# Modelling Financial Time Series with Switching State Space Models

Mehdi Azzouzi Ian T. Nabney
Neural Computing Research Group
Aston University, Birmingham B4 7ET, UK
azzouzim@aston.ac.uk i.t.nabney@aston.ac.uk

**Abstract**

The deficiencies of stationary models applied to financial time series are well documented. A special form of non-stationarity, where the underlying generator switches between (approximately) stationary regimes, seems particularly appropriate for financial markets. We use a dynamic switching (modelled by a hidden Markov model) combined with a linear dynamical system in a hybrid switching state space model (SSSM) and discuss the practical details of training such models with a variational EM algorithm due to [Ghahramani and Hinton, 1998]. The performance of the SSSM is evaluated on several financial data sets and it is shown to improve on a number of existing benchmark methods.

## 1 Introduction

Most traditional time series models are based on the assumption of stationarity: the underlying *generator* of the data is assumed to be globally time invariant. However, it is well known that for many financial time series this assumption breaks down. For instance, one of the obstacles to the effective forecasting of exchange rates is a non-constant conditional variance, known as heteroscedasticity. GARCH models have been developed to estimate a time-dependent variance [Bollerslev, 1986].

A local assumption of stationarity is nevertheless acceptable if the system switches between different regimes but each regime is (approximately) locally stationary. In fields from econometrics to control engineering, hybrid approaches have been developed in order to model this behaviour. One example is the mixture of experts [Jacobs et al., 1991], [Weigend et al., 1995] which decomposes the global model into several (linear or non-linear) local models (known as *experts* as each specialises in modelling a small region of input space). One limitation of these models is that the gating network which combines the local models has no dynamics. It is controlled only by the current value of the time series. One way to address this limitation is to use a hidden Markov model (which does have dynamics) to switch between local models. Autoregressive hidden Markov models (ARHMMs) switch between autoregressive models, where the predictions are a linear combination of past values. ARHMMs have been applied to financial engineering in order to model high frequency foreign exchange data and have shown promising results [Shi and Weigend, 1997].

Switching state space models (SSSMs) consist of multiple linear state space models with controlled by a dynamic switch. These models assume that the behaviour of the system can be characterised by a finite number of linear dynamical systems with hidden states, each of which tracks the dynamics in a different regime. A long-standing limitation of these models is that the complexity of the exact training algorithm grows exponentially with order $M^T$, where $M$ is the number of models and $T$ is the length of the time sequence. Various not completely satisfactory approximations have been proposed during the last decade [Bar-Shalom and Li, 1993]. Recently, [Ghahramani and Hinton, 1998] reintroduced the SSSM and proposed an efficient and principled approximate algorithm for training these models in a maximum likelihood approach. In this paper we propose to use switching state space models for modelling financial data. The approach is motivated by the fact that market behaviour at different time periods might be explained by different underlying regimes. Using an SSSM allows us both to create a predictive model and to discover at what times transitions occur between regimes (i.e. to *segment* the time series), based purely on price data.

The paper is structured as follows. In Section 2 we introduce switching state space models and show how the parameters can be learned by using *variational methods.* We review the problems of learning and inference and show how these models can be used for time series segmentation, probabilistic density prediction and risk estimation. In Section 3, we apply them to currency exchange rate data and compare the results with other standard techniques.

## 2 The model

Due to their flexibility and to the simplicity and efficiency of their parameter estimation algorithm, hidden Markov models (HMMs) and linear dynamical systems (LDS) have been the most widely used tools for learning probabilistic models of time series data. Both models represent the information about the past through a random variable: the *hidden* state. Conditioned on this state, the past and the future observation are independent.

In the case of HMMs, the state variable is discrete and can be viewed as a switching variable between different process regimes. For LDS, the hidden state is continuous and is specified by a linear dynamical equation (Equation 1). Both HMMs and LDSs can be trained efficiently in a maximum likelihood framework using the EM algorithm. The variant of the E-step for the HMM is known as the forward-backward algorithm and that for the LDS is the Kalman smoother.

