Dynamical Local Models for Segmentation and Prediction of Financial Time Series

Mehdi Azzouzi
azzouzim@aston.ac.uk

Ian T. Nabney
i.t.nabney@aston.ac.uk

Neural Computing Research Group
Aston University, Birmingham B4 7ET, UK

Abstract

In the analysis and prediction of many real-world time series, the assumption of stationarity is not valid. A special form of non-stationarity, where the underlying generator switches between (approximately) stationary regimes, seems particularly appropriate for financial markets. We introduce a new model which combines a dynamic switching (controlled by a hidden Markov model) and a non-linear dynamical system. We show how to train this hybrid model in a maximum likelihood approach and evaluate its performance on both synthetic and financial data.

Keywords: Time series segmentation, hidden Markov models, state space models, variational techniques, Bayesian error bars

1 Introduction

Most forecasting approaches try to predict the next value of a time series by assuming stationarity: *i.e.* the *underlying* generator of the data is globally time invariant. In many real world applications, this assumption is not valid. Even non-linear regressors like neural networks are not effective in modelling changing temporal structure in the time series. For instance, one of the obstacles to the prediction of exchange rates in the capital markets is a non-constant conditional variance, known as heteroscedasticity. GARCH models have been developed to estimate a time-dependent variance (Bollerslev, 1986).

A special form of non-stationarity, where the underlying generator switches between (approximately) stationary regimes, seems a reasonable assumption for many practical problems. In the last decade, hybrid approaches have been developed in order to model this behaviour. One example is the mixture of experts (Jacobs *et al.*, 1991; Cacciatore and Nowlan, 1994; 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 for time series analysis 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. For example, autoregressive hidden Markov models (ARHMMs) switch between autoregressive models, where the predictions are a linear combination of past values (Poritz, 1982). ARHMMs have been reintroduced in the machine learning community under the name of hidden filter HMMs (Fraser and Dimitriadis, 1994) and have been recently applied to financial engineering in order to model high frequency foreign exchange data (Shi and Weigend, 1997).

From econometrics to control, several similar hybrid models have been proposed. Their main characteristic is the mixing of discrete and continuous hidden variables (Chang and Athans, 1977; Hamilton, 1989; Shumway and Stoffer, 1991; Bar-Shalom and Li, 1993). A linear system with Markovian

coefficients, also called a jump-linear system, assumes the existence of a linear dynamical system of the general form:

$$\boldsymbol{x}_t = \boldsymbol{F}(\boldsymbol{s}_t)\boldsymbol{x}_{t-1} + \boldsymbol{u}_t \quad (1)$$

$$\boldsymbol{y}_t = \boldsymbol{G}(\boldsymbol{s}_t)\boldsymbol{x}_t + \boldsymbol{v}_t \quad (2)$$

where $\boldsymbol{x}_t$ is the state vector, $\boldsymbol{y}_t$ the measurement vector, and $\boldsymbol{s}_t$ the unknown time-varying parameter. $\boldsymbol{s}_t$ is restricted to take values from a finite set $\{q_1, \ldots, q_N\}$. In the simplest case, this parameter follows a first-order Markov process. The transition matrix governing the Markov chain and the parameters of the model are usually assumed to be known. The main problem consists thus of estimating the hidden state $\boldsymbol{x}_t$.

Chang and Athans (1977) focus on the state estimation problem for a system where the output matrix $\boldsymbol{G}$ is time independent. They show that estimation of the exact distribution of the state requires a bank of elemental estimators whose size grows exponentially in time. Mazor *et al.* (1998) review the state estimation problem for the most general case where both $\boldsymbol{F}$ and $\boldsymbol{G}$ are allowed to depend on a switch variable $\boldsymbol{s}_t$. They also show why an optimal solution is not computationally tractable and present techniques known as 'interacting multiple models' that consist of a bank of cooperating Kalman filters: at each time step t the state estimate is computed under each possible current model, with each filter using a different combination of the previous model-conditioned estimates (see also (Blom and Bar-Shalom, 1988; Bar-Shalom and Li, 1993)).

Shumway and Stoffer (1991) consider the problem of learning the parameters of a state space model with a switching output matrix $\boldsymbol{G}(\boldsymbol{s}_t)$ which is known in advance. They proposed an approximate EM algorithm where the E-step, which would require the computation of a mixture of Gaussians with an exponentially increasing number of components, is approximated at each time step t by a single Gaussian.

In this paper, we investigate switching state space models (SSSMs). These models consist of N multiple linear/non-linear state space models controlled by a dynamic switch and, in this sense are a generalisation of jump-linear systems. They assume that the behaviour of the system can be

characterised by a finite number of dynamical systems with hidden states, each of which tracks the data in a different regime. As discussed in (Ghahramani and Hinton, 1998), SSSMs can also be seen as a generalisation of the mixture of experts model.

A long-standing limitation for training these models is that the complexity of the exact training algorithm grows exponentially with order N^T, where N is the number of models and T is the length of the time sequence. Various *ad hoc* and not completely satisfactory approximations have been proposed, *e.g.* (Shumway and Stoffer, 1991). Recently, Ghahramani and Hinton (1998) reintroduced linear switching state space models in the machine learning community and proposed an efficient and principled approximate algorithm for training these models in a maximum likelihood framework.

In section 2 we first present linear switching state space models (SSSMs) and show how to train these models using variational techniques. In section 3 we present a new extension which incorporates non-linear state space models using radial basis function (RBF) networks. Although linear SSSMs enable us to model piece-wise stationarity, they may have difficulties in modelling non-linear dependencies in the time series. As the initialisation step is crucial for training mixture models due to the large number of local minima, we present a novel algorithm which addresses this problem in section 4. We then show how to use these models for time series segmentation and probabilistic density prediction. The models are finally tested on different datasets and we compare their performance with other standard techniques.

2 Linear switching state space models

Hidden Markov models and state space models are probabilistic models for time series where the information about the past is represented through a random variable: the hidden state. Conditioned on this state, the past and the future observations 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 SSMs, the hidden state is continuous and is specified by a linear dynamical equation.

A linear switching state space model (linear SSSM) is a model that combines HMMs and SSMs. More precisely, N different linear dynamical systems compete in order to describe the observation $\boldsymbol{y}_t \in \mathbb{R}^d$. Each real-valued state vector $\boldsymbol{x}_t^{(i)} \in \mathbb{R}^m$ evolves between time steps according to the system equation:

$$\boldsymbol{x}_t^{(i)} = \boldsymbol{F}_i \boldsymbol{x}_{t-1}^{(i)} + \boldsymbol{u}_i, \tag{3}$$

where $\boldsymbol{F}_i$ is the state transition matrix and $\boldsymbol{u}_i \sim \mathcal{N}(0, \boldsymbol{Q}_i)$ is a zero mean Gaussian noise associated to model i. The initial state vector is also assumed to be Gaussian: $P(\boldsymbol{x}_1^{(i)}) = \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$.

