, there is no need of re-scaling these quantities.

As already stated, by repeatedly re-estimating the parameters λ, a sequence $\lambda_0,\lambda_1,\lambda_2,\ldots$ is obtained such that $\{P(O|\lambda_i)\}_i$ converges towards a local maximum. However, there is very little known about the convergence speed. Practically, it has been observed sharp increase for $P(O|\lambda_i)$ in the first iterations and after that little improvement (Cettolo, 2004). The stopping criterion in the re-estimation sequence is thus subjective; basically, the procedure is stopped when the relative likelihood improvement is significant.

2.1.3 *Training the HMM. Initial estimates.* Some initial values for λ are necessary in order to start the re-estimation procedure. It is known that the initial values for π and A are not important; we can choose

either equal probabilities ($1/N$) or random values, subject to stochastic and non-zero value constraints (Rabiner, 1989; Cettolo, 2004).

However, initial estimates for the parameters of the continuous output distributions (C, $\Gamma$and Σ) are essential. In spite of this, there is no generally 'correct' mathematical solution for this problem. Very few suggestions are made in the literature (one of them can be found in Rabiner) and they are only given as a loose guidance, not supposed to work in any situation.

We choose the initial values for λ by using a k-means clustering algorithm (Macqueen, 1967). The k-means algorithm is a popular method of clustering a set of (vector) data points into a number of K groups such that the 'intra-cluster' variance is minimized, i.e. if $S_1, \ldots, S_K$ are the clusters, with centres $\mu_1, \ldots, \mu_K$,[3] then $V = \sum_{i=1}^{K} \sum_{x_j \in S_i} d(x_j, \mu_i)$ is minimized, where $d(x_j, \mu_i)$ is (usually) the Euclidean distance between x_j and μ_i.

The idea of the algorithm is as follows: initially, cluster the data into K sets randomly. Calculate the centre of each cluster and then reassign each point of the data to the nearest cluster, i.e. to the cluster whose centre is closest. Repeat the previous two steps until no reassignment is made.

It is worth pointing out that, although the procedure always terminates, the solution provided by the algorithm is not necessarily a global minimum of the 'intra-cluster variance' function (it is a local minimum), nor it is unique: at different runs of the algorithm we may obtain different clustering. However, the algorithm performs well in practice and converges extremely quickly.

Thus, for the initial estimates in the HMM training, we choose the matrix of mixture coefficients C as having equal entries: $c_{ij} = \frac{1}{M}, \forall i = 1, \ldots, N, \forall j = 1, \ldots, M$. We partition the historical data $O = (O_1, \ldots, O_T)$ into a set of M clusters (using the k-means algorithm). The means and the (co)variances of the clusters obtained are used as initial estimates for the means and the (co)variances of the Gaussian mixtures (the same initial estimates for each state).

2.2 *Using HMM for scenario generation and simulation*

2.2.1 *Estimating the current state. The Viterbi algorithm.* Once the parameters λ of an HMM are estimated, this model can be used as a Monte Carlo scenario simulation model for financial data. A path can be generated in the following way: at each time step of the simulation process, given the current state S_i, we choose the next state S_j according to the transition probabilities A, then generate (by Monte Carlo sampling) an output corresponding to the observation distribution of state S_j.

For this purpose, we need to estimate the state of the system at the 'current' time T (i.e. the last time period of the historical data used to train the model). This can be done with the Viterbi algorithm (Forney, 1973).

The Viterbi algorithm is designed to answer a fundamental problem in HMM: given an observation sequence $O = (O_1, \ldots, O_T)$ and a model λ, how do we choose a corresponding state sequence path $Q = (q_1, \ldots, q_T)$ that best 'explains' the observations? (i.e. that maximizes $P(Q|O, \lambda)$?)

The following quantity needs to be defined:

$$\delta_t(i) = \max_{q_1, \ldots, q_{t-1}} P(q_1, \ldots, q_{t-1}, q_t = i, O_1, \ldots, O_t|\lambda), \quad \forall i \in \{1, \ldots, N\}, \ \forall t \in \{1, \ldots, T\},$$

i.e. $\delta_t(i)$ is the highest probability along a single path, at time t, which accounts for the first t observations and ends in state S_i.

By induction: $\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1})$.

[3]The centre of a group of vectors is considered as the arithmetic mean of the component vectors.

