NEURAL COMPUTING
NCRG
RESEARCH GROUP

Neural Computing Research Group
Dept of Computer Science & Applied Mathematics
Aston University
Birmingham B4 7ET
United Kingdom
Tel: +44 (0)121 333 4631
Fax: +44 (0)121 333 4586
http://www.ncrg.aston.ac.uk/

Delay Estimation for Multivariate Time Series

Mehdi Azzouzi
azzouzim@aston.ac.uk

Ian T. Nabney
nabneyit@aston.ac.uk

Technical Report NCRG/98/026 January 5, 1999

Abstract

Most traditional methods for extracting the relationships between two time series are based on cross-correlation. In a non-linear non-stationary environment, these techniques are not sufficient. We show in this paper how to use hidden Markov models (HMMs) to identify the lag (or delay) between different variables for such data. We first present a method using maximum likelihood estimation and propose a simple algorithm which is capable of identifying associations between variables. We also adopt an information-theoretic approach and develop a novel procedure for training HMMs to maximise the mutual information between delayed time series. Both methods are successfully applied to real data: we show that HMMs are capable of modelling the oil drilling process and that they outperform existing methods for computing a crucial parameter, namely the lag for return.

Keywords: lag detection, hidden Markov models, non-stationarity, regime switching, EM algorithm, mutual information.

1 Introduction

A key part of multivariate time series analysis is identifying the lags or delays between different variables. This differs from characterising the order or degree of freedom of a single time series, where the goal is to estimate the intrinsic dimensionality of the data in order to model the deterministic component of the data generator (Broomhead and King, 1986). In the latter case, the correlation of past values usually tails off gradually, so that the most recent samples have the largest impact on the current value. This is not the case where two time series x_t and y_t are related by a lag δ. There will be no relationship between x_{t-d} and y_t for $d < \delta$.

Under the assumption of stationarity, cross-correlation is a powerful tool for measuring and modelling linear relationships between variables (Kendall and Ord, 1990). The cross-correlation can then be used in linear model identification procedures; they are often used as the basis for identifying the order of non-linear models, as they are fast to compute. However, in many real-world applications the assumptions of linear dependencies and stationarity are not valid.

In this paper, we consider the problem of modelling processes which manifest a sequentially changing behaviour: the parameters of the data generator usually remain constant, except for minor fluctuations, and then, at certain times, change to another set of values.

Our approach is based on using Hidden Markov models (HMMs) to model the distribution of the time series. More precisely, given two time series x_t and y_t, related by a lag δ and generated from a non-stationary underlying process which exhibits different regimes, we show how HMMs can be used to estimate the value of δ. Due to their flexibility and to the simplicity and efficiency of their parameter estimation algorithm, HMMs have proven to be one of the most widely used tools for learning probabilistic models of time series data. An HMM is essentially a mixture model, in which information about the past is conveyed through a single discrete variable, the hidden state. In certain circumstances, this state can be viewed as a switching variable between different process regimes.

To estimate parameters for an HMM in a maximum likelihood framework, one can use the well-known *Baum-Welch* algorithm (Baum et al., 1970), which is the relevant version of the EM algorithm (Dempster et al., 1977). In Section 3 we develop a novel procedure for training HMMs to maximise the mutual information (MMI) between two time series x_t and y_t and compare it with maximum likelihood estimation (MLE). The Baum-Welch algorithm is a hill-climbing algorithm which does not require the cost function gradient. Unfortunately, no such method is known for MMI estimation and we must therefore resort to the use of traditional maximisation techniques that do use the cost function gradient.

We apply this approach to the analysis of the oil well drilling process. The process exhibits complex time relationships between variables and a highly non-stationary behaviour. A fluid called 'mud' carries the drilling cuttings up the hole to the surface. The time it takes for the cuttings to come up to the surface is called the *lag for return* and is a crucial parameter for modelling and understanding the process. This time-varying parameter depends not only on the depth of the hole and the pressure of the drilling fluid, but also on the geology of the surrounding rock formation and the drilling mode. In Section 4 we analyse drilling data and compare our results with cross-correlations and the numerical models based on fluid mechanics which are currently used operationally.

2 Modelling Multivariate Time Series with Hidden Markov Models

A multivariate continuous time series is a sequence of continuous m-dimensional random variables $\mathcal{O}$, such that for each time t, O_t ranges over a continuous space. For simplicity, suppose that

$m = 2$ so that at time T we have seen a sequence of observations $o_1^T = [o_1, \ldots, o_t, \ldots, o_T]$[1] where $o_t = (x_t, y_t)$ is the observed data at time t. Given such a sequence, the task is to model the probabilistic distribution from which the time series was generated.

Let $\mathcal{S}$ be a discrete random variable taking values in the set $\{q_1, \ldots, q_N\}$ and assume that the system at any time t is in one and only one of the N states $q_1, \ldots, q_N$. The random variable O_t can be considered to be a probabilistic function of the underlying states, i.e. o_t is an observed measurement from the system but the underlying states are not themselves directly observable. Assuming that the state variable S_t is a stationary discrete-time first-order Markov process, the resulting model is a doubly stochastic process and is called a first-order hidden Markov model (HMM). It is called hidden because the state of the underlying process is *not* observable, but can only be observed through another set of stochastic processes that produce the sequence of observations. Thus the model assumes two sets of conditional independence relations: that O_t is independent of all other random variables given S_t and that S_t is independent of $S_1, \ldots, S_{t-2}$ given S_{t-1} (the Markov property). Using these independence relations, the joint probability for the sequence of states and observations can be written as

$$P(O_1^T, S_1^T) = P(S_1)P(O_1 \mid S_1) \prod_{t=2}^{T} P(S_t \mid S_{t-1})P(O_t \mid S_t) \tag{2.1}$$

and can be expressed graphically in the form of Figure 1. In general, the parameters of a specific

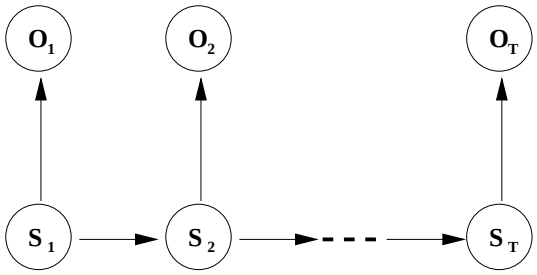

Figure 1: Graphical representation of an HMM, specifying the independence relations.

model are referred as $\Theta = \{A, B, \Pi\}$, where $A = \{a_{ij}\}$ denotes the state transition matrix, $B = \{b_i(o_t)\}$ the observation probability distribution in each state and $\Pi = \{\pi_i\}$ the initial state distribution

$$a_{ij} = P(S_t = q_j \mid S_{t-1} = q_i) \tag{2.2a}$$
$$b_i(o_t) = p(o_t \mid S_t = q_i) \tag{2.2b}$$
$$\pi_i = P(S_1 = q_i) \tag{2.2c}$$

For time series modelling, the probability distributions B are often chosen to be a mixture of Gaussians, as such models can approximate, arbitrarily closely, any finite, continuous density function, provided that enough components are used[2]. HMMs have been successfully applied in speech recognition (Rabiner, 1989), (Poritz, 1988), cryptography, and more recently in other areas such protein classification and sequence alignment (Baldi et al., 1993).

