Predicting Daily Probability Distributions Of S&P500 Returns

Andreas S. Weigend

Shanming Shi

IS-98-23

Predicting Daily Probability Distributions of S&P500 Returns

Andreas S. Weigend
Leonard N. Stern School of Business
New York University

Shanming Shi
J.P. Morgan & Co. Inc.

August 1998

Working Paper Series
Stern #IS-98-23

Replaces WP #IS-97-23

Working Paper IS-98-23 (Information Systems) and **S-98-36** (Salomon Center)
Leonard N. Stern School of Business, New York University
http://www.stern.nyu.edu/~aweigend/Research/Papers/HiddenMarkov

Predicting Daily Probability Distributions of S&P500 Returns

Andreas S. Weigend
Department of Information Systems
Leonard N. Stern School of Business, NYU
44 West Fourth Street, K-MEC 9-74
New York, NY 10012, USA
aweigend@stern.nyu.edu
www.stern.nyu.edu/~aweigend

Shanming Shi
J. P. Morgan & Co. Inc.
60 Wall Street
New York, NY 10260, USA
shi_shanming@jpmorgan.com

Abstract: Most approaches in forecasting merely try to predict the next value of the time series. In contrast, this paper presents a framework to predict the full probability distribution. It is expressed as a mixture model: the dynamics of the individual states is modeled with so-called "experts" (potentially nonlinear neural networks), and the dynamics between the states is modeled using a hidden Markov approach. The full density predictions are obtained by a weighted superposition of the individual densities of each expert. This model class is called "hidden Markov experts".

Results are presented for daily S&P500 data. While the predictive accuracy of the mean does not improve over simpler models, evaluating the prediction of the full density shows a clear out-of-sample improvement both over a simple GARCH(1,1) model (which assumes Gaussian distributed returns) and over a "gated experts" model (which expresses the weighting for each state non-recursively as a function of external inputs). Several interpretations are given: the blending of supervised and unsupervised learning, the discovery of hidden states, the combination of forecasts, the specialization of experts, the removal of outliers, and the persistence of volatility.

Keywords: Forecasting, Density Prediction, Conditional Distribution, Mixture Models, Time Series Analysis, Hidden Markov Models, Gated Experts, Hidden Markov Experts, Model Comparison, Density Evaluation, Computational Finance, Risk Management.

Data: Daily S&P500 (January 1977 to December 1997).

Code: http://www.stern.nyu.edu/~aweigend/Research/Software (in MATLAB).

1 INTRODUCTION

The introduction reviews several approaches to density forecasting in time series, informally introduces the model class of "hidden Markov experts" (HME), discusses methods for density evaluation, and relates HME to previous work. A brief overview of the sections of the paper are given at the end of the introduction.

1.1 Tasks in Time Series Prediction

A time series is a sequence of observations $\mathcal{Y}^T = \{y^t | t = 1, \ldots, T\}$. t enumerates the elements of the sequence,[1] and T is the total number of the observations. Our methodology is to split the entire set of available data into at least two sets. The first part is used to estimate the parameters of the model and is called the training set. The second part of the data is only used at the very end of the entire modeling process to compute performance measures and referred to as the test set.[2] The test set thus serves as the out-of-sample set, since waiting for genuinely new observations would just take too long for daily data.

Many forecasting methods (in particular almost all nonlinear forecasting methods) focus on predicting the next value or *point* of the time series. Such point predictions are appropriate on problems where the signal is only distorted with a small amount of noise, as typically the case in nonlinear dynamics.[3] However, in financial time series, the noise is often larger than the signal itself, requiring methods that predict not just a point but a density. This paper focuses on such density prediction, addresses the problems of a small signal to noise ratio, and includes non-Gaussian density forecasts.

We start by briefly discussing a path through various tasks for prediction.

(1) The first model uses the mean of the training set as point prediction. However, with sufficiently precise experimental resolution, the exact value of the prediction is almost always wrong: probability densities are needed.

