Optimisation of Stochastic Programming by Hidden Markov Modelling based Scenario Generation

by
Sovan Mitra

Abstract

This paper formed part of a preliminary research report for a risk consultancy and academic research. Stochastic Programming models provide a powerful paradigm for decision making under uncertainty. In these models the uncertainties are represented by a discrete scenario tree and the quality of the solutions obtained is governed by the quality of the scenarios generated. We propose a new technique to generate scenarios based on Gaussian Mixture Hidden Markov Modelling. We show that our approach explicitly captures important time varying dynamics of stochastic processes (such as autoregression and jumps) as well as non-Gaussian distribution characteristics (such as skewness and kurtosis). Our scenario generation method enables richer robustness and scenario analysis through exploiting the tractable properties of Markov models and Gaussian mixture distributions. We demonstrate the benefits of our scenario generation method by conducting numerical experiments on FTSE-100 data.

Key words: Markov Processes, Risk Analysis, Stochastic Programming, Scenarios.

1. Introduction

Businesses are constantly confronted with the position of making decisions under uncertainty e.g. portfolio investment, power production and even agricultural planning [BL97]. Therefore the ability to be able to quantify and optimise such decisions is of paramount value and the modelling paradigm of Stochastic Programming offers such a possibility. Stochastic Programming (SP) has been proven to provide practical and optimal solutions e.g. Mulvey's celebrated Towers Perrin-Tillinghast model saved US West \$450-\$1000 million [MGM00].

Stochastic Programming concerns decision making under uncertainty. A multistage stochastic programming problem is defined as:

$$\min_{x\in\Xi} \mathbb{E}_P\{f(x,\omega)\} = \min_{x\in\Xi} \int_\Omega f(x,\omega)dP(\omega), \tag{1}$$

where

- x=$\{x_t\}$ is the set of decisions at each stage t=1,2....T where T is the total number of stages;
- Ξ is the set of feasible decisions;
- ω is the stochastic process;
- Ω is the set of all possible events;
- P is the probability measure for stochastic process ω;
- $\mathbb{E}_P$ is the expectation taken with respect to probability measure P;
- $f(x,\omega)$ is the cost function.

In stochastic programming a method is required to discretise the random process ω into a set of discrete outcomes, known as scenarios. The method of obtaining scenarios is known as scenario generation.

The root node of the scenario tree represents 'today' and is known with certainty. Nodes or scenarios at later stages progressing from the root node represent possible realisations of the random process, where each scenario has a probability and a value. As Birge and Louveaux [BL97] point out, scenarios are particularly useful when the optimal solution varies considerably with changes in value of the stochastic variables. If this were not the case, the solution obtained for one stochastic variable realisation ought to be a good solution for most realisations.

The general formulation of equation (1) under scenario generation is:

$$\begin{aligned}\min_{x\in\Xi} \mathbb{E}_P\{f(x,\omega)\} &= \min_{x\in\Xi} \sum_{i=1}^{i=N_1} p_1^i f(x_1^i,\omega_1^i) + \sum_{i=1}^{i=N_2} p_2^i f(x_2^i,\omega_2^i) + ...\\ &+ \sum_{i=1}^{i=N_T} p_T^i f(x_T^i,\omega_T^i),\end{aligned}$$

where

- i=*1,2....*N_t is the scenario and N_t is the total number of scenarios at stage t;
- p_t^i is probability of scenario i at stage t;
- $f(x_t^i, \omega_t^i)$ is the cost function of scenario i at stage t.

A key issue in scenario generation that tends not to be addressed is the ability to capture time varying dynamics of the random process ω (such as jumps and reversions) and to a lesser extent, capturing non-Gaussian characteristics. Our scenario generation method specifically addresses the aforementioned problems through the properties of Gaussian Mixtures (GM) and Hidden Markov Models (HMM).

In GM HMM we model a time series as a HMM where each hidden Markov state is represented by a univariate mixture of Gaussians. To produce a scenario tree we let each Markov state correspond to a stage in the scenario tree. Then we obtain scenarios at each scenario tree stage by sampling from each state's associated univariate mixture of Gaussians. Hence the key part in our scenario generation procedure is determining the GM HMM and then selecting a state sequence for our scenario tree. Once the states have been selected, we simply sample from each state's GM to obtain scenarios at each stage.