Given a model with known parameters Θ and a sequence of observations, two algorithms are commonly used to solve two different forms of the inference problem (Rabiner, 1989). The first computes the posterior probabilities of the hidden states using a recursive algorithm known as the *forward-backward* algorithm. It is also an efficient way to compute the likelihood of the observation sequence $\mathcal{L} = \log p(O_1^T \mid \Theta)$. The other inference problem is to find a sequence of hidden states

[1] We use the notation O_i^j to denote the sequence of random variables from time i to time j, i.e. $O_i^j = [O_i, O_{i+1}, \ldots, O_j]$. A sequence of observations will be denoted $o_i^j = [o_i, o_{i+1}, \ldots, o_j]$.

[2] Although continuous density HMMs are applicable to a large number of problems, autoregressive HMMs, where the observation vectors are drawn from a state-dependent autoregressive process, have been investigated for time series modelling (Poritz, 1982). In speech recognition, neural networks are used to model $P(S_t \mid O_t)$; this probability is then converted via Bayes rule to the output probability $P(O_t \mid S_t)$ (Bourlard, 1997).

which 'best' explains the observations sequence. In that case, we attempt to uncover the hidden part of the model. The most widely used criterion is to find the single most likely path, i.e. to maximise $P(S_1^T \mid O_1^T, \Theta)$ which is equivalent to maximising Equation 2.1. For this purpose, the *Viterbi* algorithm, a particular case of dynamic programming, offers an efficient solution. Typical uses might be to learn about the structure of the model and to segment the time series into different regimes if we believe that the system operates in multiple modes and switches its dynamics.

Given a sequence of observations, the learning problem consists of estimating the parameters of the model in order to maximise the likelihood. An efficient procedure to solve this maximum likelihood estimation (MLE) problem exists and is called the *Baum-Welch* algorithm. This algorithm is a special case of the EM algorithm in which the E-step consists of using the *forward-backward* algorithm in order to compute the posterior probabilities of the hidden states. The M-step uses the expected counts of transitions to re-estimate the parameters of the model (the algorithm is described in more details in Section 3).

Several descendants of HMMs as probabilistic models for time series have been proposed. In order to include input variables in the state transition, the input output HMM (IOHMM) architecture has been suggested (Bengio and Frasconi, 1995). The probability $P(S_t \mid S_{t-1}, U_t)$, where U_t is the input variable is modelled using neural networks. Another architecture which combines real-valued and discrete states in a *mixed* state $\phi_t = (S_t, R_t)$ which summarises histories is presented in (Fraser and Dimitriadis, 1994) under the name of hidden filter HMM (HFHMM). As a simplification, the continuous part R_t is taken to be a deterministic function of past observations $R_t = f(O_1^{t-1})$.

3 Training algorithms for HMMs

In this section we begin by reviewing the usual maximum likelihood estimation (MLE) procedure for training HMMs. We then present a novel approach based on Maximum Mutual Information (MMIE) and show how these methods can be applied to detect relationships between time series in a non-linear non-stationary environment. Our problem can be stated in the following way: a sequence of observations $o_1^T = [(x_1, y_1), \ldots, (x_T, y_T)]$ is being generated by an underlying system. Unfortunately, we do not see the true sequence o_1^T but a modified version where one variable is delayed: $o_1^T(\delta) = [(x_{1-\delta}, y_1), \ldots, (x_{T-\delta}, y_T)]$. Our task is to estimate the value of δ. Given a two dimensional time series vector $O_1^T(\delta) = (X_{t-\delta}, Y_t)_{t=1,\ldots,T}$, we say that Y_t leads X_t by an unknown lag δ. For convenience and clarity, we define $O^d = (X^d, Y)$ and $X^d \equiv X_1^T(d) = [X_{1-d}, \ldots, X_{T-d}]$ the time series X_t delayed by d steps, omitting time indexes. The problem can be viewed as

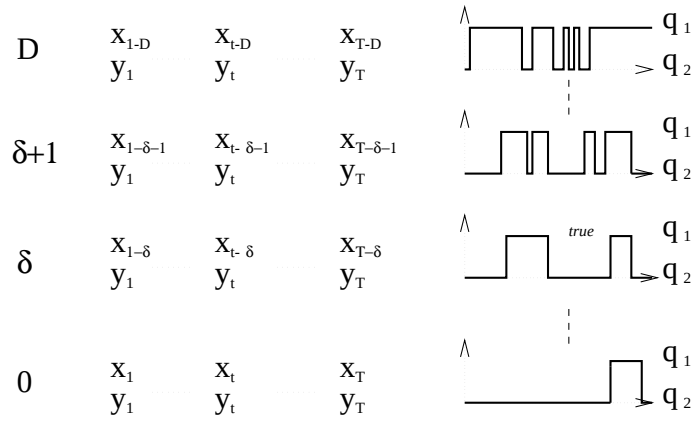

Figure 2: The synchronisation problem: we assume an underlying *true* state sequence S_{true}. This sequence is not observable but can be recovered by identifying the corresponding observation sequence O^δ, i.e. when X_t and Y_t are properly synchronised.

a *synchronisation* problem, where the goal is to recover the *correct* sequence of hidden states. Figure 2 shows the underlying sequence $S(d)$ corresponding to different delayed time series (in this

example, the system can switch between two states q_1 and q_2). The value δ corresponds to a *true* sequence of hidden states and the task is to recover this sequence by identifying the corresponding observation sequence O^δ.

3.1 Maximum Likelihood

As mentioned in Section 2, the usual procedure for training HMMs is to find parameter values of Θ that maximise the likelihood or the log-likelihood of the observed sequence of training data O_1^T:

$$\Theta^* = \arg\max_{\Theta} \log p(O_1^T \mid \Theta) \tag{3.1}$$

The standard approach is the Baum-Welch algorithm, which is the relevant version of the EM algorithm (Dempster et al., 1977). The expectation-maximisation algorithm is a general algorithm for maximum likelihood estimation (MLE) in parametrised models for incomplete data. An HMM is essentially a mixture model which includes unobserved data, the hidden variables S_t. The algorithm is iterative and generates from some initial guess Θ^0 a sequence Θ^n of estimates. Each iteration consists of two steps:

1: E-step: Compute the expected log-likelihood of the complete (observed and hidden) data

$$Q(\Theta^{new}, \Theta^{old}) = E[\log P(O_1^T, S_1^T \mid \Theta^{new}) \mid P(S_1^T \mid O_1^T, \Theta^{old})] \tag{3.2}$$

2: M-step: Find Θ^{new} which maximises $Q(\Theta^{new}, \Theta^{old})$

Q is a function of the parameters Θ^{new}, given the current parameters Θ^{old} and the time series O_1^T. At the M-step, the maximum is found by differentiating the expected log-likelihood with respect to Θ^{new} and solving the resultant linear equations. Baum and his colleagues showed that after each iteration, the likelihood will be at least as great as it was after the previous iteration[3], and convergence to a local maximum is guaranteed. However, there are many sub-optimal local maxima in the likelihood surface, and there is no guarantee to convergence to a global maximum. In practice, the choice of initial conditions is highly influential. The details of the algorithm are reviewed in Appendix A.