A discrete variable $S_t \in \{q_1, \ldots, q_N\}$, also represented by a vector $\boldsymbol{S}_t = [S_t^{(1)}, \ldots, S_t^{(N)}]$, where $S_t^{(i)} \in \{0, 1\}$, plays the role of a gate. When the system enters a specific state i, *i.e.* $S_t = q_i$ (or $S_t^{(i)} = 1$), the observation is Gaussian and is given by:

$$\boldsymbol{y}_t = \boldsymbol{G}_i \boldsymbol{x}_t^{(i)} + \boldsymbol{v}_i, \tag{4}$$

where $\boldsymbol{G}_i$ is the output matrix which maps the hidden state to the observation. The noise random variable $\boldsymbol{v}_i \sim \mathcal{N}(0, \boldsymbol{R}_i)$ is also zero mean Gaussian. The discrete state variable S_t evolves according to Markovian dynamics that can be represented by a discrete transition matrix $\boldsymbol{A} = \{a_{ij}\}$,

$$a_{ij} = P(S_t = q_j \mid S_{t-1} = q_i). \tag{5}$$

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

$$\begin{aligned} P(\mathcal{S}_1^T, \mathcal{X}_1^{T(1)}, \ldots, \mathcal{X}_1^{T(N)}, \mathcal{Y}_1^T) &= P(\boldsymbol{s}_1) \textstyle\prod_{t=2}^T P(\boldsymbol{s}_t | \boldsymbol{s}_{t-1}) \\ &\textstyle\prod_{i=1}^N \left(P(\boldsymbol{x}_1^{(i)}) \prod_{t=2}^T P(\boldsymbol{x}_t^{(i)} | \boldsymbol{x}_{t-1}^{(i)}) \right) \prod_{t=1}^T P(\boldsymbol{y}_t | \boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(N)}, \boldsymbol{s}_t). \end{aligned} \tag{6}$$

The corresponding graphical model is shown in Figure 1.

Given a sequence of observations $\mathcal{Y}_1^T$, the learning problem consists of estimating the parameters $\boldsymbol{\theta} = \{\boldsymbol{F}_i, \boldsymbol{Q}_i, \boldsymbol{G}_i, \boldsymbol{R}_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{1 \leq i \leq N}$ of each Kalman

filter and the transition matrix $\boldsymbol{A}$ of the discrete state Markov process in order to maximise the likelihood of the observations. An exact procedure to solve this maximum likelihood estimation could be derived from the Expectation-Maximisation algorithm (Dempster *et al.*, 1977). In the E-step, one computes the posterior probabilities $P(\mathcal{S}_1^T, \mathcal{X}_1^{T(1)}, \ldots, \mathcal{X}_1^{T(N)} | \mathcal{Y}_1^T, \boldsymbol{\theta})$ of the hidden states. 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 N^T. Even if $P(\boldsymbol{x}_1^{(i)} | \boldsymbol{y}_1, \boldsymbol{\theta})$ is Gaussian, then $P(\boldsymbol{x}_t^{(i)} | \boldsymbol{y}_1^t, \boldsymbol{\theta})$ is in general a mixture of Gaussians with an exponentially increasing number of terms. Like the other models described in section 1, the posterior distribution of the state variables $\boldsymbol{x}_t^{(i)}$ is a mixture of Gaussians with N^t components. Although these variables are marginally independent, they become conditionally dependent when the variable $\boldsymbol{y}_t$ is observed, namely because of the discrete variable $\boldsymbol{S}_t$ which couples all the real-valued state variables $\boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(i)}$ at time step t.

Several approximations have been proposed to circumvent this difficulty. For example, in (Shumway and Stoffer, 1991), a pseudo-EM algorithm is derived for learning a single hidden state space model with switching output matrices: at each step, the mixture of Gaussians is approximated by a single Gaussian. Recently Ghahramani and Hinton (1998) proposed a principled generalised EM algorithm. The idea is to make use of variational techniques in order to approximate the intractable true posterior distribution by a tractable distribution Q, and to maximise the lower bound on the log-likelihood:

$$\mathcal{F}(Q, \boldsymbol{\theta}) = \sum_{\mathcal{S}_1^T} \int Q(\mathcal{S}_1^T, \mathcal{X}_1^T) \log \frac{P(\mathcal{S}_1^T, \mathcal{X}_1^T, \mathcal{Y}_1^T \mid \boldsymbol{\theta})}{Q(\mathcal{S}_1^T, \mathcal{X}_1^T)} d\mathcal{X}_1^T, \tag{7}$$

where we have used Jensen's inequality and $\mathcal{X}_1^T$ denotes the whole sequence of hidden states: $\mathcal{X}_1^T = [\mathcal{X}_1^{T(1)}, \ldots, \mathcal{X}_1^{T(N)}]$. It is easy to see that the difference between the left-hand side and the right-hand side of Equation (7) is nothing else than the KL-divergence between the approximating distribution Q and the true posterior P. The KL-divergence is a non-negative expression and is minimised if and only if $Q = P$ in which case it is zero and the bound

becomes exact. However, this would not lead to any simplification of the problem.

Using a judicious structured variational approximation, the inference step can become tractable (Saul and Jordan, 1996). Because linear SSSMs are hybrid models combining HMMs and SSMs for which the E-step can be solved exactly, it is best to use an approximation that makes use of the forward-backward and Kalman smoother algorithms, which are the relevant versions for the respective E-step. The authors suggest the following approximation:

$$Q(\mathcal{S}_1^T, \mathcal{X}_1^{T(1)}, \ldots, \mathcal{X}_1^{T(i)}) = \frac{1}{Z}\Phi(\boldsymbol{s}_1)\prod_{t=2}^{T}\Phi(\boldsymbol{s}_{t-1}, \boldsymbol{s}_t)\prod_{i=1}^{N}\Phi(\boldsymbol{x}_1^{(i)})\prod_{t=1}^{T}\Phi(\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}) \tag{8}$$

which corresponds to the graphical model shown in Figure B.0.4. Z is a normalisation factor ensuring that Q integrates to one.

The motivation of such an approximation is to destroy the interaction between the hidden variables which makes the inference problem computationally intractable. Each deleted edge in the graph is replaced by a variational parameter:

$$\Phi(\boldsymbol{s}_1^{(i)}) = P(\boldsymbol{s}_1^{(i)})q_1^{(i)} \tag{9}$$

$$\Phi(\boldsymbol{s}_{t-1}^{(j)}, \boldsymbol{s}_t^{(i)}) = P(\boldsymbol{s}_t^{(i)} \mid \boldsymbol{s}_{t-1}^{(j)})q_t^{(i)} \tag{10}$$

$$\Phi(\boldsymbol{x}_1^{(i)}) = P(\boldsymbol{x}_1^{(i)})\left[P(\boldsymbol{y}_1 \mid \boldsymbol{x}_t^{(i)}, \boldsymbol{s}_1^{(i)})\right]^{h_1^{(i)}} \tag{11}$$

$$\Phi(\boldsymbol{x}_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}) = P(\boldsymbol{x}_t^{(i)} \mid \boldsymbol{x}_{t-1}^{(i)})\left[P(\boldsymbol{y}_t \mid \boldsymbol{x}_t^{(i)}, \boldsymbol{s}_t^{(i)})\right]^{h_t^{(i)}}. \tag{12}$$

By introducing these variational parameters, we decouple the state space models but keep the Markov chain assumption for each of them.