Guided Tour
The outline of this paper is as follows: firstly we review current scenario generation methods and then introduce the GM HMM model. We then explain the advantages of the GM HMM for scenario generation, showing how the particular features of GM and HMM capture the time varying dynamics of data. We also show how GM HMM provides SP modelling benefits e.g. robustness analysis. Finally, we illustrate the GM HMM scenario generation and modelling benefits by applying it to the FTSE-100 index, giving computational results.

2. Current Scenario Generation Methods

Currently there exist four main categories for scenario generation [DDBMV]: sampling, simulation, statistical property matching and hybrid methods e.g. simulation-optimisation [AGKL03]. For a detailed list of scenario generation methods see [DDBMV],[Mit06].

In simulation, a stochastic process's simulation results in a "path by path" production of the scenario tree [KW03]. To reduce the number of paths sometimes paths are bounded together by some grouping method, for instance [GRS04]. Popular stochastic processes include geometric Brownian motion and its variants.

In scenario generation by sampling we sample values from a distribution and the sampled value represents the scenario's value. Various methods of sampling exist [AGKL03], giving rise to different sampling scenario generation methods. Popular methods include: random sampling, importance sampling [DT99], bootstrap sampling, internal sampling [KW03],conditional sampling and stratified sampling.

In scenario generation by statistical property matching we do not require knowledge of the random variable's (ω) pdf (probability density function). Instead we require the statistical properties of the generated scenario tree to match the statistical properties of some target (as closely as possible) e.g. percentiles. Hoyland and Wallace in [HW01] describe a scenario generation method using moment matching.

3. Gaussian Mixture Hidden Markov Model Scenario Generation

3.1. Markov Models

Markov Models (MM) and Hidden Markov Models (HMM) are methods of mathematically modelling time varying dynamics of some statistical process. They only require a weak set of assumptions yet provide powerful results. MM and HMM model a stochastic process (or any system) as a set of states with each state possessing a set of signals or observations. Changes in the stochastic process or system are modelled by movements between states. The models have been used in diverse applications such as Finance [Har01], queuing theory [SF06], Engineering [TG01] and biological modelling [MGPG06].

A Markov model (or more specifically a Markov chain) is a stochastic process X_t with a countable set of states and possesses the Markov property. That is, given X_t is in state i_t (denoted $X_t = i_t$), X_{t+1} is not influenced by the set of values X_u for $u < t$. Formally, the Markov property is

$$P(X_{t+1} = j \mid X_0 = i_0, X_1 = i_1, ..., X_t = i_t) = P(X_{t+1} = j \mid X_t = i_t),$$

where i_t denotes a state at time t. As time passes the process may remain or change to another state (known as state transition). The state transition probability matrix (also known as the *transition kernel*) tells us these probabilities $P_{ij} = P(X_{n+1} = j \mid X_n = i)$ where P_{ij} is the probability of the process changing to state j given that we are now in state i. Note that we assume all probabilities are stationary in time.

3.2. Gaussian Mixture Hidden Markov Models

In Markov models we directly observe the state associated with each signal or observation; whereas in Hidden Markov models the states are directly unidentifiable or unobservable. However in HMM we assume that each signal or observation is a probabilistic function of each hidden state, hence we can statistically infer the HMM states from observations. If observations were not statistical functions of state then we could not determine the hidden states. The observation probabilities are represented in the observation matrix B, showing each observation probability for each state. Note that as in plain Markov models we assume all probabilities are stationary.

A hidden Markov model is fully defined when we know [Rab89]:

- N is the total number of hidden states $S_1, S_2..., S_N$;
- A represents the hidden state transition matrix of size NxN. Each element is $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ where q_t is the hidden state at time t;
- π_i (=$P(q_0 = S_i)$) for $1 \leq i \leq N$ denotes the the probability of being in state S_i at time *t=0*;
- C denotes the number of distinct observations. The set of observations are represented as $v_1, v_2 \ldots, v_C$. For example if we model weather as a HMM with only 2 observation signals hot and cold, then $C = 2$ and v_1=hot, v_2=cold;
- B denotes the observation matrix. Each entry is $b_j(c) = P(v_c | q_t = S_j)$, $1 \leq c \leq C$. So $b_j(c)$ is the probability of observation v_c at time t, given that we are in state S_j at time t.