For a multivariate time series $O = (X, Y)$, MLE will maximise the joint probability $P(X, Y \mid \Theta)$ and thus produce the most likely parameters, for a given model structure. Using this approach, training an HMM denoted by Θ^d with MLE will allow us to obtain the most likely model parameters used to generate the delayed sequence O^d. Assuming the existence of a *true* value $d = \delta$ that relates $X_{t-\delta}$ to Y_t, δ can therefore be computed by finding the most likely model:

$$\delta = \arg\max_{d} \mathcal{L}^d \tag{3.3}$$

where $\mathcal{L}^d$ denotes the log-likelihood of the delayed sequence O^d given the model Θ^d. We are thus able to derive an algorithm in order to solve the problem of lag detection. The algorithm we propose consists of two levels: the first level is to derive the estimates of the HMM Θ^d trained with the delayed sequence O^d. The second level is to estimate the likelihood of each model Θ^d in order to get the most likely one. The approach is motivated by the fact that the sequence O^d corresponds to a specific sequence of hidden states representing the dynamics of the process (Figure 2). Intuitively, we expect that $\mathcal{L}^d$ will always be less than $\mathcal{L}^\delta$ ($d \neq \delta$). Indeed, assuming that for each time step t, $X_{t-\delta}$ and Y_t have been generated by a specific state S_t^*, the system will not be able to enter that *true* state if X_t and Y_t are not properly synchronised.

3.2 Maximum Mutual Information

There are many very important properties of the maximum likelihood estimate (MLE) but most of them stem from an implicit assumption of model correctness. The justifications for using MLE

[3] For clarity on this subject, we recommend (Brown, 1987) and (Liporace, 1982).

to estimate the mean and the variance of a Gaussian distribution, for example, presume that the sample has indeed been generated by a Gaussian. If, however, we do not know the 'correct' model which has generated the data and if there is no reason to believe that the sample has been generated from any particular model then we can ask ourselves whether the use of MLE is appropriate. In previous work (Brown, 1987), the maximum mutual information (MMI) criterion for HMMs has been introduced in order to alleviate problems that may occur when several HMMs are to be designed at the same time in such a way as to maximise the discrimination power of each model. In this work, the speech recognition problem is approached from an information-theoretic perspective. As we do not have any 'correct' model for speech, Brown introduced an alternative method of parameter estimation (MMIE) which is not derived from any assumption of the model. In a nutshell, the goal of MMIE is to derive as much *information* as possible from a sequence of observation.

In contrast, the motivation of our approach is not to maximise the mutual information between observation sequence and a *complete* set of models $\Theta = (\Theta_1, \Theta_2, \dots)$, instead we are trying to estimate the parameters of an HMM that maximises the mutual information between two random variables.

In order to measure the amount of information about the random variable Y we may expect to obtain by observing the random variable X, Shannon introduced the concept of mutual information $I(X, Y)$ between X and Y (Shannon, 1948).

$$I(X, Y) = H(X) + H(Y) - H(X, Y) \tag{3.4}$$

where $H(X) = -E_X[\log P(X)]$ is the entropy of the random variable X. As the joint probability $P(X, Y)$ can be rewritten as:

$$P(X, Y) = P(Y \mid X)P(X) \tag{3.5}$$

we have

$$I(X, Y) = H(Y) - H(Y \mid X) \tag{3.6}$$

In our problem, the goal is to maximise the information with respect to Y we may get by observing a delayed time series X^d in a non-stationary environment. The first problem is to model the densities of the random variables and then estimate the parameters of the model in order to maximise Equation 3.6. In this approach, for each value d, we model the probability $P(Y)$ and the conditional probability $P(Y \mid X^d)$ with two different HMMs, say Θ_Y and $\Theta^d_{Y \mid X}$. We notice however that the first term of Equation 3.6 does not depend on d and can be discarded in the optimisation procedure. Indeed, for two different values of d, i.e. for two different time series X^{d_1} and X^{d_2}, the entropy of Y does not affect the change in the mutual information $I(X^{d_1}, Y) - I(X^{d_2}, Y)$. Thus, maximising the conditional distribution $P(Y \mid X^d)$ is equivalent to maximising the mutual information between Y and X^d.

$$\arg\max_d I(X^d, Y) = \arg\max_d \log P(Y \mid X^d) \tag{3.7}$$

Comparing Equation 3.7 to Equation 3.1, we see that MLE and MMIE differ in the objective function. In MLE, we are interested in estimating parameters that maximise the joint probability. The MMIE approach leads to maximising the conditional probability.

As we saw in Section 3.1, the Baum-Welch algorithm is a hill-climbing algorithm for maximum likelihood estimation which does not require the model gradient. Unfortunately, no such method is known for MMI estimation and we must therefore resort to the use of traditional optimisation techniques with the objective function $E = -\log P(Y \mid X, \Theta)$. Suppose that the sample vector $O_t = (X_t, Y_t)$ is a sample from a 2-dimensional Gaussian, with mean $\mu = (\mu_1, \mu_2)$ and covariance matrix

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} \tag{3.8}$$

then the conditional density $f(y_t \mid x_t)$ is Gaussian and is given by

$$f(y_t \mid x_t) \sim \mathcal{N}\left(\mu_2 + \frac{\sigma_{12}}{\sigma_1^2}(x_t - \mu_1), \sigma_2^2 - \frac{\sigma_{21}}{\sigma_1^2}\right) \tag{3.9}$$

In the case of an HMM with Gaussian observation densities, each hidden state i of the HMM is parameterised by a mean and a covariance matrix. The conditional probability density $b_i(y_t \mid x_t)$ is then given by Equation 3.9. The derivatives of the density function with respect to the parameters of each hidden state can be easily found. The derivatives of the objective function E with respect to the parameters $\Theta = \{A, B, \Pi\}$ are obtained using the forward and backward variables of the Baum-Welch algorithm. They can then be used either in a simple gradient descent algorithm or a nonlinear optimisation algorithm like conjugate gradients (Press et al., 1992), which uses the gradient of the objective function. Such methods require a line search which involves many evaluations of the objective function. Evaluating the objective function requires the computation of the forward variables, whereas the derivative of the function needs both forward and backward variables. Each forward and backward recursion requires on the order of N^2T calculations, where N is the number of hidden states and T is the length of the sequence. This can lead to a computationally expensive algorithm, especially if the HMM contains a large number of parameters. This is not a big issue for the problem we are interested in as the models we consider are relatively small.

It should also be noticed that contrary to the MLE approach where the constraints of the model[4] are satisfied at each iteration (see Appendix A), the implementation of the MMIE approach requires a re-parameterisation of the model to ensure that the constraints are satisfied. Appendix B reviews the derivation and the technical points of the MMI procedure.

4 Results

In this section, we first present the results of the maximum likelihood approach on synthetic data in order to demonstrate the HMM approach and show that it is more general than the classic standard linear techniques for lag detection. We then introduce the drilling process and show how our methods can be used to estimate a crucial parameter in the oil industry.

As mentioned in the previous sections, the lag detection problem can be formulated as a synchronisation problem where the goal is to estimate the value δ that minimises a certain cost function. Both MLE or MMIE approaches lead to a similar procedure[5]. As mentioned in Section 3, the choice of initial conditions is important for the EM algorithm since the re-estimation equations give values of the HMM parameters which correspond to a local maximum of the likelihood. For this reason, we prefer to initialise the model carefully rather than with a simple random initialisation. In this work, we have implemented a version of the segmental K-means procedure (Rabiner, 1989). Starting from an initial estimate of the HMM parameters, the observation sequence is segmented into states. This segmentation can be carried out with the Viterbi algorithm. The parameters of each state are then updated via a K-means clustering procedure. The resulting model is compared to the previous model and the overall initialisation procedure is repeated until convergence of the parameters.