To retrieve the optimal state sequence, we must keep track of the value of i for which the maximization is obtained, in the formula above—denote it by $\psi_t(j)$. The whole algorithm for retrieving the optimal state sequence is as follows:
Initialization ($t = 1$):

$$\delta_1(i) = \pi_i b_i(O_1), \quad \forall\, i \in \{1, \ldots, N\},$$

$$\psi_1(i) = 0, \quad \forall\, i \in \{1, \ldots, N\}.$$

Induction:

$$\delta_t(j) = \left[\max_i \delta_{t-1}(i) a_{ij}\right] b_j(O_t), \quad \forall\, t \in \{2, \ldots, T\}, \ \forall\, j \in \{1, \ldots, N\},$$
$$\psi_t(j) = \left[\max_i \delta_{t-1}(i) a_{ij}\right], \quad \forall\, t \in \{2, \ldots, T\}, \ \forall\, j \in \{1, \ldots, N\}.$$

Termination:

$$p* = \max_i [\delta_T(i)] - \text{the best probability},$$
$$q_T* = \arg\max_i [\delta_T(i)] - \text{ the state at time } T \text{ that gives this probability}.$$

Path (state sequence) backtracking:

$$q_t* = \psi_{t+1}(q^*_{t+1}), \quad t = T - 1, \ldots, 1.$$

What is of interest in the scenario generation procedure is q_T*, which gives the 'most likely' state of the system at time period T.

2.2.2 *Summary of the scenario generation procedure.* The whole process of generating a path of length *TP* is represented in Fig. 1.

The procedure for generating S number of paths of length *TP* is summarized below:

1. Given a sequence of (past) observations $O = (O_1, \ldots, O_T)$, estimate the parameters $\lambda = (\pi, A, C, \Gamma, \Sigma)$:
   a. Find a set of initial estimates for λ: equal probabilities for π, A and C; Γ and Σ are found via a k-means clustering algorithm, as described in Section 2.1.3.

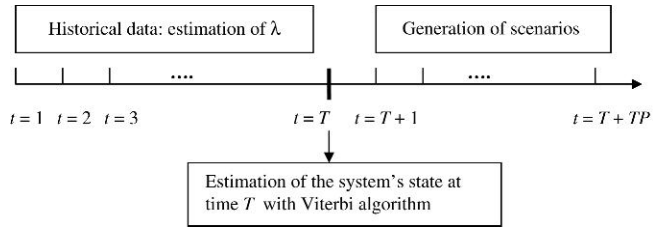

FIG. 1. The process of generating scenarios for *TP* time periods ahead, using T time periods of historical data.

b. Re-estimate λ using re-estimation formulas as described in Section 2.1.2 until the improvement in the likelihood value $P(O|\lambda)$ is satisfactory

2. Estimate the system's state at time T with the Viterbi algorithm; say, this state is S_i.
3. For $s = 1$ to S

{
Generate a sample from the output distribution of state i; $t = 1$
While ($t < TP$)
{
- $t = t + 1$
- choose a state for time t according to transition probability matrix A
- generate a sample from the output distribution of this state
}
}

2.3 *The multivariate case. Computational issues*

Suppose now that the time series of interest is multivariate, e.g. the joint returns of a number of NA stocks. The dependencies between the components are described by the covariance matrices.

The output distribution of state S_i has the density function

$$b_i(x) = \sum_{j=1}^{M} c_{ij} N(x; \mu_{ij}, \Sigma_{ij}),$$

where $\sum_{j=1}^{M} c_j = 1$, μ_{ij} is the mean vector (of dimension NA) corresponding to state i and mixture j; Σ_{ij} is the (NA-dimensional) covariance matrix corresponding to state i and mixture j and

$$N(x; \mu_{ij}, \Sigma_{ij}) = \frac{1}{(2\pi \det(\Sigma_{ij}))^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_{ij})^{\top} \Sigma_{ij}^{-1}(x - \mu_{ij})\right) \tag{2}$$

is the multivariate density of the normal distribution with vector mean μ_{ij} and covariance matrix Σ_{ij}.

There are several serious difficulties raised by the multivariate case, from a computational point of view. First, at every step of the re-estimation procedure, there are $N \cdot M$ matrices (of dimension NA) that need to be inverted and whose determinants need to be calculated. Furthermore, the entries of these matrices are usually very small (when returns are modelled) or very large (when prices are modelled). When having a large number of variables NA, calculating these determinants and inverses may be a demanding step.