The variational parameters $q_t^{(i)}$ and $h_t^{(i)}$ are obtained by minimising the KL-divergence between P and Q, which corresponds to the E-step. Ghahramani and Hinton (1998) derived the fixed point equations for these parameters. The parameters $q_t^{(i)}$ play exactly the same role as the output probabilities $P(\boldsymbol{y}_t \mid \boldsymbol{x}_t^{(i)})$ would play in a regular hidden Markov model, and are obtained by computing the expected error under the distribution Q if state

space model i were used to generate the observation y_t:

$$q_t^{(i)} = \exp\left\{-\frac{1}{2}\langle(\boldsymbol{y}_t - \boldsymbol{G}_i\boldsymbol{x}_t^{(i)})'\boldsymbol{R}_i(\boldsymbol{y}_t - \boldsymbol{G}_i\boldsymbol{x}_t^{(i)})\rangle_Q\right\} \tag{13}$$

We can see that this parameter is a function of $\boldsymbol{x}_{t|T}^{(i)} \equiv \langle\boldsymbol{x}_t^{(i)}\rangle_Q$ and $\boldsymbol{V}_{t|T}^{(i)} \equiv \langle\boldsymbol{x}_t^{(i)}\boldsymbol{x}_t^{(i)'}\rangle_Q$. These expectations can be computed by running the Kalman smoother on state space model i with the observation $\boldsymbol{y}_t$ weighted by $h_t^{(i)}$ (see Equation (11) and Equation (12)). The parameters $h_t^{(i)}$ can be viewed as being the responsibility assigned to state space model i at time t, and are obtained by computing the expected probability of being in state i at time t under the approximating distribution Q.

$$h_t^{(i)} = \langle s_t^{(i)}\rangle_Q \tag{14}$$

We therefore see that the variational parameters are inter-related: the calculation of $q_t^{(i)}$ needs $h_t^{(i)}$ and vice-versa. Starting from some initial values for q and h, the E-step consists of running a Kalman smoother for each state space model with the output noise covariance matrix $\boldsymbol{R}_i$ weighted by $1/h_t^{(i)}$. This allows us to compute $q_t^{(i)}$ according to Equation (13) and the required expectations of each real-valued variable $\boldsymbol{x}_t^{(i)}$ needed in the M-step. The $h_t^{(i)}$ parameters are obtained by running a forward-backward algorithm, where each hidden state is associated to the output probability density $q_t^{(i)}$. The process is iterated until convergence of the KL-divergence. In practice, this is achieved in no more than 10 iterations.

The M-step consists of re-estimating the parameters $\boldsymbol{\theta}$ of the model and is straightforward. Like in HMMs and SSMs, the parameters can be re-estimated analytically. Appendix A.1 gives the re-estimation equations.

The whole process (E and M steps) is iterated until convergence of the lower bound on the log-likelihood.

3 Non-linear switching state space models

Although linear SSSMs are capable of modelling multi-modality, they may have difficulties in modelling non-linear dependencies in the time series. We

present here a new extension of dynamical local models which takes into account non-linearity in the output:

$$\boldsymbol{y}_t = g_i(\boldsymbol{x}_t^{(i)}) + \boldsymbol{v}_i, \tag{15}$$

where g_i denotes now a non-linear function from the hidden state space to the observation space. By introducing this non-linearity, the posterior $P(\boldsymbol{x}_t^{(i)}|\boldsymbol{y}_1^T)$ is no longer Gaussian and optimal smoothing cannot be achieved analytically.

In order to circumvent this problem, one solution could be derived from sequential Monte Carlo integration techniques (Kitagawa, 1987; Gordon *et al.*, 1993; Kitagawa, 1996). These techniques have been applied for the inference problem in non-linear state space models, and the extension to the case of non-linear switching state space models could be investigated. In these methods also known as bootstrap filter or sequential important sampling, arbitrary non-Gaussian densities are approximated by many particles that can be considered realisations from the distribution. It is then possible to derive a learning algorithm which makes use of these particles to fit the non-linear functions. However, these techniques are computationally expensive as a huge number of particles are needed at each time step t to be representative of the posterior distribution.

If the function g_i is sufficiently smooth, a suboptimal smoothing algorithm can be derived by considering the *linearisation* of the non-linear system. At every point $\boldsymbol{x}_{t|T}^{(i)}$, the function g_i is expanded as a first-order Taylor series:

$$g_i(\boldsymbol{x}) \approx g_i(\boldsymbol{x}_{t|T}^{(i)}) + \nabla_{\boldsymbol{x}} g_i(\boldsymbol{x}_{t|T}^{(i)})(\boldsymbol{x} - \boldsymbol{x}_{t|T}^{(i)}). \tag{16}$$

This approximate solution through linearisation around the current state estimate recovers the Gaussian structure and leads to the first-order *extended* Kalman smoother which is nothing else the exact Kalman smoother for the linearised model: the equations of the Kalman smoother are still valid except those involving the output matrix $\boldsymbol{G}_i$ which is replaced by the Jacobian matrix $\mathcal{J}_{t|T}^{(i)} = \nabla_{\boldsymbol{x}} g_i(\boldsymbol{x}_{t|T}^{(i)})$.

The second complication arises in the M-step. In the case of a linear model, it is easy to re-estimate the parameters exactly. If the functions g_i

are not linear, it may be computationally difficult to re-estimate exactly the parameters of the function. For example, if g_i is represented by a multilayer neural network, exact re-estimation cannot be done and we must resort to non-linear optimisation methods.

To solve these two problems, we propose to model each non-linear function with a radial basis function network:

$$\boldsymbol{y}_t = \sum_{k=1}^{K} w_k^{(i)} \psi_k^{(i)}(\boldsymbol{x}_t^{(i)}) + \boldsymbol{v}_i = \boldsymbol{W}^{(i)} \boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_t^{(i)}) + \boldsymbol{v}_i, \tag{17}$$

where $\boldsymbol{W}^{(i)} = [w_1^{(i)}, \ldots, w_K^{(i)}]$ are the weights (including the bias) and $\{\psi_k^{(i)}\}_{2 \leq k \leq K}$ denote the $(K-1)$ Gaussian basis functions associated to model i (the bias is associated to a basis function whose activation is equal to 1):

$$\psi_k^{(i)}(\boldsymbol{x}_t^{(i)}) = \exp\left(-\frac{||\boldsymbol{x}_t^{(i)} - \boldsymbol{m}_k^{(i)}||^2}{2{\sigma_k^{(i)}}^2} \right). \tag{18}$$

Note that non-Gaussian basis functions could be used although we did not investigate their implementation in this work.

In that case, with fixed basis functions, the M-step is still tractable since the output function is linear with respect to the weight matrix $\boldsymbol{W}^{(i)}$. A good initialisation enables us to keep the centres and widths of the basis functions fixed during the learning algorithm and to re-estimate only the weights, for which a fast and efficient algorithm exists[1]. Appendix B gives the re-estimation formulae for the weight matrix $\boldsymbol{W}^{(i)}$.

The number of basis functions K controls the smoothness of the output function g_i for each state space model. It is therefore possible to implement a non-linear SSSM with a number of basis functions that are different from one state space model to another. This can be quite useful if we believe, for example, that the underlying system is switching from a piecewise linear regime to a highly non-linear regime.

In terms of previous work, our model resembles that of (Kadirkamanathan and Kadirkamanathan, 1996), where the authors used modular RBF networks for learning multiple modes. Given input-output observations $\boldsymbol{z}_1^T =$

[1] If we want to learn these parameters, a generalised EM can be implemented.

$\{\boldsymbol{x}_1^T, \boldsymbol{y}_1^T\}$, their algorithm uses the Kalman filter for supervised recursive estimation of the weight vectors $\boldsymbol{W}^{(i)}$, which plays the role of the real-valued hidden state:

$$\boldsymbol{W}_t^{(i)} = \boldsymbol{W}_{t-1}^{(i)} + \boldsymbol{u}_i \tag{19}$$

$$\boldsymbol{y}_t = \boldsymbol{W}_t^{(i)} \boldsymbol{\Psi}_i(\boldsymbol{x}_t) + \boldsymbol{v}_i. \tag{20}$$

It is assumed that each model i has an associated score of being the current underlying model for the given observation $\boldsymbol{y}_t$. The parameters of the global model, for example the output noise covariance matrices $\boldsymbol{R}_i$ or the transition matrix $\boldsymbol{A}$, are not learned but are assumed to be known in advance. Our nonlinear model differs from the modular RBF network on two major points. Firstly, in our approach, the parameters of each expert are learned in a maximum likelihood framework. Secondly, whereas the weight vectors $\boldsymbol{W}^{(i)}$ play the role of the hidden states in their model, they are considered as proper adaptive parameters of each RBF network in our work. This leads to a system where the hidden state is an input to the RBF network and keeps therefore its intuitive interpretation of representing the underlying dynamics we are trying to recover.