4.1 Synthetic Data

In order to illustrate our approach, we consider the following problem: we generated a synthetic two dimensional sequence of observations $o^* = (x_t, y_t)_{t=1\ldots T}$ from a two state continuous HMM denoted by $\Theta^* = \{A^*, B^*, \Pi^*\}$, in this case $\delta = 0$. We then trained two state continuous HMMs Θ^d with delayed sequences $o^d = (x_{t-d}, y_t)_{t=1\ldots T}, -D \leq d \leq D$ using the MLE approach (the MMIE

[4] At each iteration, we must ensure that the new estimates a_{ij} can be interpreted as transition probabilities. Another constraint concerns the output probabilities $b_i(o_t)$: if we consider a mixture of Gaussians for example, a symmetric positive definite covariance matrix is required at each step.

[5] We denote by $\mathcal{L}^d$ either the joint probability or the conditional probability depending on the approach.

approach led to the same results for this simple example). We choose a transition matrix A^* which allows balanced transitions from one state to another, i.e. $a^*_{ij} \approx a^*_{ji}$. The output probability density associated with each state is a Gaussian with a diagonal covariance matrix and does not affect the simulations significantly.

The results are shown in Figure 3. It can be seen that the cross-correlogram is not capable of detecting any relationship between the two time series, even though they are generated by the same HMM. This is simply because we chose a diagonal covariance matrix, which means that there is no linear relation between the two time series. On the other hand, plotting the log-likelihood of each model Θ^d against d shows a significant peak for $d = 0$, which corresponds to Θ^*, i.e. the true model which was used to generate the time series. The sharpness of the peak shows that we cannot use the shape of the likelihood curve to search for an optimal value of d, as the curve is practically flat for $|d| > 1$;

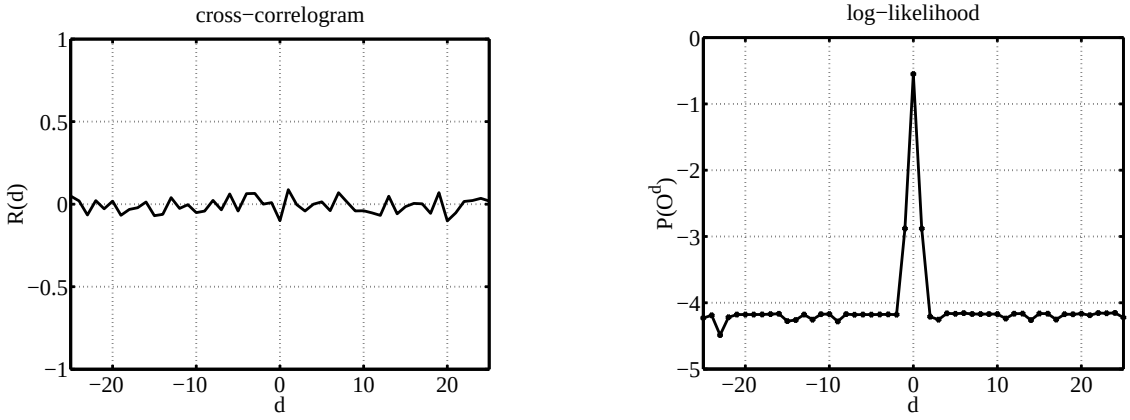

Figure 3: The left hand plot shows the cross-correlogram of the two time series generated by the 2 state continuous HMM Θ^*. The right hand plot shows the log-likelihood $\mathcal{L}^d$ against the lag d when training HMMs with MLE approach.

5 Drilling Data Analysis

One significant problem in exploration drilling is that of ensuring the drilling debris is effectively removed from the bore; this is known as the 'hole cleaning' problem (Guild et al., 1995). At present, no equipment exists to act as a monitor of hole cleaning status. In the case of vertical wells, an adequate velocity in the mud circulation is generally sufficient to guarantee that most debris are brought to the surface. The problem is more complicated when drilling deviated wells[6] since gravity settlement can occur. The gradual build up of low gravity solids increases the torque required to turn the drill string. In extreme cases, the drill pipe may get stuck or even fracture off. Retrieving a stuck pipe is a difficult and expensive operation: if the pipe breaks and cannot be recovered, the well may even be abandoned. The time it takes for the cuttings to come to the surface is called the *lag for return* and is a crucial parameter in early stuck pipe detection and modelling the drilling process. Indeed this parameter depends not only on the depth of the hole, but also on the geology of the surrounding rock formation and the rheology[7] of the mud. The current algorithms used on rigs to compute the lag for return are based on fluid mechanics but are believed to have an accuracy in the order of several minutes, mainly because of assumptions on the nature of the fluid and the flow. Fluids are divided into two general classes, namely the *Newtonian* and *non-Newtonian* fluids[8]. Generally speaking, there are two types of flow: laminar flow and

[6] Whenever possible, wells are drilled vertically, but sometimes, especially offshore, it is necessary to deviate from vertical in order to reach a wide spread of targets from a single platform.

[7] The term rheology defines the chemical properties of the mud: the most important rheological properties of mud are its plastic viscosity, its yield point and its gel strength (Rabia, 1985).

[8] A Newtonian Fluid is defined by a constant viscosity, which is only influenced by changes in temperature and pressure.

turbulent flow[9]. During drilling operations, numerical models take into account non-Newtonian nature of drilling mud, but do assume that the flow is laminar, because of our poor understanding of downhole conditions.

Recently a new device, capable of detecting fine particulate solids in drilling fluids, has been developed by Thule Rigtech Ltd. (Thule Rigtech, 1995). Our aim is to use this device to monitor trends in the volumes of drilled solids in order to obtain a better picture of downhole conditions with regard to drilled solids than has ever been possible before.

As all the data are collected on the surface, if δ represents the lag for return, then Y_t, which is the quantity of low gravity solids measured at time t, is effectively the amount of solids that has been generated by the bit at time $t - \delta$. Thus assuming that Y_t is related to other drilling parameters $X_{t-\delta}$, we propose to compute the lag of return by using hidden Markov models and the procedure described in Section 3. Our motivation is essentially based on the fact that the drilling process is actually a process that manifests a sequentially changing behaviour: the properties of the process are usually held pretty steady, except for minor fluctuations, for a certain period of time, and then, at certain instances, change to another set of properties (caused by action of the drilling engineers predicting a problem, change of geology, etc). The opportunity for more efficient modelling can be exploited if we can first identify these periods of rather steady behaviour, and then are willing to assume that the temporal variations within each of these steady periods are stochastic. A more efficient representation may then be obtained by using a common short time model for each of the steady, or well-behaved parts of the model, along with some characterisation of how one such period evolves to the next. Even if the process cannot be considered as stationary, there are strong reasons to believe that δ remains relatively constant over a 2 hour time scale. Typically X_t represents one relevant drilling parameter (although we have also considered models with more than one parameter): for instance, the pressure of the circulating fluid inside the pipe or the torque of the pipe. The total force applied on the drilling system in order to hold the drill pipe in the rig (hook load) and the rate of progress[10], are other important parameters. To illustrate the idea of our approach, we give the results of our simulations for 4 different data sets.