(2) Model 1 above can be interpreted as predicting a single Gaussian whose constant variance is that of the training set. This density implicit in the point forecast will be used as the baseline model in the empirical evaluations.

The two possible next steps are (a) to allow the mean to vary, or (b) to allow the variance to vary.

(3a) The predicted mean varies (i.e., it is a function of some inputs, x) but the variance remains constant. The input variables can be other time series (exogenous variables), or they can be lagged values of the series to be predicted (autoregression). The functional mapping from these variables to the output (expected mean) is, in the simplest case, linear. Our framework allows for general nonlinear functions, typically be expressed as neural networks. The parameters of the model can be estimated by minimizing the squared error between the prediction and the observed value. In machine learning, the observed value is called the "target", and an input-output pair is called a "pattern".

[1] To fix a specific example in mind, consider the daily closing prices p_t of Standard and Poor's S&P500 index. This sequence of observations, corresponding to the price of a weighted portfolio of stocks, increases on average over time. The first step in time series prediction (as in any machine learning task) is to find a representation such that the future looks as similar as possible to the past. This is achieved by taking the differences between the logarithm of the prices, $y^t = \log p^t - \log p^{t-1} \approx \left(p^t - p^{t-1}\right) / p^{t-1}$, which is approximately the relative change in price, i.e., the price difference between today and yesterday with respect to yesterday's price. For S&P500 closing prices, the time between observation is spaced evenly (one observation per working day). However, t can just enumerate the elements of an arbitrarily spaced sequence, such as the sequence of trades in transactional data.

[2] Additional sets can be set aside from periods earlier than the test set if there are meta-parameters, such as the number of experts.

[3] This is not a coincidence. Many nonlinear systems can generate so-called chaotic behavior where the time series continues in an "interesting" way forever. This is an important difference to linear systems that die out if they are not driven by noise.

- **(3b)** Rather than varying the mean and keeping the variance constant, Model 3b fixes the mean (to the mean of the training set) but allows for conditional variance. Estimating the parameters becomes more complicated than minimizing a squared error since, in contrast to Model 3a, we here do not have a desired value or target for each pattern. In order to estimate the parameters of the model, a more complicated statistical framework is needed. We use a maximum likelihood approach: The predicted conditional variance is written as a function of the inputs. The coefficients of this function are estimated such that the likelihood of the observed data given the model and the inputs is maximized.

  Model (3b)—mean fixed, variance varying—has important applications in finance. While it is very difficult to predict the once-differenced time series of prices (i.e., returns) better than a constant (Model 1), more accurate predictions of the variance than a constant are often possible. This reflects the well-known property of many financial time series called volatility clustering or volatility persistence: There are time periods with large (positive and negative) returns, which should be predicted with a larger variance, and there are time periods where the market is quiet and the predicted variance should be smaller.[4]

- **(4)** The fourth level of complexity predicts Gaussian densities with conditional means and conditional variances, combining the two degrees of freedom from Models (3a) and (3b).

So far, the form of the density in all the models has been Gaussian. Now we want to generate density predictions that are **non-Gaussian**. To achieve this goal, there are two philosophies: using expansions (e.g., Edgeworth expansion), and using mixture distributions. The *expansion approach* has the advantage of orthogonality. The computation of increasing orders of approximation is sequential; the term of order $(n+1)$ is not effected by terms of order n and below. The corresponding weakness is that the term of order $(n+1)$ can only patch up problems the lower orders have left for it, rather than all $(n+1)$ terms joining together and trying to find a better overall solution.[5]

The *mixture approach* expresses the density $P(y^{t+1})$ as a sum of M distributions:

$$\begin{aligned} & P\left(y^{t+1} \middle| \text{ information set at time } t, \text{model parameters}\right) \\ = & \sum_{j=1}^{M} \gamma_j^{t+1}(\bullet)\; P\left(y^{t+1} | x^{t+1}, \ldots, \text{model parameters}\right) \end{aligned}$$

In the context of nonlinear (e.g., neural network) sub-models for the mixture components, the sub-models are called "experts". This follows the notation introduced for mixture models to the neural network community by Jacobs, Jordan, Nowlan and Hinton (1991) who applied it to a classification problem, see also Jordan and Jacobs (1994).