4 Initialisation

Mixture models trained using the EM algorithm are guaranteed to reach a local maximum likelihood solution. Because there are many local maxima, experience has shown that SSSMs are particularly sensitive to the initialisation. 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 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, with each part modelled by a dynamical system. To address this problem, we have developed an efficient initialisation procedure.

For the linear case, we quickly[2] train a continuous 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. 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. This second phase can be done by estimating the covariance of the observations which allows us to initialise the output covariance $\boldsymbol{R}_i$. The system noise covariance $\boldsymbol{Q}_i$ can be, without any restriction, considered as a diagonal matrix and is simply initialised to the identity matrix. Values for $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$ are then obtained by inverting the system.

For the non-linear case, it is crucial to initialise properly the centres and the widths of each radial basis function, as these parameters will not be learned during the training algorithm. We first perform the initialisation for a linear SSSM. For each segment of the data where a linear dynamical system has been initialised, a corresponding sequence of hidden continuous states $\boldsymbol{x}_t$ can be recovered by running the Kalman filter. A Gaussian Mixture Model is fitted to each sequence, which enables us to initialise the centres and the widths of each RBF network.

The parameters a_{ij} of the discrete transition matrix $\boldsymbol{A}$ 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.

We have noticed that such an initialisation procedure alleviates problems occurring during the E-step. The KL-divergence can have several local minima corresponding to different values of the variational parameters. This means that two significantly different segmentations can lead to a similar lower bound on the log-likelihood. Ghahramani and Hinton (1998) addressed this problem and modified the training algorithm by using the technique of *deterministic annealing* (Ueda and Nakano, 1995): the approximation distribution Q is broadened with a *temperature* parameter that is annealed over time. However, with this method a large portion of training runs still converge to poor local minima.

[2] In practice, 5 iterations of the EM algorithm are sufficient.

In order to illustrate how our procedure can lead to a significant improvement, we consider the following synthetic problem involving a 2-state linear switching state space model:

$$x_t^{(1)} = 0.99x_{t-1}^{(1)} + u_1, \qquad u_1 \sim \mathcal{N}(0,1) \tag{21}$$

$$x_t^{(2)} = 0.90x_{t-1}^{(2)} + u_2, \qquad u_2 \sim \mathcal{N}(0,10) \tag{22}$$

The probability transition matrix $\boldsymbol{A}$ is such that $a_{11} = 0.99$ and $a_{22} = 0.98$. The output observation is identical for each model:

$$y_t = x_t^{(i)} + v, \qquad v \sim \mathcal{N}(0, 0.1) \qquad \forall i \tag{23}$$

We generated a sequence of $T = 1000$ points from this model and train linear SSSMs with the EM algorithm, considering three different learning techniques: our initialisation procedure, random initialisation without deterministic annealing and, random initialisation with deterministic annealing. For the deterministic annealing version, we follow Ueda and Nakano (1995): the variational parameters q and h are weighted by a decreasing temperature $\mathcal{T}$: starting with a relatively big value for $\mathcal{T}$, say $\mathcal{T} = 100$, the temperature is iteratively updated, $\mathcal{T}_i = \frac{1}{2}\mathcal{T}_{i-1} + \frac{1}{2}$, during the E-step. For each technique, 20 linear SSSMs corresponding to different random initial conditions were trained. We then evaluated the average mutual information between the true segmentation and the one obtained by each technique. Because the variational parameters h are real ($h_t^{(i)} \in [0,1]$), we first need to place a threshold on these values to obtain a *hard* segmentation[3].

Table 1 reports the results. Comparing the two random initialisations, on average, the deterministic annealing procedure performs slightly better. Our initialisation significantly outperforms both methods. We also report the average log-likelihood (lower bound) per data point for each technique. Compared to the likelihood obtained with the true model, each technique performs reasonably well. This shows the difficulty of comparing models when the exact computation of the likelihood is not tractable.

Figure 3 plots the time series and typical segmentations we obtain with the three approaches. Finding the true segmentation is actually very diffi-

[3] $h_t^{(i)} = 1$ if $h_t \geq 0.5$, $h_t^{(i)} = 0$ otherwise.

cult. Even when the inference is performed with the true model, an underestimation of the switching can occur, leading to a segmentation where only one state space model is activated.

5 Predictions and on-line model selection

In this section we show how to make one-step ahead predictions with dynamical local models. The algorithm makes use of Bayes' theorem at each time step t and is known as the multiple model approach (Bar-Shalom and Li, 1993).

At each time step t, we note that each model contributes to the explanation of the observation $\boldsymbol{y}_t$ in the following way:

$$P(\boldsymbol{y}_t \,|\, \boldsymbol{s}_t, \boldsymbol{x}_t^{(1)}, \ldots, \boldsymbol{x}_t^{(N)}) = \prod_{i=1}^{N} [P(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t^{(i)})]^{s_t^{(i)}} \tag{24}$$

Unfortunately the value of the switching variable is not known in advance, but an expected value can be derived by using Bayes' theorem:

$$\mathrm{E}[S_t = q_i \,|\, \mathcal{Y}_1^t] = \frac{P(\boldsymbol{y}_t \,|\, \boldsymbol{y}_1^{t-1}, S_t = q_i) P(S_t = q_i \,|\, \boldsymbol{y}_1^{t-1})}{P(\boldsymbol{y}_t \,|\, \mathcal{Y}_1^{t-1})} \tag{25}$$

The first term in the numerator is given by Equation (4). The second term represents the predicted probability of model i at time t given all the earlier observations. As the discrete state S_t is a first-order Markov process, this probability is given by:

$$\rho_t(i) \equiv P(S_t = q_i \,|\, \mathcal{Y}_1^{t-1}) = \sum_{j=1}^{N} a_{ji} P(S_{t-1} = q_j \,|\, \boldsymbol{y}_1^{t-1}) \tag{26}$$

The initial prior probabilities are assigned to be equal to $1/N$. The denominator is the normalising term (also known as the *evidence*) and is given by:

$$P(\boldsymbol{y}_t \,|\, \mathcal{Y}_1^{t-1}) = \sum_{i=1}^{N} \rho_t(i) P(\boldsymbol{y}_t \,|\, \mathcal{Y}_1^{t-1}, S_t = q_i) \tag{27}$$

Thus on-line estimations for each model decouple naturally. The Kalman filter recursive equations hold for each model i with the only modification that the likelihood of the observation $\boldsymbol{y}_t$ is weighted by $\rho_t(i)$.

Depending on the context, *hard* and *soft* competition can be implemented (Kadirkamanathan and Kadirkamanathan, 1996). In *hard* competition, it is believed that only one model is responsible for describing the observation at time t. This is done by considering only the model i with the highest predicted probability $\rho_t(i)$. In that case, $\rho_t(i) = 1$ and $\rho_t(j) = 0$ for the other models. In *soft* competition, $\rho_t(i) = P(S_t = q_i \mid \boldsymbol{y}_1^{t-1})$ and each model is allowed to adapt its parameters. This obviously leads to two different types of segmentations.