5.0.1 Normal drilling conditions

This data set represents the 'normal' drilling conditions as no special event was identified by the drilling engineers. It is very difficult to estimate the value of δ by a simple visual inspection of the time series. The data set contains 450 data points and represents a period of 4 hours of drilling. The numerical models suggest a value of 31 min for the lag for return. We have trained different HMMs with different numbers of hidden states and considered continuous HMMs where each state is associated with one Gaussian. Figure 4 plots the results of both approaches. The top figure plots the cross-correlogram between the quantity of low gravity solids and two important drilling parameters, namely the pipe pressure and the pipe hook load. No correlation significantly different from zero can be detected. The MLE approach suggests a value between 36 and 40 min whereas the MMIE approach is more confident and suggests a sharper peak at 37 min, which is statistically significant according to the two standard deviation error bars. The results have been obtained by training 3 state HMMs using the pipe hook load for the time series X_t. Simulations with other drilling parameters did not give interesting results. As we shall see later, depending on the drilling conditions, selecting the right variable X_t is a key problem with this approach. However, when using other data sets corresponding to normal drilling conditions, we have found that hook load always gives good results.

[9] In turbulent flow, the flow pattern is random in both time and place. The chaotic and disordered motion of fluid particles results in two components of velocity: a longitudinal and a transverse component.

[10] The rate of progress (ROP) or drilling rate simply indicates how fast we are drilling in ft/hr.

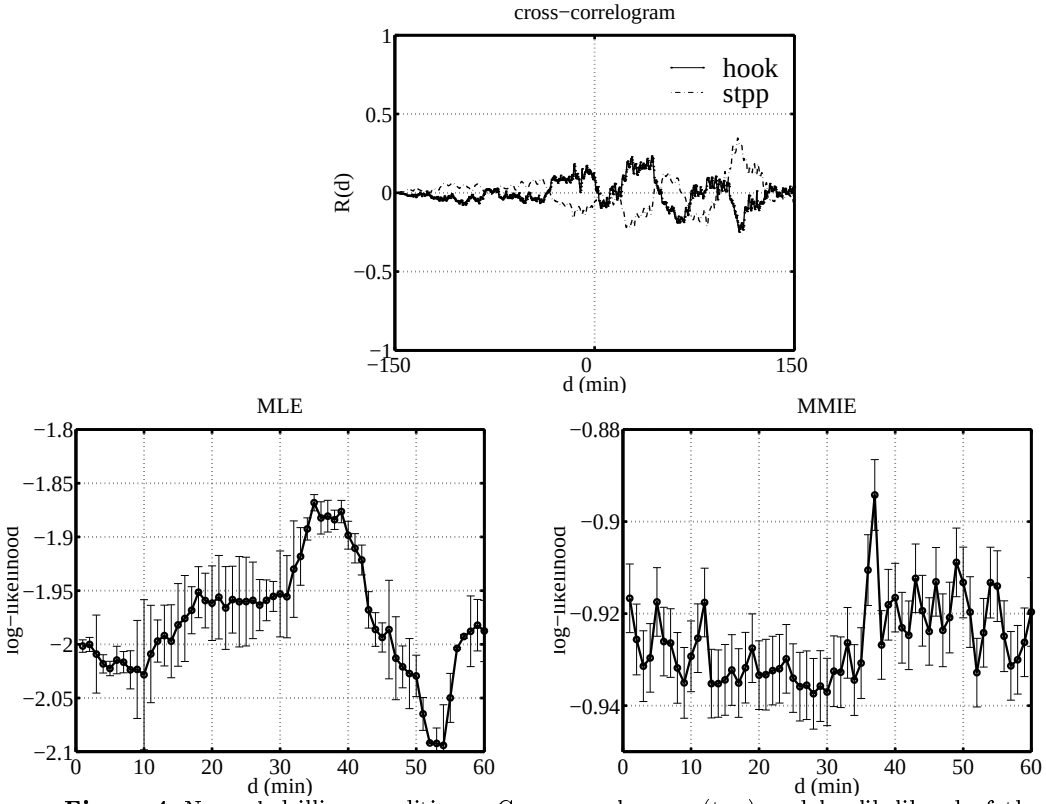

Figure 4: Normal drilling conditions: Cross-correlogram (top) and log-likelihood of the observation sequences O^d given the model HMM^d for MLE (left) and MMI (right) approaches. For each value of d, 100 models with different initial parameters have been trained. The figures plot the average and the error bars computed around the global minimum.

5.0.2 Formation change

When exploring different types of rock structure, it is sometimes possible to estimate the lag for return by visually monitoring relevant time series where a transition occurs between two geological formations. Figure 5 plots the rate of progress and the quantity of low gravity solids for a specific event. At 08h04 the formation changed from soft to hard rock, which can be easily seen in the first plot where the rate of progress decreases suddenly. The second figure plots the quantity of solids and shows a significant regime transition at 08h27: the amount of particles decreases as well. Note the noisy signal for low gravity solids. For this day, the numerical models based on fluid mechanics suggested a value of δ of 43 min, which seems to be wrong as visual inspection of the graphs gives a value of 23 min.

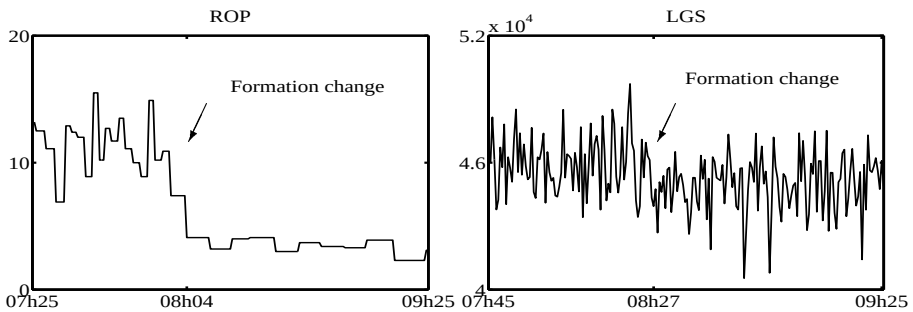

Figure 5: Evolution of the rate of progress and the quantity of low gravity solids for a particular event in drilling operations. The formation changes from soft to hard rock at a certain time. The lag of return is visually identifiable in such a situation.

Figure 6 shows our results: 2 state continuous HMMs have been trained with $O^d = (X_{t-d}, Y_t)_{t=1\ldots T}$

where X_t and Y_t denote respectively the quantity of low gravity solids and the pressure inside the drill pipe[11]. Again, the cross-correlogram of the two time series shows clearly that this approach does not suggest any value for δ. The second figure plots the results obtained with the maximum likelihood approach. A reasonably significant peak around 23 min can be seen. Moreover, by applying the Viterbi algorithm to the sequence $O^d = (X_{t-d}, Y_t)$ with $d = 23$ min, we can recover the most probable sequence of hidden states. In this sequence, the HMM stays in one state before the event, jumps to the other state precisely when the formation change occurs and then remains in the second state. Such a clear sequence could not be obtained with other HMM^d trained with a different value for d, confirming the computed value for δ. The third figure plots the results obtained with the maximum mutual information approach. Again, this method suggests a sharper peak and confirms the value of $\delta = 23$ min.