Several choices need to be made:

- The number of experts cannot be determined directly from the data. We typically choose between three and ten mixture experts, estimate the model, convince ourselves of its performance, and finally analyze the resulting experts. Their interpretation is part of the creativity of the modeling process and is hard to do automatically.

[4] For modeling the volatility of financial returns, expressing the predicted variance as a function of a few lags of returns or squared returns usually does not give good predictions. The dynamics of the underlying structure in the variance of financial returns usually requires *estimates* of the state at the input—the realizations (e.g., squared returns) are very noisy. This is a property of the χ_1^2 distribution (used to approximate volatilities): its mean and standard deviation are of the same size (Timmer and Weigend 1997). To predict volatility well, the model needs to have knowledge about the estimate of the current state. We use a GARCH(1,1) model as a representative in the performance comparison on S&P500 returns in Section 5.

[5] A parallel can be drawn between the two philosophies for modeling densities (expansions vs. mixtures), and the two philosophies for function approximation (polynomials vs. neural networks). Expansions and polynomials are computationally cheap, have incremental updates, and are often amenable for an analytical treatment of convergence properties. Mixtures and neural networks are computationally expensive, since the entire model needs to be re-estimated when the number of components changes.

- The functional form of the individual densities generated by the experts can be any member of the exponential family. The theoretical derivations are kept as general as possible. However, when a specific distribution needs to be chosen (in the computer implementation and the comparisons), we assume these individual distributions to be Gaussian. The idea of mixture models can be traced back to Pearson (1894) who "mined" a data set consisting of measurements of the forehead size of crabs with a mixture of two Gaussians, thus "discovering" two sub-populations.
- γ_j^{t+1} is the weight given to Gaussian j for the prediction for time $(t+1)$, with $\sum_{j=1}^{M} \gamma_j = 1$. The key question is: what should $\gamma(\bullet)$ depend on?

Three possible answers to the last item form the basis for the remaining three model classes.

(5) In the simplest case of an unconditional density, the γ_j's do not depend on anything: a mixture of Gaussians is fitted to the training set and all parameters are constants. The parameters (the mixture weights γ_j, and the means and variances of the individual Gaussians) are estimated in a maximum likelihood framework using the EM algorithm (explained in Section 2.5). This unconditional mixture will be one model in empirical comparisons.

(6) The mixture weights γ_j depend on a set of external variables. Based on the performance of all the experts on each pattern of the training set, a "gate" learns the mapping from its inputs, the exogenous variables, to the γ_j's. This model class is called **gated experts** (GE) (Weigend, Mangeas and Srivastava 1995) and represents a regression model. When used in forecasting, the temporal structure of the time series enters only through the construction of the patterns (the input-output pairs). Note that once these patterns have been generated from the raw data, randomizing the order of the training data has no effect on the resulting model. In the real world, there are time series problems where a regression approach is appropriate. A successful application of this architecture is energy demand forecasting where the inputs into the gate represent cloud coverage, temperature, special tariff days, and other exogenous variables (Weigend et al. 1995). However, there are other time series problems where the nature of the problem requires time to be taken into account in a more fundamental way. One such example is given by the model class of HME.

(7) This model class is called **hidden Markov experts** (HME). It is best described by its underlying assumptions:

- There are several discrete states. Their corresponding functional input-output mapping can be expressed as feedforward networks. These sub-models are called experts.
- At each time step, a single expert is responsible for generating the corresponding observation. We do not know which of the experts actually generated the observation—the probabilities of the experts for each time step need to be estimated from the data.
- Modeling the sequence of the hidden states, we assume that the dynamics of the hidden states can be described by a first order Markov process, i.e., the next state depends only on the current state. This is expressed as a matrix of transition probabilities between the hidden states. We do not know these transition probabilities either; they also must be estimated from the data.