A Gaussian mixture is a pdf composed of a weighted sum of univariate Gaussians, where all the weights sum to one. Let $\xi(\omega)$ be a random variable $\xi(\omega) \in \mathbb{R}$ whose pdf $f(\xi)$ is represented by m Gaussian mixtures [EH81]:

$$f(\xi) = w_1 N(\xi, \mu_1, \sigma_1) + w_2 N(\xi, \mu_2, \sigma_2) + ... + w_m N(\xi, \mu_m, \sigma_m), \quad (2)$$

where μ_m, σ_m, w_m denotes the mth Gaussian's mean, standard deviation and weighting respectively. Note that the mean and all other moments are weighted sums too [EH81], for instance the overall distribution's mean μ_f is:

$$\mu_f = \sum_{k=1}^{m} w_k \mu_k. \quad (3)$$

Now a Gaussian mixture HMM is simply a HMM where each state's observations is represented by a Gaussian mixture pdf. In other words $f_j(\xi)$ is our observation signal $b_j(c)$ where [Lev05]:

$$f_j(\xi) = \sum_{k=1}^{m} w_{jk} N(\xi, \mu_{jk}, \sigma_{jk}), \xi \in \mathbb{R}, \quad (4)$$

where

- $f_j(\xi)$ is the pdf of ξ in state j where $1 \leq j \leq N$;
- w_{jk} is the mixture weighting in state j for Gaussian mixture k where $1 \leq k \leq m$, $w_{jk} \geq 0 \ \forall j,k$ and $\sum_{k=1}^{m} w_{jk}$=1 $\forall j$;
- μ_{jk} is the mean for each mixture k, in state j;
- σ_{jk} is the standard deviation for each Gaussian k in state j;
- m is the total number of Gaussians in the mixture.

3.3. GM HMM Scenario Generation Method

Given we are applying GM HMM to financial data we may expect the stock market to belong to 3 economic phases (growth, recession and a transitional state) so we set N=3. Additionally, we choose m=2 given that most empirical financial return distributions can be described by a mixture of 2 Gaussians. We should also note by the Akaike information criterion that it is better to minimise the number of variables to prevent overfitting.

Scenario Generation Algorithm

Let $\hat{r}$ where $\hat{r} \sim U[0,1]$ denote a random number drawn from a unit interval.

1. Randomly select an initial state:
   Draw a random number $\hat{r}$.
   If:
   - $\hat{r} < \pi_1$ set j=1;
   - $\pi_1 < \hat{r} < \pi_1 + \pi_2$ set j=2;
   - otherwise set j=3.
2. Randomly select a Gaussian component k ([EH81],[BDM71]):
   Draw a random number $\hat{r}$.
   Set k=1 if $\hat{r} < w_{j1}$, otherwise k=2.
3. Sample Gaussian $N(\mu_{jk}, \sigma_{jk})$.
4. Randomly select the next state:
   Draw a random number $\hat{r}$.
   If:
   - $\hat{r} < a_{j1}$ set j'=1;
   - $a_{j1} < \hat{r} < a_{j1} + a_{j2}$ set j'=2;

- otherwise set j'=3.

  Set current state to new state: $j \mapsto j'$.

5. Increment I.

   If I equals the maximum number of scenarios required then stop, otherwise goto step 2.

To calibrate the GM HMM we apply the Baum-Welch algorithm, which is an expectation-maximisation algorithm. The algorithm determines the GM HMM model parameters such that it maximises the statistical likelihood of a given observation data. The algorithm is guaranteed to converge to some local optimum for calculating the likelihood of a given observation sequence. The reader is referred to [Rab89],[JR91],[JR85] for more information.

If we define λ={A, B, π_i} to denote the set of HMM parameters we wish to estimate then the pseudo code for the Baum-Welch algorithm is:

1. Initialisation:

   Input initial values of λ (otherwise randomly initialise) and calculate $P(O|\lambda)$ using the forward algorithm.