Thus the model inherits the properties from both HMMs and SSMs: the first-order Markov assumption for the discrete variable allows us to do on-line model selection. The state space model plays the role of the predictive model within each regime. As the mean and the covariance of the hidden states are updated on-line, the models allow us to obtain a full description of the predictive distribution.

6 Experimental results

We have assessed the performance of dynamical local models on different problems. We first run simulations on synthetic data in order to evaluate and compare the performances of linear and non-linear local dynamical models on data which exhibit local non-linearity. We finally show promising results of both models for modelling financial time series.

6.1 Synthetic data

We generated data from a bimodal process (Weigend *et al.*, 1995):

$$y_{t+1} = \begin{cases} 2(1 - y_t^2) - 1 & \text{if } s_t = 0, \\ \tanh(-1.2 y_t + \epsilon) & \text{if } s_t = 1. \end{cases} \tag{28}$$

where $\epsilon \sim \mathcal{N}(0, 0.1)$. The first mode is a deterministic chaotic process whereas the second mode is a noisy non-chaotic process. The switching obeys a first order Markov process with diagonal entries $a_{ii} = 0.98$. Both training and test datasets contain 500 points.

We trained both a linear and a non-linear switching state space model. The dimension of the hidden states $\boldsymbol{x}_t^{(i)}$ has been taken to be $m = 1$ and a RBF network with $K = 3$ hidden units has been used for the non-linear SSSM.

Figure 4 plots the test dataset and the corresponding segmentations obtained by the three models. Compared to the true segmentation, we can see that both models capture the underlying regime well, but that the non-linear SSSM is slightly more successful. Indeed, the correlations between the true segmentation and the ones obtained by the linear and non-linear SSSM are respectively 0.78 and 0.85.

Figure 5 plots the accuracy of the two models under the deterministic chaotic regime. Although the linear SSSM is able to capture the non-linearity, the non-linear SSSM seems to be more accurate[4]. This is particularly significant in the central region where there is a perfect match between the true underlying function and the output of the non-linear model.

We have also trained linear and non-linear dynamical systems on this dataset and we end this section by comparing linear and non-linear SSSMs with these single mode systems. The hidden state dimension of the linear models and the number of RBF units for non-linear models have been taken to be 3. Table B.0.4 reports the log-likelihood per datum and the normalised mean squared error (NMSE) on the test set and shows the significant improvement of the switching models. For each model, we report the average and the spread over 10 different initial conditions. It is interesting to note that an LDS of hidden state dimension 3 does not outperform the simple LDS with an hidden state of dimension 1. This remark does not apply to linear switching state space models: a linear SSSM of hidden state dimension 1 gives rise on average to a likelihood of -0.60 and a NMSE of 0.025.

6.2 Financial data

Because of the capability of state space models for tracking quasi-stationarity and the power of HMMs for uncovering the hidden switching between regimes,

[4]This is more obvious in the next table which reports the log-likelihood and the normalised mean squared error on the test dataset.

we investigate their performance on financial data. 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 $\boldsymbol{X}_t$ is also estimated at each time step t. One immediate and important application in financial engineering is risk estimation. In addition, the value of the discrete hidden 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 present here results of our simulations on DEM/USD and GBP/USD foreign exchange rate daily returns:

$$r_t = \log p_t - \log p_{t-1} \approx \frac{p_t - p_{t-1}}{p_{t-1}} \tag{29}$$

where p_t is the closing daily exchange rate at time t. This quantity can be seen as the logarithm of the geometric growth and is known in finance as continuous compounded returns.

Figure 6 plots the datasets. The DEM/USD 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 GBP/USD training set contains 2000 points from 01/06/73 to 29/01/81 and the test set contains 1164 from 30/01/81 to 21/05/87.

The first application of the model is to uncover underlying regimes. As an example, Figure 7 plots the segmentation obtained on the DEM/USD test set with a simple 3-state non-linear SSSM ($N = 3$). The dimension of each state space has simply been taken to $m = 1$ and the number of radial basis functions is $K = 5$. The figure shows how the model is capable of detecting abrupt changes in the time series structure. We can clearly see that the third model is responsible for the low volatility segments, the second for the higher volatility segments, and the first model is mainly responsible for the time period around $t = 500$ where the running mean is negative (rather than zero). In a simplistic view, the underlying regimes may be related to some macro-economical variables. Other simulations on higher frequency data have shown strong correlations between market movements and external events during the day, and it is easier to identify such regimes

when dealing with intra day data. For example, it is well known that market movements are more volatile at the open or the close of a trading day than at noon. Another example concerns news during the day that perturbs the financial markets. This volatility segmentation is easier to track during the day but there is no reason not to believe that daily closing price time series behave similarly on a lower frequency.

Another important application of dynamical local models in finance is the possibility of obtaining on-line estimates of the covariance of our prediction. Figure 8 shows a contour plot for a small window of time where a regime transition occurred at time $t = 35$. The model moves progressively from a high volatility region to a relatively low volatility region and the predictive distribution $P(\boldsymbol{y}_t|\mathcal{Y}_{t-1})$ is clearly affected by this change. Of course, understanding the volatility regimes is important for pricing of options.

We end this section by evaluating the performance of dynamical local models using 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 the same data sets. A GARCH model (Bollerslev, 1986) consists of a linear AR model for the conditional mean and an exponential AR model for the conditional variance. They are very often used in finance engineering for modelling quasi-stationarity. For AR, NN and ARHMM models, the input dimension has been simply taken to be 5 lagged values of the observations (which represent the history of the previous week), although no careful analysis of the input dimension has been carried out. Similarly, the neural network contains 10 hidden non-linear nodes and the ARHMM contains, like our models, 3 hidden states.

We have computed the log-likelihood per datum and the normal mean squared error (NMSE). For each model, we report the average and the spread over 10 different initial conditions. Dynamical local models have been initialised by the procedure we presented in Section 4.

Table 3 reports the results. On average, the NLSSSM seems to be the best model to describe the data, as the likelihood suggests it. When comparing the NMSE, we see that none of these models seem to outperform the

naive prediction, which would consist of making predictions based on the mean of the training set. Note, for example, that the log-likelihood for such a naive model is equal to -1.1575 on the DEM/USD dataset.

These simulations were intended to compare dynamical local models with other standard techniques used in computational finance and confirm the fact that predicting the daily return is a very difficult task. A better understanding of financial markets could be obtained by considering high frequency data. For example, Shi and Weigend (1997) modelled high frequency foreign exchange data with autoregressive hidden Markov models and showed promising results.

7 Discussion

In this paper we have reviewed hybrid models that combine hidden Markov models and state space models. These models have emerged from different scientific communities because of the necessity of modelling processes where the assumption of global stationarity does not hold.

We reviewed linear switching state space models and proposed a new extension which incorporates local non-linearity. This is done by using a local RBF network which maps the hidden state space to the observation space[5]. The structured variational approach allows us to perform a principled approximate maximum likelihood estimation of the parameters. The inference decouples nicely into the inference algorithms for HMMs and SSMs. In the case of non-linear dynamical models, a linearisation of the local function leads to the extended Kalman filter.

We also proposed an efficient and fast initialisation algorithm which alleviates problems of multiple local minima during the variational inference. This procedure leads to a significant improvement in the reliability of training compared to the deterministic annealing version.

In contrast to other hybrid models such as mixture of experts or autoregressive HMMs, dynamic local models provide a full description of the

[5]It must be emphasized that a Radial Basis Function network can be hardly seen as a 'true' generative model.

predictive distribution. This is an important issue, especially in finance where robust error bars need to be developed.