Fortunately, the statistically solid framework of hidden Markov models (Baum and Eagon 1963, Poritz 1988, Rabiner 1989, Rabiner and Juang 1993) provides algorithms to estimate the unknown quantities. We combine this framework with connectionist techniques. We show how we can learn the potentially nonlinear functions of each expert, the parameters of the transition matrix, and the probability vector across the states at each time step.

The distinction made above emphasizes that GE and HME model time in a fundamentally different way. We now focus on the common aspects and consequences thereof. Both model classes share the goal to generate non-Gaussian density forecasts, and both are based on mixture models. The implications that hold for both cases include:

- **Discovering hidden states.** Conventional data analysis, data mining, and knowledge discovery often do not have a clearly defined concept of what it means to "discover" "hidden" "knowledge." This paper clearly defines *hidden states* as the components of the mixture density. The solid statistical basis allows for a principled interpretation in terms of probabilities, enabling the discovery of interesting relations. In the case of predicting financial returns, the hidden states can be related to volatilities. Methodologically, it is important to clarify that the HME approach does not insert knowledge that volatilities are important for characterizing regimes, but it does make statistical assumptions that in turn yield this knowledge.

- **Blending supervised with unsupervised learning.** Approaches to learning from data and computational intelligence are traditionally dichotomized into supervised learning (regression and classification where the desired outcome is known for the training data) and unsupervised learning (clustering where no target is available and the goal is to discover the underlying structure). Both GE and HME combine the strengths of supervised learning with those of unsupervised learning: They build on the advantage of supervised learning that allows for performance evaluation, while providing the flexibility of unsupervised learning that has the advantage of discovering and interpreting hidden states.

- **Combining forecasts.** The idea of combining forecasts, going back to (Bates and Granger 1969), has become increasingly important in areas ranging from applied forecasting (Clemen, Murphy and Winkler 1995) to computational learning theory (Cesa-Bianchi, Freund, Helmbold, Haussler, Schapire and Warmuth 1997). Both GE and HME softly combine the forecasts of the experts. Also, the relative weights for each expert vary at each time step. These weights are the estimates of the posterior probabilities. They reflect the training set performance for similar situations. For GE, the similarity is given through the gate, and for HME through the previous state and the transition matrix.

- **Becoming experts through competition.** In most approaches to forecast combination, the individual models give equal weight to all their training points. GE and HME use *competitive learning*. For each training pattern, all experts compete. If one expert's prediction is better than the predictions of the other experts, it receives a larger share of the data point to update its parameters than the others. It thus learns to improve its predictions in areas where it is already quite good, and learns to ignore areas where some of its competitors are better. For both GE and HME, the experts become true experts and the algorithm learns about their area of applicability. Since we use unconditional variances for each expert, one delineation of the experts is according to the local noise level. Weigend et al. (1995) show that the adaptation of each expert to its (overall) local noise level helps to avoid overfitting. The standard assumption of constant variance often leads to local underfitting in some regions, and to local overfitting in others. When predicting financial returns, the different noise levels correspond to different volatility regimes. Given volatility clustering, this pulls the solution in the same direction as the Markov assumption of staying in a regime rather than switching to another one. In general, the grouping depends both on similar noise levels and on similar functional forms of the experts.

- **Modeling outliers.** Many practical problems in data mining use some heuristic to remove outliers. Given the strong effect outliers have on the model, the specific heuristic can determine the resulting model. As an alternative to removing outliers, robust statistics uses an influence function that downweighs patterns where the observation and prediction are far apart. This practice can be dangerous in risk management, a new area of increasing importance for financial firms. Risk management focuses on rare events and on tails of distributions. Removing outliers or reducing their influence leads to an underestimation of risk that can be detrimental. In contrast, GE and HME model outliers naturally. In our experience, one expert has a relatively large variance compared to the others. Its role is thus to become the "garbage-collector", effectively removing the outliers and "explaining" them much better than all of the other experts whose likelihoods vanish at that point. In turn, the remaining experts have cleaner data which often allows the models to be interpreted more easily.