When modelling returns, the determinants of Σ_{ij} are extremely small numbers, which appear in the denominator of the density function (2). This makes the value of the density function (2) very large and thus the value of the likelihood $P(O|\lambda)$ very large. At every step of the re-estimation, new parameters λ are obtained which further increase this probability; at some point, this value becomes too large and the re-estimation procedure has to stop after a (not enough sometimes) number of iterations. For a large number of variables NA, even a first step in the re-estimation procedure may be impossible.

To summarize, although theoretically the HMM-based time series modelling can tackle the multivariate case, from a practical point of view it has limited applicability.

3. The portfolio selection problem

In this section, we describe a financial application for which the HMM-based scenario generator described in Section 2 may be particularly suitable.

The portfolio selection (or asset allocation) problem can be shortly described as follows. At the present time there is an amount of capital to invest in a set of *NA* available assets. Decisions are required on the proportion of capital to be invested in each asset such that, after a specified period (the end of the investment period), the return is as high as possible. Each asset j in $\{1, \ldots, NA\}$ gives a return R_j at the end of the investment period. R_j is a random variable (since the future price of the asset is not known) whose distribution is approximated, via scenario generation, by a discrete distribution with a limited number of outcomes.

Let x_j be the proportion of capital invested in asset j ($x_j = w_j/w$, where w_j is the capital invested in asset j and w is the total amount of capital to be invested), and let $x = (x_1, \ldots, x_{NA})$ represent the portfolio resulting from this choice. This portfolio's return, at the end of the investment period, is a random variable: $R_x = x_1 R_1 + \cdots + x_{NA} R_{NA}$ that depends on the choice $x = (x_1, \ldots, x_{NA})$.

Thus, in order to take a decision $(x_1, \ldots, x_{NA})$, a criterion for choosing among random variables (representing returns) should be specified or, other said, a decision model should be chosen.

There are several decision models for the portfolio selection problem; perhaps the most popular of them are the mean-risk models. With mean-risk models, decisions are taken such that the expected value (mean) of the portfolio return is maximized while, at the same time, the 'risk' (as quantified by a specified risk measure) is minimized.

There are several established risk measures; some of them penalize deviations from the expected value (variance, mean absolute deviation) and some of them penalize only downside deviations from target (semivariance, lower partial moments). In recent years, because of regulatory restrictions, risk measures concerned with extremely unfavourable outcomes ('tail risk measures') have become increasingly popular. One such risk measure is the CVaR, which has attractive theoretical and computational properties (Rockafeller & Uryasev, 2000).

Loosely speaking, CVaR at a specified confidence level $\alpha \in (0, 1)$ measures the expected value of losses in the worst $A\% = \alpha$ of cases (e.g. if the portfolio return is a random variable with 500 equally possible outcomes, CVaR at confidence level 0.05 express the mean of the worst 25 outcomes).

The usual optimization approach in mean-risk models is to minimize the risk value, while imposing a constraint on the expected return.

The optimization problem that results in our case is

$$\begin{aligned} &\min && \mathrm{CVaR}_\alpha(R_x) \\ &\text{subject to:} && E(R_x) \geqslant d, \end{aligned} \tag{3}$$

where d is a prespecified target for the expected return.

In order to solve (3) numerically, the distribution of the random returns of the component assets has to be approximated by a discrete distribution via scenario generation. For this type of problem, it is particularly important that the scenario generator captures the extreme (downside) price movements of the component assets.

It was proven (Rockafeller & Uryasev, 2000, 2002) that, if the distributions of the assets returns are represented as discrete, with a finite number of outcomes, then (3) is a linear programming (LP). The

algebraic form of this LP is given below:

$$\begin{aligned}
&\text{Minimize } -\upsilon + \frac{1}{\alpha}\sum_{i=1}^{S} p_i y_i \\
&\text{Subject to} \\
&\quad \sum_{j=1}^{NA} -r_{ij}x_j + \upsilon \leqslant y_i, \quad \forall\, i = 1, \ldots, S, \qquad (4)\\
&\quad y_i \geqslant 0, \quad \forall\, i = 1, \ldots, S, \\
&\quad \sum_{j=1}^{NA} \mu_j x_j \geqslant d, \\
&\quad \sum_{j=1}^{NA} x_j = 1, \\
&\quad x_j \geqslant 0, \quad \forall\, j = 1, \ldots, NA,
\end{aligned}$$

where the parameters are

$(r_{ij})_{i=1,\ldots,S,\, j=1,\ldots,NA}$: the return of asset j under scenario i;