We evaluated the performance of the models on different data sets and compared them to other standard techniques. This was done by evaluating the log-likelihood per datum over a test set, as this measure allows direct comparisons between different models. Another evaluation of the density forecasts, based on the cumulative probability distribution, could complement our comparisons. This technique was proposed by Diebold *et al.* (1998) and consists of estimating the following random variable:

$$Z_{t+1} = \int_{-\infty}^{y^{t+1}} P(\eta \,|\, \mathcal{Y}_1^t)\; d\eta. \tag{30}$$

In order to assess the quality of the prediction, the random variable is tested against the hypothesis of a uniform distribution, which would correspond to a good model for the true predictive distribution $P^*(y_{t+1} \,|\, \mathcal{Y}_1^t)$. To test whether Z is uniformly distributed, Diebold *et al.* (1998) refer to standard techniques, the simplest of which consists of plotting the histogram.

These models have been applied to financial time series to extract two different types of information. Firstly, we can model the stochastic volatility, outperforming a GARCH model by a small but statistically significant margin. Secondly, we can segment the time series into different regimes. This is important, as there is growing evidence that financial time series are better modelled by a combination of local models, each of which specialises in a different segment, than a single complex global model.

The variational inference approach maximises a lower bound on the log-likelihood. An interesting problem concerns the quality of this bound which is a current open question. Empirical simulations using a dynamical local model containing a relatively small number of state space models, say $N = 2$ and a short time series, could be done to evaluate this quality. In that case, the exact estimation of the true posterior distribution of the hidden states can be performed and compared to the variational approximation.

Another comparison could be done by considering Monte Carlo integration techniques, such as Gibbs sampling, which provide a more accurate representation of the true posterior. This would also help us to evaluate

the performance of the extended Kalman filter for highly local non-linear dynamics.

Obviously, our models can be extended into several directions. In our work we did not consider exogenous variables as only a single time series is modelled. An immediate and straightforward extension consists of considering previous values of the time series as inputs in the dynamics of the hidden states:

$$\boldsymbol{x}_t^{(i)} = \boldsymbol{F}_i \boldsymbol{x}_{t-1}^{(i)} + \boldsymbol{H}_i \boldsymbol{y}_{t-q}^{t-1} + \boldsymbol{u}_i, \tag{31}$$

where the vector $\boldsymbol{y}_{t-q}^{t-1} = [y_{t-q}, \ldots, y_{t-1}]$ contains, for example, the last $q-1$ observations. We also did not consider non-linearities for the system equation. This is also an immediate extension of the non-linear dynamical local models, although we believe that the resulting algorithm would be too computationally costly and complex for practical application.

References

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

Blom, H. A. P. and Y. Bar-Shalom (1988). The interactive multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transaction on Automatic Control* **33** (8), 780–783.

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

Cacciatore, T. W. and S. J. Nowlan (1994). Mixtures of controllers for jump linear and non-linear plants. In J. D. Cowan, G. Tesauro, and J. Alspector (Eds.), *Advances in Neural Information Processing System*, Volume 6, pp. 719–726. San Francisco: Morgan Kaufmann.

Chang, C. B. and M. Athans (1977). State estimation for discrete systems with switching parameters. *IEEE Transactions on Aerospace and Electronic Systems* **14** (2), 418–424.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society* **39** (1), 1–38.

Diebold, F. X., T. A. Gunther, and A. S. Tay (1998). Evaluating density forecasts, with evaluation to risk management. *International Economic Review*.

Fraser, A. M. and A. Dimitriadis (1994). Forecasting probability densities by using hidden Markov models with mixed states. In A. S. Weigend and N. A. Gershenfeld (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*, pp. 264–281. Addison-Wesley.

Ghahramani, Z. and G. E. Hinton (1998). Switching state-space models. Technical report, Department of Computer Science, University of Toronto.

Gordon, N. J., D. J. Salmon, and A. F. M. Smith (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Proceedings of the IEEE*, Volume 140, pp. 107–113.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica* **57**, 357–384.

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

Kadirkamanathan, V. and M. Kadirkamanathan (1996). Recursive estimation of dynamic modular RBF networks. In G. Tesauro, D. S. Touretsky, and T. K. Leen (Eds.), *Advances in Neural Information Processing Systems*, Volume 8, pp. 239–245. MIT Press.

Kitagawa, G. (1987), December. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association* **82** (400), 1032–1063.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* **5**, 1–25.

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

Poritz, A. B. (1982), May. Linear predictive hidden Markov models and the speech signal. In *Proceedings of ICASSP*, pp. 1291–1294.

Saul, L. K. and M. I. Jordan (1996). Exploiting tractable substructures in intractable networks. In D. S. Touretsky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems*, Volume 8, pp. 486–492. Cambridge, MA: MIT Press.

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

Shumway, R. H. and D. S. Stoffer (1991). Dynamic linear models with switching. *Journal of the American Statistical Association* **86**, 763–769.

Ueda, N. and R. Nakano (1995). Deterministic annealing variant of the EM algorithm. In G. Tesauro, D. S. Touretsky, and J. Alspector (Eds.), *Advances in Neural Information Processing Systems*, Volume 7, pp. 545–552. Morgan Kaufmann.

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

A Implementation of dynamical local models

A.1 The EM algorithm for linear switching state space models

A.1.1 The E-step

The E-step involves the Kalman smoother for each state space model i where the output covariance matrix $\boldsymbol{R}_i$ is weighted by $1/h_t^{(i)}$ at each time step t. This allows to compute the variational parameters $q_t^{(i)}$ (Equation (13)). These parameters are then used in the forward-backward algorithm as output density probabilities, and this enables us to estimate the responsibility $h_t^{(i)}$ of each model. The whole process is repeated until convergence of the KL divergence, or similarly convergence of the lower bound.

A.1.2 The M-step

For the M-step, we make use of the re-estimations formulae for HMMs and SSMs. The re-estimation equations for the transition matrix A and the initial probabilities Π are exactly the same as those obtained for an HMM. Concerning the re-estimation equations of each linear dynamical filter, the equations are also the same except for the output matrices $\boldsymbol{G}_i$ and the output noise covariance matrices $\boldsymbol{R}_i$. We must indeed take into account the responsibility of each state space model. This responsibility is given by the value of the variational parameters $h_t^{(i)}$. It is easy to obtain:

$$\boldsymbol{G}_i^{new} = \left(\sum_{t=1}^{T} h_t^{(i)} \boldsymbol{y}_t \boldsymbol{x}_{t|T}^{(i)\prime}\right) \left(\sum_{t=1}^{T} h_t^{(i)} \boldsymbol{V}_{t|T}^{(i)}\right)^{-1} \tag{32}$$

$$\boldsymbol{R}_i^{new} = \sum_{t=1}^{T} h_t^{(i)} \left(\boldsymbol{y}_t \boldsymbol{y}_t' - \boldsymbol{F}_i^{new} \boldsymbol{x}_{t|T}^{(i)} \boldsymbol{y}_t'\right) / \sum_{t=1}^{T} h_t^{(i)}, \tag{33}$$

where $\boldsymbol{x}_{t|T}^{(i)}$ and $\boldsymbol{V}_{t|T}^{(i)}$ are obtained by running the Kalman smoother on each state space model.