- **Saving inputs.** When learning from data, one can never be sure that one has the "best" set of inputs. In many cases there is no shortcut to the creative process of arguing for several sets of inputs, building the model, and then evaluating the out-of-sample performance to learn which inputs are important. This paper does not address the problem of input selection. However, once a set of inputs has been decided on, it is often possible to have different experts look at different subsets of the full set of inputs. When linear experts suffice, standard linear theory helps determine the significance of the inputs, which often leads to further reduction. Individual experts can end up with only a fraction of the union of the inputs. This simpler structure also lends clearer interpretations of the individual models. Note that the formalism matches the noise level of each expert to the noise level of its corresponding data. This has been shown to be an important aspect against overfitting of GE.

This part of the introduction showed several angles on the proposed architecture that complement the rigorous evaluation of out-of-sample performance that any data driven modeling has to follow.

1.2 Evaluating Predictions

In the model comparison, GE (Model 6) and HME (Model 7) are chosen to have identical assumptions whenever possible. They have the same number of experts, the same inputs, the same functional form for the experts (e.g., linear or neural network), and, on the output side, the same noise model and degrees of freedom (i.e., expert-specific variances, and expert and input-dependent means). The only difference is the gate. Since many financial time series exhibit volatility clustering, the gate inputs should include some volatility proxy such as exponentially smoothed square returns.

In addition to the comparison between the mixture architectures HME and GE, we also compare them with several simpler architectures: unconditional Gaussian (Model 1), unconditional mixture of Gaussians (Model 5), and a simple GARCH(1,1) model (constant mean but varying variance, Model 3b). The main two questions the empirical evaluation tries to answer are:

- HME vs. GE: Are there hidden states in the market that cannot be observed directly? The answer is positive if the assumption of an underlying hidden Markov process improves predictive accuracy compared to conditioning on exogenous variables.
- HME vs. GARCH: Do HME predicting non-Gaussian densities generate better forecasts than a GARCH model predicting Gaussian densities?

To answer these questions, we compare the out-of-sample performance on a test set, i.e., data from a time period after the end of the training period. No single measure suffices: we use several measures that capture different aspects of the density prediction.

- The first measure focuses on the predicted probability density function (pdf) and computes the average log-likelihood of the test data given the model. This measure, evaluated on test data, allows us to compare the performance of different architectures.
- The second measure focuses on the predicted cumulative density function (cdf). This integral transform method was suggested by Diebold, Gunther and Tay (1998).
- In addition, we also provide the normalized mean squared error. Note that it only evaluates the quality of the point forecast, but does not measure the quality of the density forecast, thus missing the central goal of this paper.

While point forecasting is predominant in the forecasting literature, some studies discuss interval forecasts (Chatfield 1993, Christoffersen 1997) and probability forecasts (Murphy and Winkler

1992, Clemen et al. 1995). As Diebold et al. (1998) point out, the reasons for the relative neglect of density forecasting and evaluation include: uncertainty about the specific distribution, difficulty in evaluation, and the lack of demands from practice. This has changed in the recent past: risk management has become central for financial firms, and trading and pricing models increasingly depend on good density estimates.

1.3 Related Work

A hidden Markov model is a parametric stochastic probability model with which a time series can be generated or analyzed. A hidden Markov model has two interrelated processes: a finite-state Markov chain that cannot be observed, and an emission model associated with each state. The Markov chain is characterized by the matrix of transition probabilities between states. The output probability densities given by the emission model can be characterized along two axes:

1. The output probability densities can be represented non-parametrically or parametrically.
2. The output probability densities may depend on an input (conditional) or they may be a constant for each expert (unconditional).