2. Estimate new values of λ.

   Iterate until convergence:

   (a) Using current λ calculate $\gamma_t(i,j) = P(q_t = S_i, q_{t+1} = S_j|O, \lambda)$ for each t,i,j where

   $$\gamma_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}. \quad (5)$$

   Variables $\alpha_t(i), \beta_t(j)$ are known as the forward and backward variables respectively (see [Rab89]):

   $$\alpha_t(i) = P(O_1 O_2 .....O_t, q_t = S_i|\lambda), \quad (6)$$
   $$\beta_t(j) = P(O_{t+1} O_{t+2} .....O_T|q_t = S_j, \lambda). \quad (7)$$

   (b) Calculate new λ parameter estimates using $\gamma_t(i,j)$ (see [Rab89]).

   (c) Calculate $P(O|\lambda)$ with new λ values using the forward algorithm.

3. Goto step 4 if two consecutive calculations of $P(O|\lambda)$ are equal (or converge within a specified range). Otherwise repeat iterations: goto step 2.

4. Output λ.

Explicit Modelling of Reversionary and Jump Dynamics

Stochastic processes or time series tend to exhibit reversionary characteristics (e.g. autoregression) and jumps. These dynamics make continuous time modelling challenging and are typically ignored in scenario generation since most methods are unable to accommodate these dynamics. Using HMM we can explicitly capture reversionary characteristics and jumps; in fact HMM have been used widely in Finance to capture such aspects [Reb04],[Har01],[Ham89]. Important reversion properties captured are:

- autoregression: this is captured by the transition probabilities of returning to the same current Markov state $P(q_{t+1} = S_i | q_t = S_i)$ ([Ham89],[Ham]).
- conditional heteroscedasticity:since each Markov state has a different variance and each future state's transition depends on the current state, variance changing is conditional on the past.
- cyclical behaviour [Ham89]: the cyclical nature of data is captured through the transition probability matrix A, so that the MM cycles through its entire set of states.

Jumps are explicitly modelled by the transition probability of switching between 2 states with significantly different distributions [Reb04].

Modelling Diverse Distributional Characteristics

Gaussian mixtures can capture virtually any empirical distribution. In fact GM have been used to accurately capture outliers and "noisiness" in empirical data [TSM85]. Various distribution characteristics are captured by simply adjusting the mixture component means, variances and weighting.

Tractable Statistical Calculations

Gaussian distributions tend to be applied to data modelling due to their analytical tractability yet most (empirical) distributions tend to be non-Gaussian e.g. asymmetric and fat tailed. Additionally non-Gaussian models tend to possess intractable statistical calculations e.g. the Levy distribution.

Using a mixture of Gaussians we can model most non-Gaussian distributional characteristics (e.g. skewness, kurtosis etc....) without losing tractability. For example, the cumulative distribution function for a GM mixture consisting of 0 mean components is simply a weighted sum of each component's cumulative distribution value. Thus the

the overall cumulative distribution function for any univariate GM $F(\xi)$ gives [BS07]:

$$F(\xi) = \sum_{k=1}^{m} w_k \Phi\left(\frac{\xi - \mu_k}{\sigma_k}\right), \tag{8}$$

where $\Phi(z)$ is the *standard normal* cumulative distribution function for random variable z.

Copula Approximation

Copulas map marginal distributions to 1 multivariate distribution and so are particularly useful in risk analysis. However copulas can become analytically intractable, causing calculation and computation difficulties e.g. sampling and scenario generation [HKW03]. Furthermore, there exist significant implementation problems since we would need to 1)estimate the covariance matrix, which tends to be unstable over time 2)specify the copula type (e.g. median, Gaussian) 3)determine a feasible method of scenario generation from the overall multivariate distribution. All these tasks are non-trivial.

Mixture of Gaussians can be used as an alternative to copulas [BS07], where all the component distributions are mapped into one univariate Gaussian mixture. Using univariate GM the problematic steps of 1-2 are removed, rather only the weighting for each component distribution need be estimated.

If we map to a univariate (rather than a multivariate) distribution then the distribution model is more amenable to analytical calculations, computational methods and scenario generation. For instance GM can be applied to models of geometric Brownian motion [BMS03] whereas for multivariate Gaussians this becomes non-trivial. Univariate GM can be applied to a wider and more efficient range of scenario generation and computational methods.