Switching state space models (SSSMs) are a generalisation of both HMM and LDS: the dynamics can transition in a discrete manner from one linear regime to another. They can be regarded as a generalisation of mixture of experts and autoregressive HMMs [Poritz, 1988] (every autoregressive model can be indeed rewritten in a state space model form).

In an SSSM, $M$ different linear dynamical systems (or Kalman filters) compete in order to describe the observation $Y_t$. Each state vector $X_t^m$ evolves between time steps according to the system equation:

$$P(X_t^m \,|\, X_{t-1}^m) \sim \mathcal{N}\left(F_m X_{t-1}^m, Q_m\right) \tag{1}$$

where $F_m$ is the state transition matrix and $Q_m$ is the process covariance matrix associated with the state vector $X_t^m$. If we assume that the initial state vector has a Gaussian distribution $P(X_1^m) \sim \mathcal{N}(\mu_m, \Sigma_m)$, Equation 1 ensures that $P(X_t^m)$ is Gaussian at each time step $t$. A discrete variable $S_t \in \{1, \ldots, M\}$ plays the role of a gate. When the system enters a specific mode $m$, i.e. $S_t = m$, the observation is Gaussian and is given by:

$$P(Y_t \,|\, X_t^m) \sim \mathcal{N}(G_m X_t^m, R_m) \tag{2}$$

where $R_m$ is the output noise covariance matrix associated to the model (or state) $m$. The discrete state variable $S_t$ evolves according to Markov dynamics and can be represented by a discrete transition matrix $A = \{a_{ij}\}$

$$a_{ij} = P(S_t = j \,|\, S_{t-1} = i) \tag{3}$$

Therefore, an SSSM is essentially a mixture model, in which information about the past is conveyed through two types of random variable: one continuous and one discrete. Using the Markov dependence relations, the joint probability for the sequence of states and observations can be written as[1]:

$$\begin{aligned} P(S_1^T, X_1^{T^1}, \ldots, X_1^{T^M}, Y_1^T) = P(S_1) \prod_{t=2}^{T} P(S_t|S_{t-1}) \\ \prod_{m=1}^{M} P(X_1^m) \prod_{t=2}^{T} P(X_t^m|X_{t-1}^m) \prod_{t=1}^{T} P(Y_t|X_t^1, \ldots, X_t^m, S_t) \end{aligned} \tag{4}$$

## 2.1 Learning algorithm

Given a sequence of observations $Y_1^T = [y_1, \ldots, y_T]$, the learning problem consists of estimating the parameters $\Theta = \{F_m, Q_m, G_m, R_m, \mu_m, \Sigma_m\}_{1 \leq m \leq M}$ of each Kalman filter and the transition matrix $A$ of the discrete state Markov process in order to maximise the likelihood. An efficient procedure to solve this maximum likelihood estimation can be derived from the Expectation - Maximisation algorithm [Dempster et al., 1977]. The E-step (also called inference step) consists of computing the posterior probabilities $P(S_1^T, X_1^{T^1}, \ldots, X_1^{T^M} | Y_1^T, \Theta)$ of the hidden states $S_t$ and $X_t$. The M-step uses the expected values to re-estimate the parameters of the model.

Unfortunately, it can be shown that exact inference is not computationally tractable, since it scales as $M^T$. Indeed, even if $P(X_1^m|Y_1)$ is Gaussian, $P(X_t^m|Y_t)$ is in general a mixture of Gaussians with an exponential number of terms. Several approximations have been proposed to circumvent the inference problem [Bar-Shalom and Li, 1993], [Shumway and Stoffer, 1991]. Recently [Ghahramani and Hinton, 1998] proposed a generalised EM algorithm. The posterior distribution over the hidden states is *approximated* by a tractable distribution $Q$. The method maximises a lower bound on the likelihood by approximating the posterior probabilities with a parameterised distribution, called a *variational approximation* [Parisi, 1988]. Indeed, it can be shown that a judicious choice for $Q$ can render the inference step tractable [Saul and Jordan,

[1] We use the notation $O_1^T$ to denote the sequence of random variables $O_t$ from time 1 to time $T$.