The mathematical representation that describes the observation probabilities is called the **emission model.** Viewed from the perspective of time series *generation*, the Markov chain generates a sequence of discrete states that we call a path. Based on this path, the emission model generates the probability density for each time step. The specific realization (the "observation") is then generated from this probability density for each time step.

Viewed from the perspective of time series *analysis*, the output probabilities impose a "veil" between the states and the observer of the time series (Ferguson 1980). The task is to lift that veil. The term *hidden* is used because these states cannot be seen directly from the observed data. It is called *Markov* since it assumes that the probability of the next state depends only on the current state and the transition probabilities between the states. Both the states and the observed process can be either discrete or continuous. In state space models, the states and the observations are both continuous (Harvey 1989, Timmer and Weigend 1997). HME use discrete states (corresponding to the experts) and continuous variables (corresponding to the observed time series).

Next, we need to address the question of how to estimate the parameters of the model from the observed sequence. Baum and Eagon (1963) solved this problem for hidden Markov models with discrete observation densities. Baum, Petrie, Soules and Weiss (1970) extend the algorithm to many of the classical distributions. Hidden Markov models have been widely used in speech recognition (Huang, Ariki and Jack 1990). In the neural network community, Bengio and Bengio (1996) proposed the "Input-Output Markov model" which allows for non-constant transition probabilities in addition to nonlinear emission models. The concept of the transition among states can also be used to model the time dependency of regime switching. Poritz (1982) first combined hidden Markov models with linear prediction. Hamilton (1990) introduced switching models to economics and econometrics, spawning a large body of research (Engel and Hamilton 1990, Hansen 1992, Durland and McCurdy 1994, Hamilton 1994, Lahiri and Wang 1994).

Most of these applications focus on point predictions but not on densities. Fraser and Dimitriadis (1994), predicting one of the data sets of the Santa Fe Competition (Weigend and Gershenfeld 1994), used a hidden Markov model and generated non-Gaussian through a Monte Carlo approach (generating many continuations and then essentially presenting a histogram for each time step.) Hamilton and Susmel (1994) proposed an approach to model the conditional variances within Markov switching framework, where they combined the regime switching process with an autoregressive conditional heteroskedasticity (ARCH) model by allowing the parameters of the ARCH process to come from different regimes. Gray (1996) proposed a more comprehensive method to nest the generalized ARCH (GARCH) model into regime switching model. However, these two models are limited to the first and second conditional moment of the distribution.

None of these approaches focused on the prediction and evaluation of more general densities. We emphasize the fact that the Markov switching models by their nature of being mixture models generate densities, and that these densities should be evaluated with appropriate measures. Furthermore, we allow for nonlinear experts.

This paper is organized as follows: Section 2 explains the notation, describes the likelihood function, and illustrates the Expectation Maximization (EM) algorithm used in HME. Section 3 explains how to generate density predictions using HME and describes methods to evaluate the density. Section 4 shows what can be learned from computer generated data for the HME approach to density forecasting. Section 5 presents the empirical results on comparing HME with GE, GARCH, an unconditional mixture, and an unconditional Gaussian for the daily density forecasts of S&P500 returns. Some conclusions are drawn in Section 6.

2 THE ASSUMPTIONS AND THE ALGORITHM

2.1 Notation

1. **Observations.** $\mathcal{Y}^T = \{y^t | t = 1, ..., T\}$ refers to the observed time series data. T is the number of the observations and t is the time index. Similarly, $\mathcal{X}^T = \{x^t | t = 1, ..., T\}$ represents the *input* to the emission model. x^t itself can be a vector or a scalar. In the example of auto-regression, x^t is given by the previous d values, $x^t = \{y^{t-1}, y^{t-2}, ..., y^{t-d}\}$, where d is the dimension of the input. x^t can also consist of exogenous variables.