Robustness and Scenario Analysis

In SP we may wish to evaluate a SP model's robustness by stress testing and shock analyses [BL97]. Dupacova suggests a contamination method for stress testing [Dup96] where randomness is added to the scenarios. For shock analysis we typically expose the SP model to an extreme event or a *worst case scenario*.

Using GM we can enrich stress and shock analysis, in fact GM already have been used as robustness methods in statistical modelling [TSM85] but not in SP. For stress testing, we can add a GM of any choice to the overall distribution, representing a contamination as in Dupacova's method. Furthermore, we can calculate the impact on the overall distribution's moments and pdf for given weights, means and variances.

For shock analyses, rather than simply testing with respect to an extreme event scenario we can model a shock by a GM of chosen weighting, mean and variance and this is more realistic. For instance an overall GM distribution may represent a company's profit and we may wish to determine the impact of some extreme event e.g. hurricanes can severely disrupt natural gas supplies. We could more realistically model a hurricane event by a GM compared to testing with respect to one extreme event scenario.

Adaptability to Multivariate Distributions

The GM HMM can model multivariate Gaussian mixtures (see [Lev05] for a detailed application in Engineering) and there exist significant statistical literature, algorithms and programs available. Additionally, multivariate GM HMM capture time varying covariance matrices by specifying one for each hidden Markov state [Rab89]. This is a significant advantage over other multivariate modelling methods particularly for financial applications, where correlations are critical to portfolio diversification.

4. Numerical Experiment: Case Study Application to the FTSE-100

We present results of applying the GM HMM to modelling the monthly returns of the FTSE-100 r_t, where r_t is defined by [Hul00]:

$$r_t = ln\frac{I_t}{I_{t-1}}. \tag{9}$$

The variable I_t is defined by the FTSE-100 index value in month t. We used monthly data from April 1984-February 2007; the GM HMM was calibrated by the Baum-Welch algorithm [Rab89] using an implementation by Murphy [Mur08]. The results are now presented.

Table 1: Transition Matrix A Results

Current State	Next State		
	State 1	State 2	State 3
State 1	0.71	0.057	0.23
State 2	0.13	0.51	0.36
State 3	0.45	0.19	0.36

Table 2: Mixture Means μ_{jk} (monthly % return)

Component Distribution (k)	State 1	State 2	State 3
1	1.662	-2.588	0.776
2	-30.170	1.113	-0.680
Overall	1.41	0.097	0.44

Table 3: Standard Deviations σ_{jk} (monthly % return)

Component Distribution (k)	State 1	State 2	State 3
Gaussian 1	1.201	2.940	2.447
Gaussian 2	0.100	2.728	2.715
Overall	1.192	2.88	2.51

Table 4: Initial State Probabilities (π_i)

State	Probability
1	1.0434e-024
2	0.99
3	1.6637e-017

Table 5: Weighting Matrix (w_{jk})

State	Component k=1	Component k=2
1	0.99207	0.0079285
2	0.73889	0.26111
3	0.7665	0.2335

Table 6: Stress Test Example Scenario Tree

Scenario Number	Unstressed	Stressed
1	-0.2521	-10.1146
2	-0.0686	-10.0433
3	0.0572	-0.3384
4	0.8312	1.6168
5	2.0156	1.8125
6	2.3479	1.8717
7	2.5199	2.0075
8	2.6925	2.0551
9	3.1681	3.0902
10	3.6119	3.0923

Table 7: Scenario Generation Under Different Credit Risk Weightings

Scenario Number	Credit Risk Weighting		
	0.1	0.3	0.5
1	-0.3384	-0.3384	-0.3384
2	-0.1115	-0.1115	-0.1115
3	-0.0967	-0.1043	-0.1043
4	-0.0881	-0.0987	-0.0987
5	-0.1043	-0.0967	-0.0983
6	1.6168	-0.0881	-0.0971
7	1.8125	1.6168	-0.0967
8	1.8717	1.8717	-0.0881
9	2.0075	2.0075	-0.0881
10	3.0902	3.0902	1.6168

Figure 1: FTSE-100 Empirical Monthly Returns (%) for each Month [1]