1996]. The authors show that the E-step can be approximated by decoupling into forward-backward recursions on a HMM [Baum et al., 1970] and Kalman smoothing recursion [Rauch, 1963] on each state-space model, which are the relevant versions of the E-step for hidden Markov models and linear dynamical systems[2].

Once the posterior probabilities have been approximated, it is easy to derive re-estimations of the parameters $\Theta$. The parameters of the HMM are re-estimated using Baum-Welch equations and the parameters of each Kalman filter are re-estimated separately by weighting the observation $y_t$ by the responsibility assigned to each of them [Shumway and Stoffer, 1982].

## 2.2 Initialisation

Mixture models trained using the EM algorithm are guaranteed to reach a local maximum likelihood solution. Because there are many local maxima, such models are sensitive to how they are initialised. Therefore, the choice of initial conditions is crucial and we prefer to initialise the model carefully rather than a simple random initialisation.

For switching state space models, the initialisation is an important part of the learning algorithm, as both the HMM and the linear dynamical systems must be initialised. The key point is to start with a good segmentation of the data set, where by segmentation we mean a partition of the data, each being modelled by an LDS. [Ghahramani and Hinton, 1998] mentioned in their paper the importance of good methods for initialisation and modified the training algorithm so that the approximation distribution $Q$ is broadened with a parameter that is annealed over time. However, a large portion of training runs can converge to poor local maxima.

The initialisation algorithm we propose is the following: we first train an autoregressive hidden Markov model with as many discrete states as our SSSM on the data set and run the *Viterbi* algorithm in order to obtain the *most likely* path, i.e. the sequence of hidden states which 'best' explains the observation sequence [Rabiner, 1989]. Each data point is assigned to the most probable hidden state and thus gives us a segmentation of the data. A simple linear dynamical system is then initialised for each segment.

The parameters $a_{ij}$ of the discrete transition matrix $A$ can also be initialised by counting the number of transitions from state $i$ to state $j$ and dividing it by the number of transitions from state $i$ to any other state. For financial data, we have noticed problems with this method as it underestimates the number of samples the HMM remains in each state $i$. During the proper learning phase, this can lead to a model where some linear dynamical systems never update their parameters and so some LDSs are never responsible for data points. We therefore prefer to make an *ad hoc* adjustment where the diagonal entries are initialised to values closed to 0.90.

For synthetic data and the sleep apnea data set used in [Ghahramani and Hinton, 1998], we have shown elsewhere that our approach is a significant improvement (technical report in preparation).

[2] We recommend [Rabiner, 1989] and [Anderson and Moore, 1979] for a good overview of these models.

### 2.3 Making predictions

One-step ahead predictions on new data can be made by noting that each model contributes to the prediction of the observation $y_t$:

$$P(Y_t \mid S_t, X_t^1, \dots, X_t^M) = \prod_{m=1}^{M} [P(Y_t \mid X_t^m)]^{s_t^m} \tag{5}$$

where we have rewritten the switch variable $S_t$ as a vector $S_t = [s_t^1, \dots, s_t^M]$ with $s_t^m \in \{0, 1\}$. Thus on-line estimations for each model decouple naturally with the only modification that the likelihood of the observation $y_t$ is weighted by the *responsibility* $s_t^m$: Therefore the Kalman filter recursive equations hold for each model $m$ with the output covariance matrix $R_m$ weighted by $1/s_t^m$.

In Equation 5, the responsibility $s_t^m$ is unfortunately not known in advance but an expected value can be obtained by using Bayes' theorem:

$$s_t^m = E[S_t = m \mid Y_1^t] = \frac{P(Y_t \mid Y_1^{t-1}, S_t = m) P(S_t = m \mid Y_1^{t-1})}{P(Y_t \mid Y_1^{t-1})} \tag{6}$$

The first term in the numerator is given by Equation 2. The second term represents the predicted probability of the model $m$ at time $t$. As the discrete state $S_t$ is a Markov process, this probability is given by:

$$P(S_t = m \mid Y_1^{t-1}) = \sum_{n=1}^{M} a_{nm} P(S_{t-1} = n \mid Y_1^{t-1}) \tag{7}$$