$(p_i)_{i=1,\ldots,S}$: the probability of scenario i (in our numerical experiments, we considered equal probabilities: $p_i = 1/S$);

$(\mu_j)_{j=1,\ldots,NA}$: the expected return of asset j $\left(\text{i.e. } \mu_j = \frac{1}{S}\sum_{i=1}^{S} r_{ij}\right)$;

and the variables are

$(x_j)_{j=1,\ldots,NA}$: the portfolio weights;

$(y_i)_{i=1,\ldots,S}$ and υ are auxiliary variables.

A more advanced model of choice for portfolio selection was proposed by Roman *et al.* (2007), in which both variance and CVaR are used as risk measures. However, since our study concerns the generation of scenarios, we use the simpler mean–CVaR decision model.

4. Numerical results

4.1 *Scope and data set*

In this section, we investigate the performance of the HMM-based scenario generator method described in Section 2.

The data set consists of historical monthly returns of five stocks (chosen randomly from FTSE 100), between January 1993 and December 2003 ($T = 132$ time periods of past observations).

Scenario joint returns are generated for the next time period, following the training data (January 2004). These scenarios are integrated in the mean–CVaR optimization problem (4) with CVaR at confidence level $0.01 = 1\%$ and the expected return target $d = 1\%$. On the basis of these generated returns, a decision $x = (x_1, \ldots, x_5)$ is taken at the 'current' time period (December 2003).

4.2 *The estimated parameters*

Using the AIC, we find evidence of a three-state model for all the five series. Thus, we consider an HMM with $N = 3$ states.

(This is consistent with Geyer & Ziemba (2008) which used three regimes in their pension fund model where the three regimes are driven by the volatility; they labelled the three regimes as 'normal', 'highly volatile' or 'extreme'/'crash' regimes.)

The number of mixtures that led to the minimal intra-variance (in a k-means clustering algorithm) is $M = 3$. Thus, we consider an HMM in which the output distribution of each state is a Gaussian mixture with $M = 3$ mixtures.

The rest of the parameters are estimated as described in Section 2.

The initial parameters used in the 're-estimation' procedure are

- equal initial state probabilities: $\pi_i = 1/3$, $i = 1, \ldots, 3$;
- equal values for the transition probabilities $a_{ij} = 1/3$, $i, j = 1, \ldots, 3$;
- equal values for the mixture coefficients $c_{ij} = 1/3$, $i, j = 1, \ldots, 3$;
- the means and covariances of the corresponding Gaussians are obtained via a k-means clustering algorithm with $M = 3$ clusters (each state has the same initial output distribution).

The estimated parameters of the HMM are written below.

The initial probabilities:

$$\pi = (0.989186 \quad 0.00344584 \quad 0.00736835).$$

The transition probabilities:

$$A = \begin{pmatrix} 0.78 & 0.19 & 0.03 \\ 0.61 & 0.30 & 0.09 \\ 0.87 & 0.11 & 0.02 \end{pmatrix}.$$

The mixture coefficients:

$$C = \begin{pmatrix} 0.43 & 0.25 & 0.32 \\ 0.40 & 0.34 & 0.26 \\ 0.42 & 0.15 & 0.43 \end{pmatrix}.$$

The means of the Gaussians mixtures (for each $i = 1, \ldots, 3$, $\Gamma[i]$ is a $(3, 5)$ matrix in which the jth line, $j = 1, \ldots, 3$, is the mean vector of the of the jth mixture corresponding to the ith state):

$$\Gamma[1] = \begin{pmatrix} 0.020 & -0.025 & -0.007 & -0.006 & -0.008 \\ -0.001 & 0.003 & 0.083 & 0.077 & 0.004 \\ -0.005 & 0.127 & 0.017 & 0.026 & 0.022 \end{pmatrix},$$

$$\Gamma[2] = \begin{pmatrix} -0.016 & -0.155 & -0.103 & -0.095 & 0.036 \\ 0.006 & 0.026 & 0.149 & 0.098 & -0.010 \\ 0.056 & 0.087 & -0.001 & -0.026 & 0.024 \end{pmatrix},$$

$$\Gamma[3] = \begin{pmatrix} -0.079 & -0.040 & -0.139 & -0.075 & 0.083 \\ 0.012 & 0.011 & 0.028 & 0.107 & 0.017 \\ 0.173 & 0.329 & 0.042 & 0.229 & 0.029 \end{pmatrix}.$$