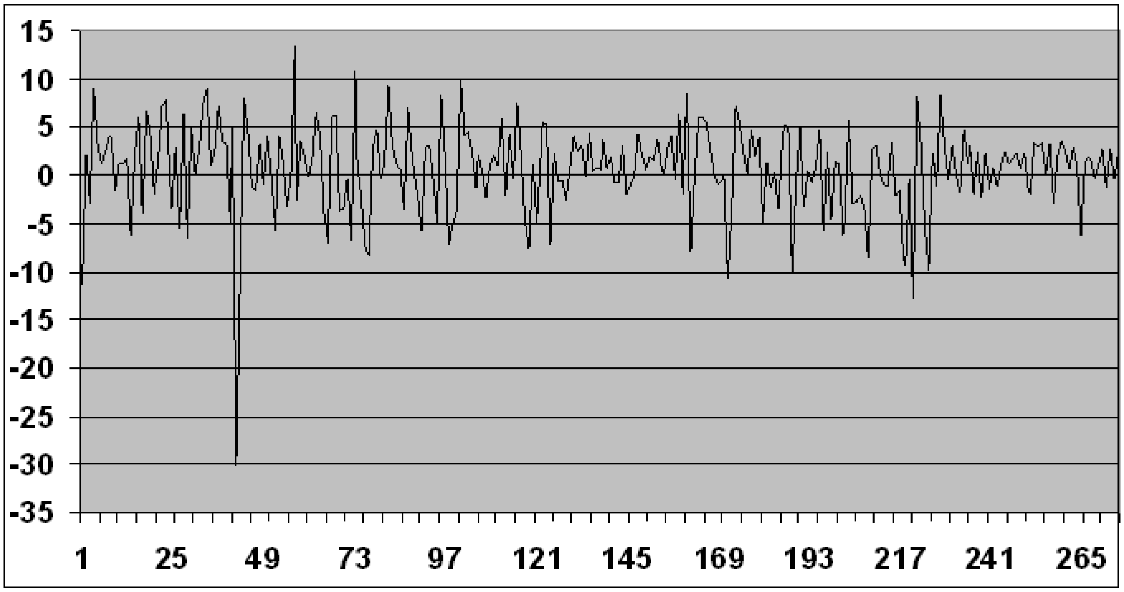
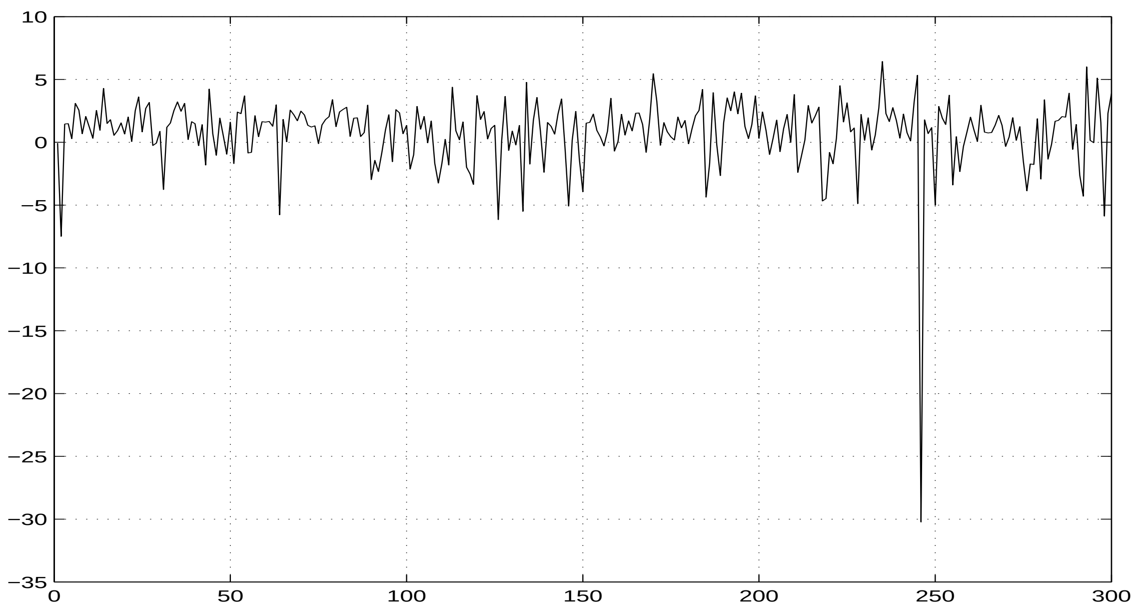