The denominator is the normalising term and is given by:

$$P(Y_t \mid Y_1^{t-1}) = \sum_{m=1}^{M} P(Y_t \mid Y_1^{t-1}, S_t = m) P(S_t = m \mid Y_1^{t-1}) \tag{8}$$

*Hard* or *soft* competition can be implemented. In *hard* competition, only the Kalman filter $m$ with the highest predicted probability is running. In that case, $s_t^m = 1$ and $s_t^n = 0$ for the other models. In *soft* competition, $s_t^m = P(S_t = m \mid Y_1^t)$ and each model is allowed to adapt its parameters. Hence it is possible to estimate at each time $t$ the responsibility of each model and therefore detect mode transitions.

These recursion equations have been used in the control system community and are known as the *multiple model algorithm*. [Mazor et al., 1998].

## 3 Simulations

We propose to model financial time series in this probabilistic framework. Because of the capability of a Kalman filter to track quasi-stationary data and the power of HMMs for uncovering the hidden switching between regimes, we believe that such models are appropriate for financial time series. An advantage of viewing the model in a probabilistic framework is that we can also attach confidence intervals to the predictions, as the covariance matrix of the random variable $X_t$ is also estimated at each time step $t$. One immediate and important application for financial engineering concerns risk estimation. In addition, the value of the gate variable $S_t$ can be viewed as indicating the regime that the market is in at time $t$: this gives us a segmentation of the data, which is of value in its own right.

We have trained the model on several data sets of foreign exchange rates. We first present results of our simulations on DEM/USD foreign exchange rate daily returns. The training set contains 3000 points from 29/09/1977 to 15/09/1989. The test set contains 1164 points from 16/09/1989 to 05/11/1994. The first application of the model is to uncover underlying regimes: as an example, Figure 1 plots the segmentation obtained on the test set with a simple 2 state space model ($M = 2$). Because each Kalman filter learns the dynamics of a specific regime, the model is capable of detecting abrupt changes in the time series. For instance, we can see that each Kalman filter is activated for a certain range of volatility. *Soft* competition has been used and we clearly see segments where the two linear dynamical systems are used to explain the observation.

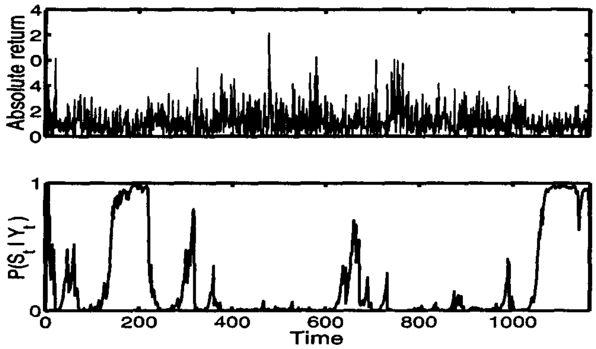

Figure 1: The top figure plots the DEM/USD absolute returns for the test set. The bottom figure shows the responsibility of one linear dynamical system for each time step.

As mentioned above, the model allows us to have an on-line estimate of the covariance of our prediction. Figure 2 plots the predictive distribution for a small window of time where a mode transition occurred at time $t = 35$. The figure shows how the confidence intervals change with respect to the mode. Indeed, the time window has been selected in order to show how the prediction are affected by a change of the volatility: the model moves from a high volatility region to a low volatility region: the predictions are of course affected by this change and we clearly see how the confidence intervals are sensitive to this transition.

We have also evaluated the performance of SSSMs with objective measures and compared them with other models. We trained autoregressive models (AR), GARCH models, MLP neural networks (NN) and autoregressive hidden Markov models (ARHMM) on three data sets: DEM/USD, GBP/USD and YEN/USD. The training GBP/USD and YEN/USD data sets contain 2000 points from 01/06/73 to 29/01/81 and the test sets contain 1164 points from 30/01/81 to 21/05/87.