The covariances of the Gaussians mixtures (for each $i = 1, \ldots, 3$, $j = 1, \ldots, 3$, Σ_{ij} is a $(5, 5)$ matrix: the covariance of the jth mixture in the ith state):

$$\Sigma_{11} = \begin{pmatrix} 0.0042 & 0.0016 & 0.0013 & -0.0012 & 0.0013 \\ 0.0016 & 0.0056 & 0.0015 & 0.0003 & 0.0009 \\ 0.0013 & 0.0015 & 0.0062 & -0.0006 & 0.0027 \\ -0.0012 & 0.0003 & -0.0006 & 0.0068 & -0.0023 \\ 0.0013 & 0.0009 & 0.0027 & -0.0023 & 0.0046 \end{pmatrix},$$

$$\Sigma_{12} = \begin{pmatrix} 0.0015 & -0.0011 & 0.0003 & 0 & 0.0002 \\ -0.0011 & 0.0106 & 0.0036 & 0.0024 & 0 \\ 0.0003 & 0.0036 & 0.0096 & 0.0012 & -0.0003 \\ 0 & 0.0024 & 0.0012 & 0.0057 & 0.0017 \\ 0.0002 & 0 & -0.0003 & 0.0017 & 0.0021 \end{pmatrix},$$

$$\Sigma_{13} = \begin{pmatrix} 0.0036 & 0.0003 & -0.0004 & -0.0005 & 0.0018 \\ 0.0003 & 0.0065 & 0.0022 & 0.0015 & 0.0003 \\ -0.0004 & 0.0022 & 0.0059 & 0.0005 & -0.0011 \\ -0.0005 & 0.0015 & 0.0005 & 0.0045 & 0.0003 \\ 0.0018 & 0.0003 & -0.0011 & 0.0003 & 0.0026 \end{pmatrix},$$

$$\Sigma_{21} = \begin{pmatrix} 0.0117 & 0.0109 & 0.0054 & 0.0049 & 0.0048 \\ 0.0109 & 0.0249 & 0.0048 & 0.0070 & -0.0011 \\ 0.0054 & 0.0048 & 0.0179 & 0.0069 & 0.0048 \\ 0.0049 & 0.0070 & 0.0069 & 0.0058 & 0.0027 \\ 0.0048 & -0.0011 & 0.0048 & 0.0027 & 0.0125 \end{pmatrix},$$

$$\Sigma_{22} = \begin{pmatrix} 0.0032 & 0.0029 & -0.0035 & 0.0006 & -0.0023 \\ 0.0029 & 0.0205 & 0.0100 & 0.0082 & -0.0051 \\ -0.0035 & 0.0100 & 0.0313 & 0.0101 & 0.0041 \\ 0.0006 & 0.0082 & 0.0101 & 0.0255 & -0.0038 \\ -0.0023 & -0.0051 & 0.0041 & -0.0038 & 0.0054 \end{pmatrix},$$

$$\Sigma_{23} = \begin{pmatrix} 0.0104 & 0.0027 & 0.0016 & -0.0016 & 0.0025 \\ 0.0027 & 0.0112 & 0.0024 & 0.0022 & -0.0014 \\ 0.0016 & 0.0024 & 0.0117 & 0.0020 & -0.0038 \\ -0.0016 & 0.0022 & 0.0020 & 0.0045 & -0.0027 \\ 0.0025 & -0.0014 & -0.0038 & -0.0027 & 0.0089 \end{pmatrix},$$

$$\Sigma_{31} = \begin{pmatrix} 0.0107 & -0.0016 & 0.0077 & 0.0021 & -0.0018 \\ -0.0016 & 0.0265 & 0.0014 & 0.0064 & 0.0094 \\ 0.0077 & 0.0014 & 0.0121 & 0.0038 & 0.0006 \\ 0.0021 & 0.0064 & 0.0038 & 0.0038 & 0.0024 \\ -0.0018 & 0.0094 & 0.0006 & 0.0024 & 0.0108 \end{pmatrix},$$