Figure 2: GM HMM Simulation of FTSE-100 Monthly Returns (%)



[1]from May 1984

From table 2's results we can see state 1 is the growth state, state 2 the recession state, state 3 is transitional (since the overall distribution means are positive, negative and about 0 respectively). We see from table 1 that if we are in state 2, then 51% of the time we will remain in state 2, similarly 71% for state 1.

We provide a GM HMM simulation of the FTSE-100 in figure 2 to compare its modelling performance to the empirical data (see figure 1). Simulating the GM HMM involves randomly selecting a Markov state in accordance to the transition matrix A and at each state randomly sampling a value from its pdf. Notice that the model captures many key features of the empirical data, namely extreme losses, autoregression and jumps.

Copula Approximation

Copulas are used to see a factor's impact upon the overall multivariate distribution e.g. mean, variance etc... . For example, we may wish to determine the impact upon a FTSE-100 index tracking portfolio (or any portfolio) from credit risk or operational risks [Voi03]. Rather than applying copulas to analyse the impact of operational and credit risks, we can apply a GM model and represent these two factors as two Gaussian components. For example, if we choose to examine the impact on state 1 then we may have the distribution:

Risk Factor	Mean	Sigma	Weighting
FTSE-100 Portfolio Return Component Gaussian 1	1.66	1.2	0.69
FTSE-100 Portfolio Return Component Gaussian 2	-30	0.1	0.01
Operational Risk	0	1	0.2
Credit Risk	-0.1	0.01	0.1

We can also calculate the moments, the pdf is a simply weighted sum:

$$f(\xi) = 0.7N(\xi, 1.41, 1.2) + 0.2N(\xi, 0, 1) + 0.1N(\xi, -0.1, 0.1). \tag{10}$$

Since the overall distribution is univariate, standard GM sampling and scenario generation techniques can be applied. This demonstates the simplicity and computational advantages of GM models compared to copulas.

Factor Specific Stress and Shock Testing (Robustness Analysis)

Rather than conducting a worst case scenario analysis for shock testing, we can more accurately model a "shock event" (such as a "credit crunch") with a Gaussian mixture distribution component of chosen weighting, mean and variance. For instance we may propose to stress test state 1's distribution using:

$$f(\xi) = 0.9(0.99N(\xi, 1.66, 1.2) + 0.01N(\xi, -30, 0.1)) + 0.1N(\xi, -10, 0.1), \qquad (11)$$

where the first two Gaussians represent state 1's (unstressed) distribution and the third represents an extreme event. Notice that the extreme event's inclusion fattens the left tail of the overall distribution so *all* scenarios sampled from this distribution are affected, unlike in standard worst case scenario analysis. We show a scenario tree generated from equation (11) in table 6; notice the presence of more negative events scenarios due to the "shock" event Gaussian.

Using Gaussian mixtures, we can conduct robustness analysis with respect to 1 particular risk factor and calculate the impact on the overall distribution. For example we vary the credit risk's weighting by 0.1,0.3,0.5 and show the impact on scenario generation in table 7. As can be seen in table 7, as we increase the credit weighting the impact of negative scenarios becomes more prevalent.

5. Conclusion

In this paper, a new method of scenario generation has been proposed and demonstrated. We have shown how the properties of GM HMM enable us to capture important time varying dynamics of data, in particular reversion and jumps. This enables us to produce more effective scenario trees. The tractable properties of GM enable us to further explore SP models, through richer stress and shock testing with respect to particular factors. Additionally we offer a simple method of approximating copulas for risk analysis. We have demonstrated these properties of GM HMM scenario generation by numerical experiments on FTSE-100 data.

References

[AGKL03] J. April, F. Glover, J.P. Kelly, and M. Laguna. Simulation-based optimization. *Proceedings of the 35th conference on Winter simulation*, 1(1):71–78, 2003.

[BDM71] A. Bignami and A. De Matteis. A Note on Sampling from Combinations of Distributions. *IMA Journal of Applied Mathematics*, 8(1):80, 1971.