Figure 3 shows the profit and loss curve of three models for a time window of 350 points of the DEM/USD test data set. For illustrative purposes, we assume no transaction cost. The SSSM has the highest profit in comparison with the ARHMM and neural networks. The other models do not give better results and

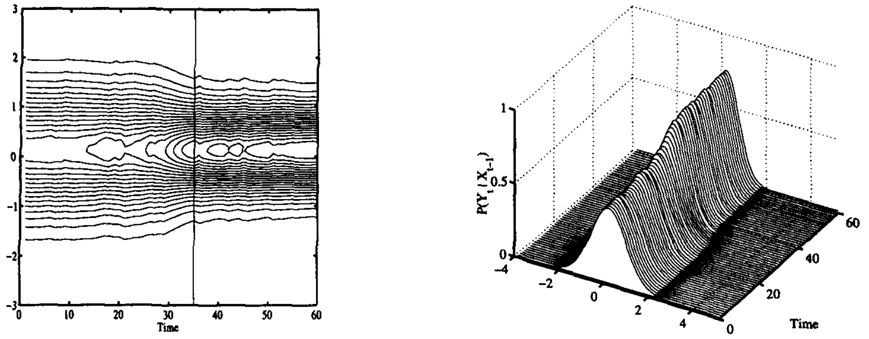

Figure 2: Contour plot and predictive distribution $P(Y_t|X_{t-1})$. The model switches from one state to another one, corresponding to a change of volatility.

to retain clarity we do not report the corresponding curves. Table 1 gives also

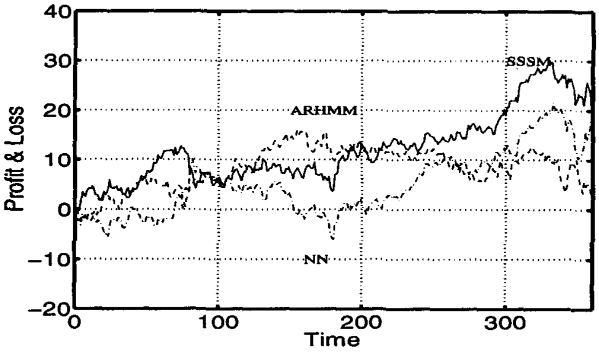

Figure 3: Profit and Loss of switching state space models (solid line) compared to autoregressive hidden Markov models (dashed line) and neural networks (dash dotted line).

the profits of the different models at the end of the same time window compared to simple 'short and hold' and 'buy and hold' strategies.

| AR | GARCH | NN | ARHMM | SSSM | Short | Buy |
|---|---|---|---|---|---|---|
| −3.47% | 13.35% | 10.28% | 7.03% | 23.45% | −16.15% | 16.15% |

Table 1: Profits of the different models compared to 'short and hold' and 'buy and hold' strategies at the end of the selected time window.

Table 3 compares the average performance of each model on each test data set. In this purpose, we have computed the likelihood, the normal mean squared error (NMSE) and the percentage accuracy (correct target sign prediction). For each model, 10 models initialised with different seeds have been trained.

Compared to other models, we clearly see that SSSM behave well on unseen data. Although exact computation of the likelihood is not tractable, the bound

DEM/USD

| Model | likelihood | | NMSE | | Hits | |
|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std |
| AR | −2.3957 | — | 1.0002 | — | 49.52 | — |
| GARCH | −1.1488 | — | 1.0000 | — | 35.04 | — |
| NN | −1.1950 | 0.0149 | 1.0190 | 0.0094 | 49.79 | 1.3500 |
| ARHMM | −1.0456 | 0.0020 | 0.9998 | 0.0000 | 49.32 | 0.1200 |
| SSSM | −1.1045 | 0.0154 | 0.9995 | 0.0004 | 53.30 | 0.0080 |

GBP/USD

| Model | likelihood | | NMSE | | Hits | |
|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std |
| AR | −2.5268 | — | 1.0020 | — | 47.04 | — |
| GARCH | −1.2174 | — | 0.9994 | — | 49.09 | — |
| NN | −1.2191 | 0.0316 | 1.0720 | 0.0188 | 47.91 | 0.8500 |
| ARHMM | −1.0730 | 0.0000 | 1.0030 | 0.0000 | 45.46 | 0.0001 |
| SSSM | −1.1362 | 0.0283 | 0.9996 | 0.0000 | 49.27 | 0.0020 |