$$\Sigma_{32} = \begin{pmatrix} 0.0040 & 0.0086 & -0.0027 & 0.0079 & 0.0005 \\ 0.0086 & 0.0346 & -0.0042 & 0.0277 & -0.0023 \\ -0.0027 & -0.0042 & 0.0180 & -0.0048 & -0.0015 \\ 0.0079 & 0.0277 & -0.0048 & 0.0447 & -0.0020 \\ 0.0005 & -0.0023 & -0.0015 & -0.0020 & 0.0055 \end{pmatrix},$$

$$\Sigma_{33} = \begin{pmatrix} 0.0203 & 0.0280 & 0.0195 & 0.0001 & 0.0004 \\ 0.0280 & 0.0440 & 0.0234 & 0.0115 & 0.0007 \\ 0.0195 & 0.0234 & 0.0266 & -0.0165 & -0.0004 \\ 0.0001 & 0.0115 & -0.0165 & 0.0457 & 0.0011 \\ 0.0004 & 0.0007 & -0.0004 & 0.0011 & 0.0012 \end{pmatrix}.$$

It may be noticed that State 1, which is the most likely to occur, does not generate extreme returns. States 2 and 3 generate extreme observations, each in a different way.

By applying the Viterbi algorithm, we obtain the 'most likely' sequence of states in the 132 time periods of our data set. As expected, State 1 occurs most of the times; at the opposite, State 3 occurs only four times.

The estimated state of the system at the 'current' time period (i.e. the last historical time period) is State 1.

4.3 *Stability tests*

Stability is a basic condition required when assessing the quality of a scenario generator.

A scenario generator is said to manifest 'in-sample stability' if, when generating several scenario sets of the same size and solving the optimization problem on each of these scenario sets, the optimal objectives are similar[4] (see, e.g. Kaut & Wallace, 2007).

A scenario generator is said to manifest 'out-of-sample stability' if, when generating several scenario sets of the same size and solving the optimization model on each of these scenario sets, the optimal solutions obtained yield similar 'true' objective function values (i.e. the solutions obtained are evaluated on the 'true' distribution of the random variables involved).

While it is straightforward to test in-sample stability (we only solve the scenario-based optimization problems), it is difficult to test the out-of-sample stability. The 'true' distributions must be known and even then it may be not straightforward to evaluate the objective function. What is usually done in practice is to use a 'benchmark' scenario tree: a very large scenario set obtained with an exogenous scenario generation method that is known to be stable. This scenario set will stand for the 'true' distribution. (Of course, if the optimal solutions are similar, then there is an out-of-sample stability, since these solutions are evaluated on the same scenario set. However, as stated before, this may be a too 'strong' condition required–we could have out-of-sample stability even if the optimal solutions are not similar.)

Thus, when testing the in-sample stability, the same scenario set is used for both making a decision and evaluating it. When testing the out-of-sample stability, a scenario set is used for making a decision and another scenario set (the 'benchmark') is used for evaluating it.

The out-of-sample stability is the important one, since this says that the real performance of the solution is stable, i.e. it does not depend on which scenario set we solved the optimization problem.

[4] Although it is good if optimal solutions are also similar, this is not a necessary condition, since there could be very different solutions leading to the same or similar values of the objective functions.

The problem is how to test it. In-sample stability does not imply the out-of sample one or vice versa. It is possible to have in-sample instability (of the objectives) but stability of the solutions—in this case, it is likely to have out-of-sample stability, since, in this case, all the solutions are tested on the same scenario set.

We test the in-sample stability of the scenario generator based on HMM in the mean–CVaR model using different number of scenarios: 500, 700, 1000, 2000 and 3000 scenarios. For every size, we generate 30 scenario sets, solve the mean–CVaR optimization model (3) on each of them, thus obtaining 30 different optimal objective functions (which are portfolios CVaR) and 30 different optimal solutions (the portfolio weights).

The results regarding the in-sample stability are presented in Table 1. For each scenario size, we obtain a set of 30 optimal objective values; their statistics are displayed below. As expected, the more scenarios, the more 'bad' possibilities are taken into account and thus the values of CVaR (i.e. mean losses) get higher. The standard deviation and the range should normally get smaller with an increase in the number of scenarios. However, it is not always the case in our tests. For 3000 scenarios, there is a quite big difference in the optimal objectives obtained: from 12 to 22%. We cannot conclude that the results concerning the in-sample stability look very promising.