[BL97] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 1997.

[BMS03] D. Brigo, F. Mercurio, and G. Sartorelli. Alternative asset-price dynamics and volatility smile. *Quantitative Finance*, 3(3):173–183, 2003.

[BS07] Saunders D. Buckley, I. and L. Seco. Portfolio optimization when asset returns have the Gaussian mixture distribution. *European Journal of Operational Research*, 2007.

[DDBMV] N. Di Domenica, G. Birbilis, G. Mitra, and P. Valente. Stochastic Programming and Scenario Generation within a Simulation Framework: An Information Systems Perspective. *Posted as SPEPS*, pages 15–2004.

[DT99] M.A.H. Dempster and R.T. Thompson. EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures. *Annals of Operations Research*, 90(1):161–184, 1999.

[Dup96] J. Dupacova. Scenario based stochastic programs: Resistance with respect to sample. *Annals of Oper. Res*, 64:21–38, 1996.

[EH81] B.S. Everitt and D.J. Hand. *Finite mixture distributions*. Monographs on Applied Probability and Statistics, London: Chapman and Hall, 1981.

[GRS04] N. Gülpınar, B. Rustem, and R. Settergren. Simulation and optimization approaches to scenario tree generation. *Journal of Economic Dynamics and Control*, 28(7):1291–1315, 2004.

[Ham] J.D. Hamilton. Regime-Switching Models.

[Ham89] J.D. Hamilton. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica*, 57(2):357–384, 1989.

[Har01] M.R. Hardy. A Regime-Switching Model of Long-Term Stock Returns. *North American Actuarial Journal*, 5(2):41–53, 2001.

[HKW03] K. Høyland, M. Kaut, and S.W. Wallace. A Heuristic for Moment-Matching Scenario Generation. *Computational Optimization and Applications*, 24(2):169–185, 2003.

[Hul00] J. Hull. *Options, futures and other derivatives*. Prentice Hall, 2000.

[HW01] K. Hoyland and S.W. Wallace. Generating Scenario Trees for Multistage Decision Problems. *Management Science*, 47(2):295–307, 2001.

[JR85] B.H. Juang and L.R. Rabiner. A probabilistic distance measure for hidden Markov models. *AT & T Bell Laboratories technical journal*, 64(2):391–408, 1985.

[JR91] B.H. Juang and L.R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[KW03] M. Kaut and S.W. Wallace. Evaluation of scenario-generation methods for stochastic programming. *Stochastic Programming E-Print Series*, 14(1), 2003.

[Lev05] S.E. Levinson. *Mathematical Models for Speech Technology*. Wiley, 2005.

[MGM00] J.M. Mulvey, G. Gould, and C. Morgan. An Asset and Liability Management System for Towers Perrin-Tillinghast. *Interfaces*, 30(1):96–114, 2000.

[MGPG06] C. Melodelima, L. Guéguen, D. Piau, and C. Gautier. A computational prediction of isochores based on hidden Markov models. *Gene*, 385(1):41–49, 2006.

[Mit06] S. Mitra. Scenario Generation for Stochastic Programming. *Optirisk Systems: White Paper Series*, 2006.

[Mur08] K. Murphy. Hidden markov model (hmm) toolbox for matlab. *http://www.cs.ubc.ca/ murphyk/Software/HMM/hmm.html*, 2008.

[Rab89] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[Reb04] R. Rebonato. *Volatility and Correlation: The Perfect Hedger and the Fox*. Wiley & Sons, Chichester, UK, 2004.

[SF06] T. Salih and K. Fidanboylu. Modeling and analysis of queuing handoff calls in single and two-tier cellular networks. *Computer Communications*, 29(17):3580–3590, 2006.

[TG01] E. Trentin and M. Gori. A survey of hybrid ANN/HMM models for automatic speech recognition. *Neurocomputing*, 37(1-4):91–126, 2001.

[TSM85] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical analysis of finite mixture distributions*. Wiley, 1985.

[Voi03] J. Voit. From Brownian motion to operational risk: Statistical physics and financial markets. *Physica A: Statistical Mechanics and its Applications*, 321(1-2):286–299, 2003.