YEN/USD

| Model | likelihood | | NMSE | | Hits | |
|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std |
| AR | −2.4508 | — | 1.0080 | — | 46.18 | — |
| GARCH | −1.1243 | — | 1.0020 | — | 47.63 | — |
| NN | −1.1253 | 0.03 | 1.0970 | 0.0120 | 47.54 | 0.8600 |
| ARHMM | −0.8930 | 0.00 | 1.0030 | 0.0000 | 46.55 | 0.0000 |
| SSSM | −1.0436 | 0.03 | 0.9996 | 0.0000 | 52.40 | 0.0020 |

Table 2: Average log-likelihood, normalised mean squared errors and hits on the test set over 10 runs. For AR, NN and ARHMM models, the input dimension has been simply taken to be 5. The neural network contains 10 hidden non-linear nodes and both ARHMM and SSSM have 3 hidden states. For SSSM, we give a lower bound on the likelihood.

is comparable to the best value (obtained with ARHMMs). When comparing the NMSE and the percentage accuracy, SSSMs always give the best results.

## 4 Conclusions

Switching state space models are powerful probabilistic models for modelling time series and their application in finance is new. In this paper, we showed how to train and initialise the models and how to use them for prediction. Because we model both the mean and variance of the conditional distribution, an interesting application for these models in financial engineering is risk estimation and building trading models. One promising extension of these models is to model the interaction of different currencies, by using them, either as inputs to the dynamical system or as multivariate output time series. We also intend to remove restriction of the Kalman filters to linear state space models in future work.

## 5 Acknowledgments

This work is funded by EPSRC contract K51815. We would like to thank Hing-Wan Cheng for the use of his MATLAB implementation of GARCH models.

## References

Anderson, B. D. O. and Moore, J. B. (1979). *Optimal Filtering.* Prentice-Hall, Englewood Cliffs, NJ.

Bar-Shalom, Y. and Li, X. R. (1993). *Estimation and Tracking.* Artech House, Boston, MA.

Baum, L., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41:164–171.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc.*, 39(1):1–38.

Ghahramani, Z. and Hinton, G. E. (1998). Switching state-space models. Technical report, Departement of Computer Science, University of Toronto. ftp:/ftp.cs.toronto.edu/pub/zoubin/switch-ftp.ps.gz.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixture of experts. *Neural Computation*, 3:79–87.

Mazor, E., Averbush, A., Bar-Shalom, Y., and Dayan, J. (1998). Interacting multiple model methods in target tracking: a survey. *IEEE Transactions on Aerospace and Electronic Systems*, 38(1):103–123.

Parisi, G. (1988). *Statistical Field Theory.* Addison-Wesley, Redwood City, CA.

Poritz, A. B. (1988). Hidden Markov models: A guided tour. In *IEEE International Conference on Acoustic Speech Signal Proceedings*, volume 198-8.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected application in speech recognition. In *Proceedings of the IEEE*, volume 77-2, pages 257–286.

Rauch, H. E. (1963). Solutions to the linear smoothing problem. *IEEE Transactions on Automatic Control*, 8:371–372.

Saul, L. and Jordan, M. I. (1996). Exploiting tractable substructures in intractable networks. In Touretsky, D. S., Mozer, D., and Hasselmo, M., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 427–234, Cambridge, MA. MIT Press.

Shi, S. and Weigend, A. S. (1997). Taking time seriously: Hidden Markov experts applied to financial engineering. In *Conference on Computational Intelligenge for Financial Engineering.* CIFEr, IEEE/IAFE.

Shumway, R. H. and Stoffer, D. S. (1982). An approach to time series smoothing and forecasting using the EM algorithm. *Journal of Time Series Analysis*, 3(4):253–264.

Shumway, R. H. and Stoffer, D. S. (1991). Dynamic linear models with switching. *J. Amer. Stat. Assoc.*, 86:763–769.

Weigend, A. S., Mangeas, M., and Srivastava, A. N. (1995). Nonlinear gated experts for time series. *International Journal of Neural Systems*, 3:373–399.