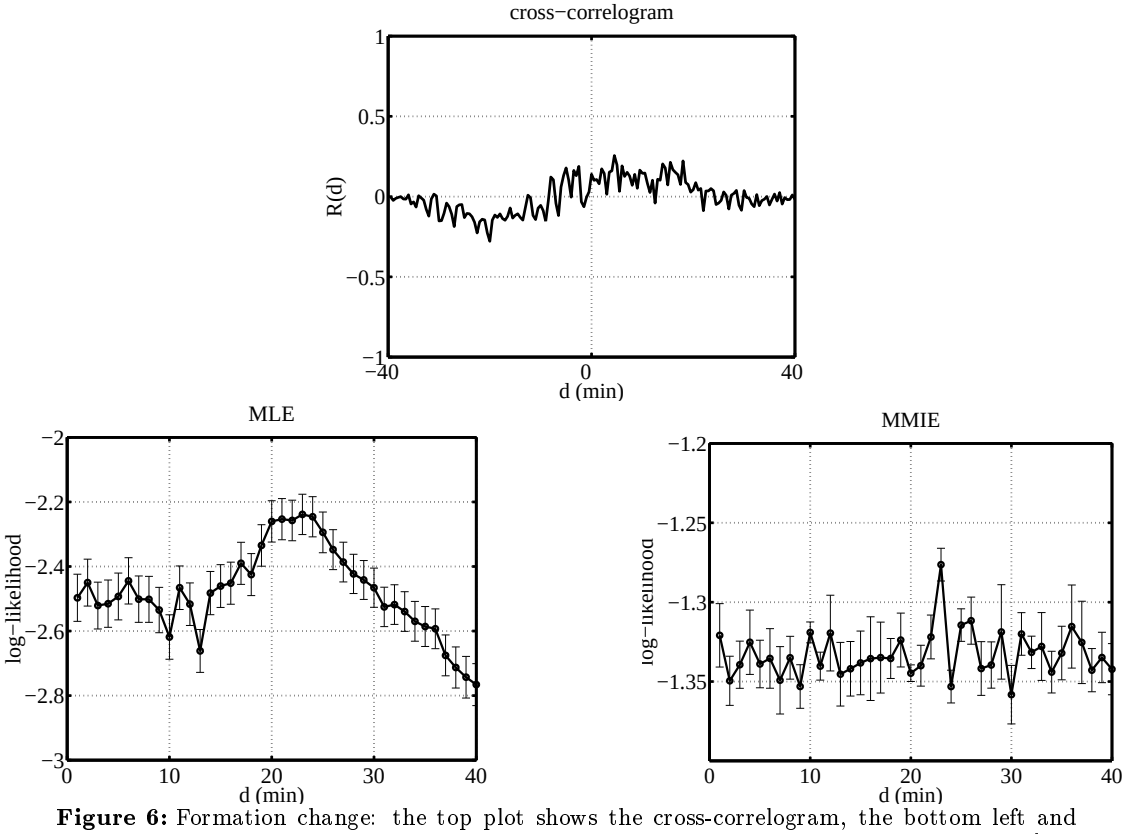

Figure 6: Formation change: the top plot shows the cross-correlogram, the bottom left and right plots show the average log-likelihood of the observation sequences $O^d = (X_{t-d}, Y_t)_{t=0\ldots T}, 0 \leq d \leq D$ given the model HMM^d when trained with MLE and MMIE approaches. Both methods suggest a value around 23 min.

To end with this section, Table 1 reports the results of our simulations on different data sets and shows how they differ from the ones obtained from the fluid mechanics model.

Dataset	Mud report	MLE	MMIE
A	68 min	72-75 min	74 min
B	31 min	36-40 min	37 min
C	36 min	40-43 min	42 min
D	43 min	22-25 min	23 min

Table 1: Our results compared to the ones reported in the mud report.

[11] The hook load parameter does not give good results for this data set.

6 Conclusions

In this paper, we have shown how hidden Markov models can be used to identify relationships between variables. We proposed two approaches for training the models for this applications. The first one uses the usual maximum likelihood estimation and consists of maximising the joint probability of the two variables. The second approach uses a novel mutual information estimation approach, which maximises the conditional probability of one variable with respect to the other. The proposed methods were tested on data from a real-world process and clearly demonstrated their ability to outperform traditional cross-correlation methods. We focussed on the estimation of a crucial parameter of the drilling process and obtained better results than the numerical models based on fluid mechanics used in the oil industry. When comparing the MLE and the MMIE approaches, the latter seems consistently to estimate the lag more precisely. However, as indicated in Section 3.2, the MMIE implementation is time consuming and we have noticed that it is more sensitive to initialisation than the MLE (for a drilling dataset, a typical MMIE procedure needs roughly 20 times more forward-backward passes than MLE).

Concerning the estimation of the lag for return for the oil drilling data, as mentioned above, we have noticed difficulties associated with the variable selection, as no general characterisation of the most relevant drilling parameters is available at this stage. One way to tackle this problem would be to use the qualitative relationships given by numerical models.

We hope to extend this work to provide an on-line estimate of the lag for return in order to track the amount of solids coming up to the surface for the purposes of early stuck pipe detection. The current algorithms are batch algorithms. By considering only a forward pass in the forward-backward algorithm, it is indeed possible to derive sequential learning algorithms.

7 Acknowledgement

This work is funded by EPSRC (grant GR/L08632), Shell and Agip. The authors would like to thank Mike Affleck of Thule Rigtech Ltd. for helping in collecting and interpreting the data. Mehdi Azzouzi is grateful to Francesco Vivarelli for helpful discussions.

A The Baum-Welch algorithm

The Baum-Welch algorithm is an EM algorithm specifically for time series. The first steps are two passes through the time series: one 'forwards' and the other 'backwards'.

The E-step
Using notations of (Rabiner, 1989), we introduce the *forward* variables $\alpha_t(i)$ for each state i:

$$\alpha_t(i) = P(O_1^t, S_t = q_i \mid \Theta) \tag{A.1}$$

We obtain the forward recursion

$$\alpha_1(i) = \pi_i b_i(o_1) \qquad 1 \leq i \leq N \tag{A.2a}$$

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(o_{t+1}) \qquad 1 \leq t \leq T-1 \tag{A.2b}$$

At the end of the forward recursion, the likelihood of the observation sequence is given by

$$P(O_1^T \mid \Theta) = \sum_{i=1}^{N} \alpha_T(i) \tag{A.3}$$

In a similar way, we can define the *backward* variable $\beta_t(i)$:

$$\beta_t(i) = P(O_{t+1}^T \mid S_t = q_i, \Theta) \tag{A.4}$$

and solve inductively:

$$\beta_T(i) = 1 \qquad 1 \leq i \leq N \tag{A.5a}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \qquad t = T-1, \ldots, 1 \tag{A.5b}$$

Combining Equation A.1 and Equation A.4 gives another way to compute the likelihood for any value of t:

$$P(O_1^T \mid \Theta) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{A.6}$$

The posterior probability of the state at time t is obtained by multiplying α_t and β_t and normalising:

$$\gamma_t(i) = P(S_t = q_i \mid O_1^T, \Theta) = \frac{\alpha_t(i)\beta_t(i)}{P(O_1^T \mid \Theta)} \tag{A.7}$$