B The EM algorithm for non-linear switching state space models

B.0.3 The E-step

The E-step involves the linearisation of each output function g_i. This function is approximated by an RBF network:

$$g_i(\boldsymbol{x}_t) = \boldsymbol{W}^{(i)}\boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_t^{(i)}), \tag{34}$$

where $\boldsymbol{W}^{(i)} = [w_1^{(i)}, \ldots, w_K^{(i)}]$ represents the weights (including the bias) and $\boldsymbol{\Psi}^{(i)} = [\psi_1^{(i)}, \ldots, \psi_K^{(i)}]$ are the basis functions. By linearising each basis function $\psi_k^{(i)}$, we get:

$$\begin{aligned}
\langle \boldsymbol{\Psi}^{(i)}(x_t^{(i)})\rangle_Q &= \boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)}) \\
\langle \boldsymbol{\Psi}^{(i)}(x_t^{(i)})\boldsymbol{\Psi}^{(i)}(x_t^{(i)})'\rangle_Q &= \boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})\boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})' + \mathcal{J}_{t|T}^{(i)}\boldsymbol{P}_{t|T}\mathcal{J}_{t|T}^{(i)'}
\end{aligned}$$

where $\mathcal{J}_{t|T}^{(i)} \equiv \left.\frac{\partial \psi_k^{(i)}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_{t|T}^{(i)}}$ is the Jacobian matrix.

B.0.4 The M-step

By taking the derivatives of the expected log-likelihood and setting them to zero, re-estimation formulae for the parameters are easily obtained. Because we just introduce non-linearity in the output function, the equations are the same as the ones for a linear switching state space model, except the output covariance matrix $\boldsymbol{R}_i$. We get:

$$\begin{aligned}
\boldsymbol{W}^{(i)^{new}} &= \left(\sum_{t=1}^{T} h_t^{(i)} \boldsymbol{y}_t \boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})'\right)\boldsymbol{\Lambda}_i^{-1} \\
\boldsymbol{R}_i^{new} &= \sum_{t=1}^{T}\left[h_t^{(i)}\left(\boldsymbol{y}_t - \boldsymbol{W}^{(i)^{new}}\boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})\right)\boldsymbol{y}_t'\right] / \sum_{t=1}^{T} h_t^{(i)}
\end{aligned}$$

with $\boldsymbol{\Lambda}_i = \sum_{t=1}^{T} h_t^{(i)}\left[\boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})\boldsymbol{\Psi}^{(i)}(\boldsymbol{x}_{t|T}^{(i)})' + \mathcal{J}_{t|T}^{(i)}\boldsymbol{P}_{t|T}^{(i)}\mathcal{J}_{t|T}^{(i)'}\right]$.

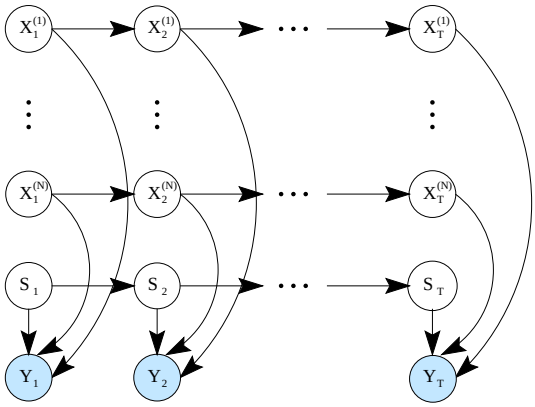


Figure 1: Graphical representation of a switching state space model. All the hidden variables have Markovian dynamics. At each time t, N real-valued hidden variables compete in order to explain the observation $\boldsymbol{y}_t$ and the discrete variable $\boldsymbol{s}_t$ plays the role of a gate.

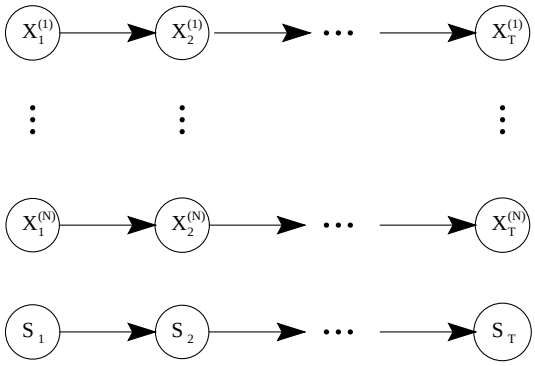


Figure 2: Structured variational approximation of a switching state space model. We have uncoupled the state space models but kept the Markov chain for each hidden variables. Exact inference for each hidden variable is now tractable.

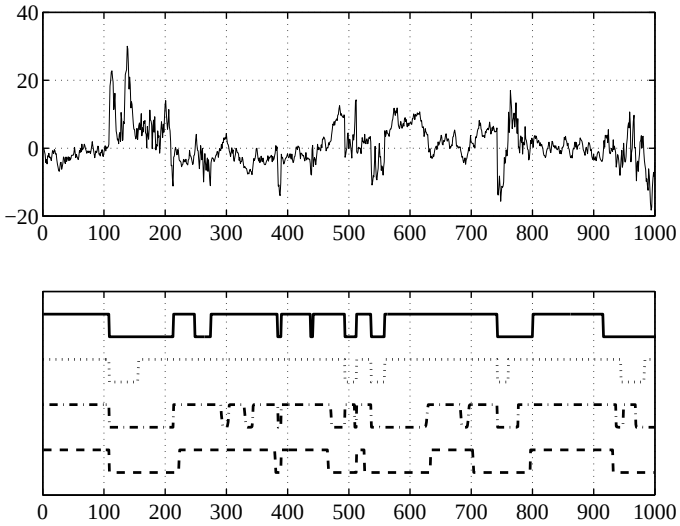


Figure 3: Synthetic time series (top) and segmentations (bottom) obtained with linear SSSMs compared to the true one (solid line): random initialisation without annealing (dotted line), random initialisation with annealing (dash dotted line) and initialisation (dashed line).

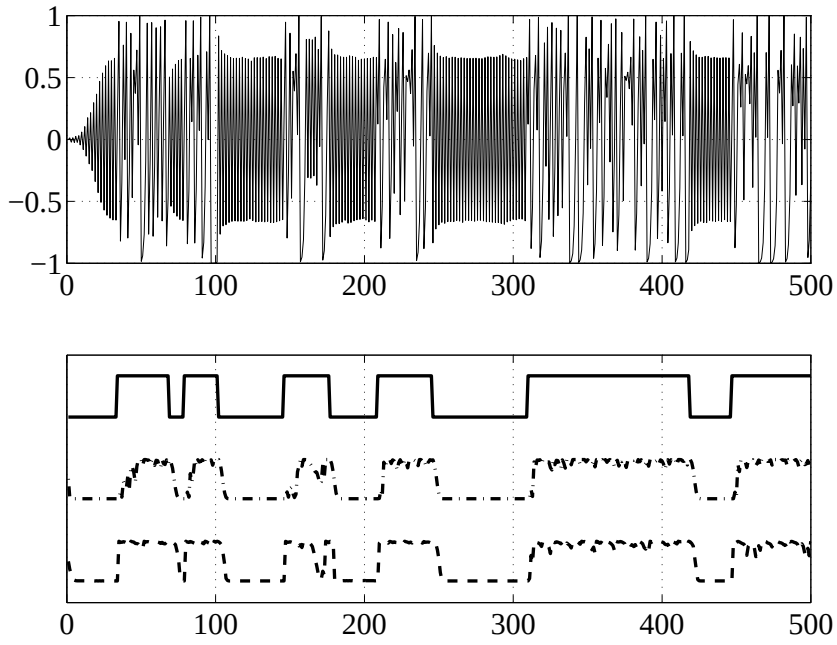
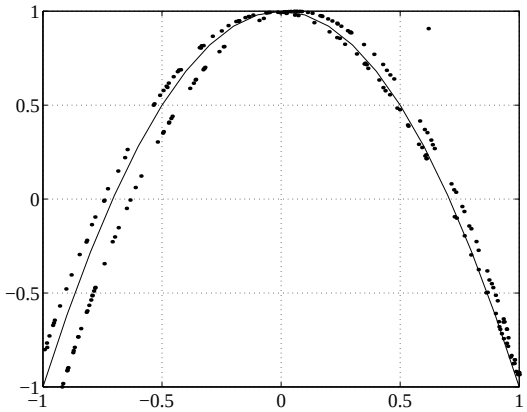
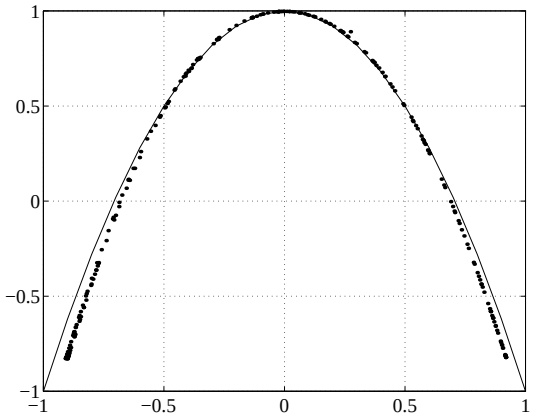