However, the optimal solutions obtained are quite similar, especially for large number of scenarios: nothing invested in Assets 2 and 3 and the biggest proportion (in almost all cases, more than 50%) invested in Asset 5.

This 'stability of optimal solutions' encourages to test the out-of-sample stability.

For testing the out-of-sample stability, we use a benchmark scenario set of 30 000 scenarios, obtained by using geometric Brownian motion—the standard in finance for generating asset prices.

Table 2 presents the results regarding the out-of-sample stability.

It can be seen that the quality of the solutions improves while the scenario size increases: better (i.e. smaller) CVaRs and lower standard deviation, lower range. The sets of 'true' objective functions have a reasonably small spread, especially for a large number of scenarios. We can conclude that the scenario generator based on HMM gives a reasonable out-of-sample stability.

As a final remark, it can be noted the low CVaR values obtained when considering scenarios generated via geometric Brownian motion (Table 2). We considered this method for generating the benchmark scenario set, due to its large use in financial practice. However, it is clear that it underestimates the 'seriousness' of losses in worst case scenarios. This underlines one more time the importance of appropriate scenario generators for asset prices when considering 'lower tail' risk.

TABLE 1 *In-sample stability test: statistics of the sets of optimal solutions, for various sizes of the scenario set*

	Scenarios				
	500	700	1000	2000	3000
Mean	0.1389	0.1414	0.1455	0.1439	0.1559
Standard deviation	0.0275	0.0273	0.0251	0.0184	0.0246
Range	0.1085	0.1259	0.1209	0.0824	0.1001
Minimum	0.0968	0.0958	0.0854	0.0910	0.1238
Maximum	0.2054	0.2217	0.2062	0.1734	0.2239

TABLE 2 *Statistics of the sets of 'true' objective functions. For every scenario size, we solve the mean–CVaR model* 30 *times. Each optimal solution is evaluated on the benchmark scenario set, thus obtaining* 30 *true objectives for each scenario size*

	Scenarios				
	500	700	1000	2000	3000
Mean	0.0035	0.0033	0.0031	0.0029	0.0026
Standard deviation	0.0012	0.0012	0.0010	0.0006	0.0003
Range	0.0051	0.0042	0.0048	0.0022	0.0011
Minimum	0.0024	0.0023	0.0023	0.0023	0.0023
Maximum	0.0074	0.0065	0.0071	0.0045	0.0034

5. Summary and conclusions

We have considered generating scenarios for multivariate financial time series by using HMMs. Capturing extreme movements of such time series is important in financial decision-making for taking hedged decisions against downside risk. These extreme movements are captured in the proposed scenario generation method via a regime-switching-type model in which (at least) one of the states generates 'extreme' observations.

With HMM, a time series is modelled by considering another stochastic process that describes the system's state and that cannot be directly observed (hidden). Different states of the system generate observations (for the time series of interest) according to probability distributions with different parameters. In our approach, observations are generated in each state according to mixtures of normal distributions. This choice is motivated by the fact that a correctly chosen mixture of Gaussians can approximate any continuous density function.

The estimation of the parameters of an HMM is a difficult step and the implementation problems in the multivariate case can be quite big. The most important problem is that the likelihood values, used in parameter estimation, may be very large, sometimes to the extent that it makes the estimation procedure impossible. Thus, although theoretically the multivariate case can be handled, in practice the parameter estimation can be done only for a limited number of time series.

In the numerical experiments, we generated scenarios for the joint returns of five stocks chosen from FTSE 100. The values of the estimated parameters confirm the idea that extreme observations may be properly generated in an HMM by considering (at least) a state, occurring with low probability, that generates such observations. The resulting scenarios were integrated in a mean–CVaR optimization model used for portfolio selection.

The tests showed a reasonable out-of-sample stability of the scenario generator. The optimal solutions (obtained using the HMM-based scenario generator) were evaluated on a large scenario set generated via geometric Brownian motion. The results confirmed one more time the need for appropriate generation of extreme events: traditional methods for modelling asset prices (like geometric Brownian motion) led to an underestimation of the downside risk associated with financial decisions.

Funding

Optirisk Systems; Brunel University Research Innovation and Enterprise Fund.