We define also the posterior probability of being in state q_i at time t and state q_j at time $t+1$

$$\xi_t(i,j) = P(S_t = q_i, S_{t+1} = q_j \mid O_1^T, \Theta) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O_1^T \mid \Theta)} \tag{A.8}$$

It is easy to verify that

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{A.9}$$

We are now able to compute the expected log-likelihood and complete the E-step (Equation 3.2). We represent the variable S_t as a N-dimensional vector $S_t = [S_t^1, \ldots, S_t^N]$, where $S_t^i \in \{0,1\}$. For example $s_1 = [0,1,0,\ldots,0]$ means that at time $t = 1$, the system is in state 2. Using this

representation, each term of the joint probability $P(O_1^T, S_1^T)$ (Equation 2.1) can be rewritten as:

$$\begin{aligned}
\log P(S_1) &= \log \prod_{i=1}^{N} \pi_i^{S_1^i} \\
\log P(S_t \mid S_{t-1}) &= \log \prod_{i=1}^{N} \prod_{j=1}^{N} (a_{ij})^{S_{t-1}^i S_t^j} \\
\log P(O_t \mid S_t) &= \log \prod_{i=1}^{N} [P(O_t \mid S_t^i)]^{S_t^i}
\end{aligned}$$

where, again, we represent the initial state probabilities as a vector $\pi = [\pi_1, \ldots, \pi_N]$. This allows us to write each term of the logarithm of Equation 2.1 as:

$$\log P(O_1^T, S_1^T) = \sum_{i=1}^{N} S_1^i \log \pi_i + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} S_{t-1}^i S_t^j \log a_{ij} + \sum_{t=1}^{T} \sum_{i=1}^{N} S_t^i \log P(O_t \mid S_t^i) \tag{A.10}$$

The function to evaluate in the E-step is the expectation of Equation A.10 under the posterior distribution of the hidden states given the observation sequence:

$$Q(\Theta^{new}, \Theta^{old}) = E[\log P(O_1^T, S_1^T \mid \Theta^{new}) \mid P(S_1^T \mid O_1^T, \Theta^{old})] \tag{A.11}$$

This expectation is a function of $E[S_t^i \mid P(S_1^T \mid O_1^T, \Theta^{old})]$ and $E[S_{t-1}^i S_t^j \mid P(S_1^T \mid O_1^T, \Theta^{old})]$. The first term represents the probability of being in state i at time t given the current parameters and the observation sequence. This is therefore equal to $\gamma_t(i)$. The second term is the probability of being in state i at time $t-1$ and in state j at time t: this is $\xi_{t-1}(i,j)$:

$$Q(\Theta^{new}, \Theta^{old}) = \sum_{i=1}^{N} \gamma_1(i) \log \pi_i + \sum_{t=1}^{T-1} \sum_{i=1}^{N} \sum_{j=1}^{N} \xi_t(i,j) \log a_{ij} + \sum_{t=1}^{T} \sum_{i=1}^{N} \gamma_t(i) \log P(O_t \mid s_t^i) \tag{A.12}$$

The M-step

A typical initial probability π_i^{new} is the solution of

$$\frac{\partial}{\partial \pi_i^{new}} \left\{ Q(\Theta^{new}, \Theta^{old}) - \lambda \left(\sum_{j=1}^{N} \pi_i^{new} - 1 \right) \right\} = 0 \tag{A.13}$$

$$\tag{A.14}$$

where λ is the Lagrange multiplier.

$$\pi_i^{new} = \gamma_1(i) \tag{A.15}$$

Similarly, for a typical transition a_{ij}^{new}, we get the re-estimation formula (using Equation A.9)

$$a_{ij}^{new} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{A.16}$$

Depending on the parametrisation of the output observation distribution $b_i(o_t)$, re-estimates can be easily found. We give here the re-estimation equations for a full covariance Gaussian density function $b_i(o_t) \sim \mathcal{N}(\mu_i, \Sigma_i)$:

$$\mu_i^{new} = \frac{\sum_{t=1}^{T} \gamma_t(i) o_t}{\sum_{t=1}^{T} \gamma_t(i)} \tag{A.17a}$$

$$\Sigma_i^{new} = \frac{\sum_{t=1}^{T} \gamma_t(i) (o_t - \mu_i^{new})(o_t - \mu_i^{new})'}{\sum_{t=1}^{T} \gamma_t(i)} \tag{A.17b}$$

B Maximum Mutual Information Estimation

In this appendix, we derive the equations used in the maximum mutual information estimation procedure. For simplicity, we consider a 2-dimensional Gaussian distribution $f(x_t, y_t) \sim \mathcal{N}(\mu, \Sigma)$ with mean and covariance matrix

$$\mu = (\mu_1, \mu_2) \tag{B.1a}$$

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{pmatrix} \tag{B.1b}$$

In this case, the conditional density is also Gaussian

$$f(y_t \mid x_t) \sim \mathcal{N}\left(\mu_2 + \frac{\sigma_{12}}{\sigma_1^2}(x_t - \mu_1), \sigma_2^2 - \frac{\sigma_{12}}{\sigma_1^2}\right) \tag{B.2}$$

Setting

$$m = \mu_2 + \frac{\sigma_{12}}{\sigma_{11}}(x_t - \mu_1) \tag{B.3}$$

$$\sigma = \sigma_2^2 - \frac{\sigma_{12}}{\sigma_1^2} \tag{B.4}$$

we have, by taking the logarithm of Equation B.2

$$\log f(y_t \mid x_t) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\sigma - \frac{1}{2}\frac{(y-m)^2}{\sigma} \tag{B.5}$$

It is straightforward to calculate the derivatives with respect to m and σ

$$\frac{\partial}{\partial m}\log f(y_t \mid x_t) = \frac{y-m}{\sigma} \tag{B.6}$$

$$\frac{\partial}{\partial \sigma}\log f(y_t \mid x_t) = -\frac{1}{2\sigma}\left(1 - \frac{(y-m)^2}{\sigma}\right) \tag{B.7}$$

Derivatives of the log-likelihood
Given an observation sequence O_1^T, we need to compute the derivatives of the log-likelihood with respect to θ, a component of Θ

$$\frac{\partial}{\partial\theta}\log p(Y_1^T \mid X_1^T, \Theta) = \frac{1}{p(Y_1^T \mid X_1^T, \Theta)}\frac{\partial}{\partial\theta}p(Y_1^T \mid X_1^T, \Theta) \tag{B.8}$$

For clarity, we will omit the term X_1^T, as for each hidden state i we model the conditional probability density$b_i(o_t) = f(y_t \mid x_t)$. We will also denote the likelihood by

$$\mathcal{L} = \log p(Y_1^T \mid X_1^T, \Theta) \tag{B.9}$$

Using the chain rule, the gradient with respect to a particular transition probability a_{ij} is given by

$$\frac{\partial\mathcal{L}}{\partial a_{ij}} = \sum_{t=1}^{T}\frac{\partial\mathcal{L}}{\partial\alpha_t(j)}\frac{\partial\alpha_t(j)}{\partial a_{ij}} \tag{B.10}$$

By differentiating Equation A.6 with respect to $\alpha_t(j)$, we get

$$\frac{\partial\mathcal{L}}{\partial\alpha_t(j)} = \beta_t(j) \tag{B.11}$$

and differentiating Equation A.2b with respect to a_{ij}

$$\frac{\partial\alpha_t(j)}{\partial a_{ij}} = b_j(o_t)\alpha_{t-1}(i) \tag{B.12}$$