Figure 4: Test data and model probabilities for true (solid), linear (dash dotted) and non-linear (dashed) models.



(a)



(b)

Figure 5: Accuracy of the linear (a) and the non-linear (b) SSSMs in the chaotic regime. The solid line is the true function.

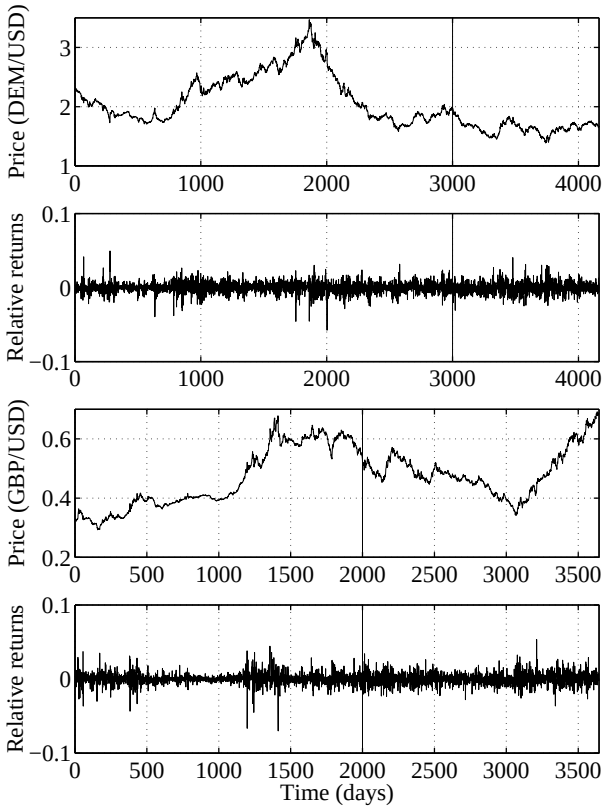


Figure 6: DEM/USD and GBP/USD training and test datasets used for evaluating dynamical local models.

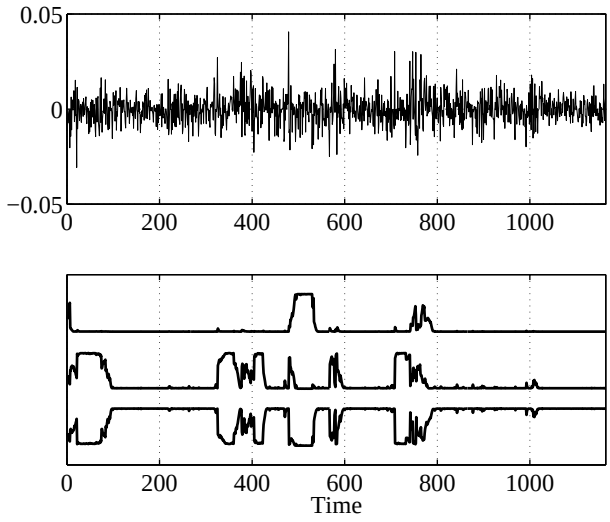


Figure 7: Predictive model probabilities $P(s_t \mid \mathcal{Y}_1^{t-1})$ obtained by a non-linear dynamical local model on the DEM/USD test set.

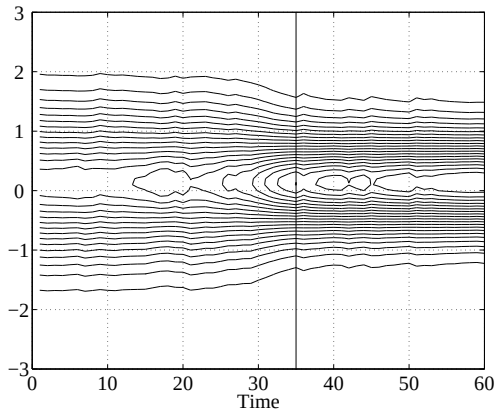


Figure 8: Contour plot of the predictive distribution $P(\boldsymbol{y}_t|\mathcal{Y}_{t-1})$. The model switches from one state to another one, corresponding to a change of volatility. The distribution is clearly more sharply peaked after the switch.

Technique	Mutual Info	Log-likelihood
No annealing	0.42	−2.26
Annealing	0.49	−2.26
Initialisation	0.77	−2.21
True model	1.73	−2.17

Table 1: Average mutual information and log-likelihood (lower bound) per data point when training linear dynamical model with and without initialisation. For information, we report the results obtained with the true model: the entropy of the true segmentation is 1.73.

Model	Log-likelihood		NMSE	
	mean	std	mean	std
LDS	−0.8601	0.0001	0.0339	0.0001
NLDS	−0.8020	0.0040	0.0292	0.0003
LSSSM	−0.5667	0.0107	0.0228	0.0004
NLSSSM	−0.4523	0.0221	0.0183	0.0013

Table 2: Average log-likelihood and NMSE on the test set for a simple linear dynamical system (LDS), a non-linear dynamical system (NLDS), a 2-state linear SSSM ($m = 3$) and a 2-state non-linear SSSM.

DEM/USD

Model	Log-likelihood		NMSE	
	mean	std	mean	std
AR	−2.3957	—	1.0002	—
GARCH	−1.1488	—	1.0000	—
NN	−1.1950	0.0149	1.0190	0.0094
ARHMM	−1.0456	0.0020	0.9998	0.0000
LDS	−1.1574	0.0000	0.9997	0.0000
NLDS	−1.1366	0.0030	0.9997	0.0001
LSSSM	−1.1045	0.0154	0.9995	0.0004
NLSSSM	−1.0361	0.0111	0.9995	0.0003

GBP/USD

Model	Log-likelihood		NMSE	
	mean	std	mean	std
AR	−2.5268	—	1.0020	—
GARCH	−1.2174	—	0.9994	—
NN	−1.2191	0.0316	1.0720	0.0188
ARHMM	−1.0730	0.0000	1.0030	0.0000
LDS	−1.2500	0.0000	0.9999	0.0000
NLDS	−1.2214	0.0020	0.9999	0.0001
LSSSM	−1.1362	0.0283	0.9996	0.0002
NLSSSM	−1.0581	0.0121	0.9996	0.0002

Table 3: Average log-likelihood and normalised mean squared errors on the DEM/USD and GBP/USD test sets over 10 runs corresponding to different initial conditions.