REFERENCES

AKAIKE, H. (1974) A new look at the statistical model identification. *IEEE Trans. Autom. Control*, **AC19**, 716–723.

AZZOUZI, M. & NABNEY, I. T. (1999) Modelling financial time series with switching state space models. *Proceedings of the IEEE/IAFE 1999 Conference on Computational Intelligence for Financial Engineering*, pp. 240–249. Available at http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=771123&isnumber=16641. doi:10.1109/CIFER.1999.771123.

BAUM, L. E., PETRIE, T., SOULES, G. & WEISS, N. (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *Ann. Math. Stat.*, **41**, 164–171.

BENGIO, Y., LAUZON, V. P. & DUCHARME, R. (2001) Experiments on the application of IOHMMs to model financial returns series. *IEEE Trans. Neural Netw.*, **12**, 113–123.

BOLLERSLEV, T. (1986) Generalized autoregressive conditional heteroskedasticity. *J. Econ.*, **31**, 307–327.

CETTOLO, M. (2004) *An Introduction to Hidden Markov Models*. Available at http://hermes.itc.it/people/cettolo/papers/talk-partch.pdf.

DEMPSTER, P., LAIRD, N. M. & RUBIN, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc.*, **39**, 1–38.

ENGLE, R. F. (1982) Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. *Econometrica*, **50**, 987–1007.

EPHRAIM, Y. & MERHAV, N. (2002) Hidden Markov processes. *IEEE Trans. Inf. Theory*, **48**, 1518–1569.

FISHBURN, P. C. (1977) Mean-risk analysis with risk associated with below-target returns. *Am. Econ. Rev.*, **67**, 116–126.

FORNEY, G. D. (1973) The Viterbi algorithm. *Proc. IEEE*, **61**, 268–278.

GEYER, A. & ZIEMBA, W. T. (2008) The Innovest Austrian pension fund financial planning model InnoALM. *Oper. Res.*, **56**, 797–810.

HAMILTON, J. D. & SUSMEL, R. (1994) Autoregressive conditional heteroskedasticity and changes in regime. *J. Econom.*, **64**, 307–333.

HARDY, M. R. (2001) A regime-switching model of long-term stock returns. *North Am. Actuar. J.*, **5**, 41–53.

KAUT, M. & WALLACE, S. W. (2007) Evaluation of scenario-generation methods for stochastic programming. *Pac. J. Optim.*, **3**, 257–271.

LEE, Y., LEE, K. Y. & LEE, J. (2006) Estimating optimal number of Gaussian mixtures based on incremental *k*-means for speaker identification. *Int. J. Inf. Technol.*, **12**, 13–21.

MACQUEEN, B. (1967) Some methods for classification and analysis of multivariate observations. *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability* (L. M. Le Cam & J. Neyman eds), vol. 1. Berkeley, CA: University of California Press, pp. 281–297.

MANSINI, R., OGRYCZAK, W. & SPERANZA, M. G. (2007) Conditional value at risk and related linear programming models for portfolio optimization. *Ann. Oper. Res.*, **152**, 227–256.

MESSINA, E. & TOSCANI, D. (2008) Hidden Markov models for scenario generation. *IMA J. Manage. Math.*, **19**, 379–401.

PSARADAKIS, Z. & SPAGNOLO, N. (2003) On the determination of the number of regimes in Markov-switching autoregressive models. *J. Time Ser. Anal.*, **24**, 237–252.

RABINER, L. R. (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**, 257–286.

ROCKAFELLER, R. T. & URYASEV, S. (2000) Optimization of conditional value-at-risk. *J. Risk*, **2**, 21–42.

ROCKAFELLER, R. T. & URYASEV, S. (2002) Conditional value-at-risk for general loss distributions. *J. Bank. Financ.*, **26**, 1443–1471.

ROMAN, D., DARBY-DOWMAN, K. & MITRA, G. (2007) Mean-risk models using two risk measures: a multi-objective approach. *Quant. Financ.*, **7**, 443–458.

SHI, S. & WEIGEND, A. S. (1997) Taking time seriously: hidden Markov experts applied to financial engineering. *Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering*, pp. 244–252. Available at http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=618944&isnumber=13472. doi:10.1109/CIFER.1997.618944.

ZHANG, Y. (2004) Prediction of financial time series with hidden Markov models. *M.Sc. Thesis*, Simon Fraser University, China.