Substituting in Equation B.10, we get the result

$$\frac{\partial \log \mathcal{L}}{\partial a_{ij}} = \frac{1}{\mathcal{L}} \sum_{t=1}^{T} \beta_t(j) b_j(o_t) \alpha_{t-1}(i) \tag{B.13}$$

and for an initial probability π_i

$$\frac{\partial \log \mathcal{L}}{\partial \pi_i} = \beta_1(i) b_i(o_1) \tag{B.14}$$

In a similar way, the gradient with respect to a particular observation probability $b_i(o_t)$ is

$$\frac{\partial \mathcal{L}}{\partial b_i(o_t)} = \frac{\partial \mathcal{L}}{\partial \alpha_t(i)} \frac{\partial \alpha_t(i)}{\partial b_i(o_t)} \tag{B.15}$$

By differentiating Equation A.2b with respect to $b_i(o_t)$

$$\frac{\partial \alpha_t(i)}{\partial b_i(o_t)} = \frac{\alpha_t(i)}{b_j(o_t)} \tag{B.16}$$

and substituting in Equation B.15 (using Equation A.6 again)

$$\frac{\partial \log \mathcal{L}}{\partial b_i(o_t)} = \frac{\gamma_t(i)}{b_i(o_t)} \tag{B.17}$$

Parameterisation
For MLE, we incorporated Lagrange multipliers in the M-step in order to ensure constraint satisfaction. For MMIE, we use general unconstrained optimisation algorithms. For instance, in order to ensure that the parameters a_{ij} can be interpreted as transition probabilities, they must satisfy

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{B.18a}$$

$$0 \leq a_{ij} \leq 1 \tag{B.18b}$$

These constraints can be satisfied by choosing a_{ij} to be a *softmax* version of an unconstrained variable w_{ij} (Bridle, 1990)

$$a_{ij} = \frac{e^{w_{ij}}}{\sum_{k=1}^{N} e^{w_{ik}}} \tag{B.19}$$

Differentiating Equation B.19 and using the chain rule, we have

$$\frac{\partial a_{ij}}{\partial w_{ik}} = \delta_{jk} a_{ik} - a_{ij} a_{ik} \tag{B.20}$$

$$\frac{\partial \mathcal{L}}{\partial w_{ij}} = \sum_{k} \frac{\partial \mathcal{L}}{\partial a_{ik}} \frac{\partial a_{ik}}{\partial w_{ij}} \tag{B.21}$$

Substitution in Equation B.13 gives the gradient of the log-likelihood with respect to the new parameters w_{ij}[12].

We must also ensure that the covariance matrix Σ is positive definite and symmetric. This can be done by defining an upper triangular matrix U which represents the Cholesky decomposition of Σ. U must have positive diagonal entries, but upper triangular entries are arbitrary.

$$U = \begin{pmatrix} e^{u_1} & u_3 \\ 0 & e^{u_2} \end{pmatrix} \tag{B.22}$$

So, writing $\Sigma = U^{'} U$, we obtain

$$\Sigma = \begin{pmatrix} e^{2u_1} & u_3 e^{u_3} \\ u_3 e^{u_3} & u_3^2 + e^{2u_2} \end{pmatrix} \tag{B.23}$$

[12] A similar constraint applies also for the initial probabilities π.

Again using the chain rule in Equation B.7, we can easily compute the derivatives of the output probabilities $b_i(o_t)$ with respect to the new parameters and therefore the derivatives of the log-likelihood

$$\frac{\partial \log \mathcal{L}}{\partial \mu_1^i} = -u_3 e^{-u_1} \sum_{t=1}^{T} \gamma_t(i) \frac{(y_t - m^i)}{\sigma^i} \tag{B.24}$$

$$\frac{\partial \log \mathcal{L}}{\partial \mu_2^i} = \sum_{t=1}^{T} \gamma_t(i) \frac{(y_t - m^i)}{\sigma^i} \tag{B.25}$$

$$\frac{\partial \log \mathcal{L}}{\partial u_1^i} = -u_3 e^{-u_1} \sum_{t=1}^{T} \gamma_t(i) \frac{(x_t - \mu_1^i)(y_t - m^i)}{\sigma^i} \tag{B.26}$$

$$\frac{\partial \log \mathcal{L}}{\partial u_2^i} = -\sum_{t=1}^{T} \gamma_t(i)(1 - \frac{(y_t - m^i)^2)}{\sigma^i} \tag{B.27}$$

$$\frac{\partial \log \mathcal{L}}{\partial u_3^i} = e^{-u_1} \sum_{t=1}^{T} \gamma_t(i) \frac{(x_t - \mu_1^i)(y_t - m^i)}{\sigma^i} \tag{B.28}$$

These derivatives can then be used with a nonlinear optimisation algorithm, such as conjugate gradients, in order to minimise the objective function.

References

Baldi, P., Chauvin, Y., Hunkapiller, T., and McClure, M. A. (1993). Hidden Markov Models in molecular biology: New algorithms and application. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo.

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.

Bengio, Y. and Frasconi, P. (1995). An input-output HMM architecture. In Tesauro, G., Touretsky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems*, volume 7, pages 427–234, Cambridge, MA. MIT Press.

Bourlard, H. (1997). State-of-the-art and recent progress in hybrid HMM/ANN speech recognition. In *Proc. of the Intl. Conf. on Artificial Neural Networks*, pages 875–884. Springer (ISBN 3-540-63631-5).

Bridle, J. S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Fogelman, F. and Herault, J., editors, *Neurocomputing: Algorithms, Architectures and Applications*, pages 227–236. New York: Springer-Verlag.

Broomhead, D. S. and King, G. P. (1986). Extracting Qualitative Dynamics from Experimental data. *Physica*, 20D:217–236.

Brown, P. F. (1987). *The Acoustic-Modeling Problem in Automatic Speech Recognition.* PhD thesis, IBM Thomas J. Watson Research Center.

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.

Fraser, A. M. and Dimitriadis, A. (1994). Time Series Prediction: Forecasting the Future and Understanding the Past. In Weigend, A. S. and Gershenfeld, N. A., editors, *Forecasting Probability Densities by Using Hidden Markov Models with Mixed States*, pages 264–281. Addison-Wesley.

Guild, G. J., Wallace, I. M., and Wassenborg, M. J. (1995). Hole Cleaning Program for Extended Reach Wells. *Society of Petroleum Engineers*, pages 425–433. In SPE/IADC Drilling Conference.

Kendall, M. and Ord, J. K. (1990). *Time Series*. Edward Arnold.

Liporace, L. A. (1982). Maximum Likelihood Estimation for Multivariate Observations of Markov Sources. *IEEE Transactions on Information Theory*, 28:729–734.

Poritz, A. B. (1982). Linear Predictive Hidden Markov Models and the Speech Signal. In *Proc. ICASSP*, pages 1291–1294.

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

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. (1992). *Numerical Recipes in C. The Art of Scientific Computing*. Cambridge University Press, second edition.

Rabia, H. (1985). *OilWell Drilling Engineering: Principles and Practice*. Graham & Trotman.

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.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656.

Thule Rigtech (1995). Personal communication.