2. **States.** $\mathcal{S} = \{1, 2, \ldots, j, \ldots, M\}$ denotes the state. M is the number of states in the model and j refers to a specific state.[6] The analysis of the model usually provides interpretations for the states in terms of physical significance or economic meaning such as relations to market sentiment, growth, recession, interest rates or volatility.

3. **Transition probabilities.** a_{ij} is the transition probability of switching from state i to j,

$$\mathbf{A} = \{a_{ij}, \quad i, j \leq M, \quad a_{ij} = P(s^{t+1} = j | s^t = i)\}$$

   where $a_{ij} \geqslant 0$, $\sum_j a_{ij} = 1$, and s^t describes the state at time t.

4. **Emission probabilities.** b_j^t is the probability of observing y^t given the state and the model. In GE and HME this probability depends on the inputs x^t into the experts at time t through the conditional mean

$$\mathbf{B} = \{b_j^t, \quad j \leq M, \quad t \leq T, \quad b_j^t = P(y^t | s^t = j, x^t)\} \quad .$$

5. **Initial probabilities of each state.** $\Pi = \{\pi_i, i = 1 \ldots M\}$, where the probabilities have to sum to unity, $\sum_{i=1}^{M} \pi_i = 1$.

For convenience, $\theta = \{\mathbf{A}, \mathbf{B}, \Pi\}$ denotes the entire set of parameters of the model. The emission probability can thus be written as $P(y^t | s^t, x^t, \theta)$.

2.2 The Likelihood Function

To define the likelihood function, we impose the constraint that the probability of the current state depends only on the previous state:

$$P(s^t | s^{t-1}, s^{t-2}, ..., s^1, \mathcal{X}^{t-1}, \mathcal{Y}^{t-1}) = P(s^t | s^{t-1}) \quad . \tag{1}$$

[6] While theoretical approaches such as the Minimum Message Length principle (Wallace and Boulton 1968) can give an optimal number of states under certain assumptions (Baxter 1996), we take a more pragmatic approach of building models for different M and evaluating the quantity that the modeler is truly interested in. For example, trading models that use the predicted densities as input are often evaluated with a risk-adjusted measure (Choey and Weigend 1997).

With q^T denoting a specific sequence or path of states from $t = 1$ to T, this first-order Markov assumption enables us to write the probability of path $q^T = (s^1, s^2, \ldots, s^T)$ as

$$\begin{aligned} P(q^T) &= P(s^T, s^{T-1}, \ldots, s^t, \ldots, s^1) \\ &= P(s^1) \prod_{t=2}^{T} P(s^t|s^{t-1}) \quad . \end{aligned} \tag{2}$$

Given the current input x^t and the previous state s^{t-1}, earlier values of s and y are irrelevant,

$$P(y^t, s^t|q^{t-1}, \mathcal{X}^{t-1}, \mathcal{Y}^{t-1}) = P(y^t, s^t|s^{t-1}, x^t) \quad . \tag{3}$$

With Eq. 1 this expression can be transformed in the following way:

$$P(y^t, s^t|s^{t-1}, x^t) = P(y^t|s^t, x^t)P(s^t|s^{t-1}) \quad . \tag{4}$$

The central problem of hidden Markov models is to find the entire set of parameters of the model. Using Eq. 3 and Eq. 4, the likelihood $P(\mathcal{Y}^T|\theta)$ is then given as

$$\begin{aligned} P(\mathcal{Y}^T|\theta) &= \sum_{q^T} P(\mathcal{Y}^T, q^T|\theta) \\ &= \sum_{q^T} P(y^T, s^T|q^{T-1}, \mathcal{Y}^{T-1}, \theta) P(\mathcal{Y}^{T-1}, q^{T-1}|\theta) && \text{conditional probability} \\ &= \sum_{q^T} P(y^T, s^T|s^{T-1}, x^T, \theta) P(\mathcal{Y}^{T-1}, q^{T-1}|\theta) && \text{using Eq. 3} \\ &= \sum_{q^T} P(y^T|s^T, x^T, \theta) P(s^T|s^{T-1}) P(\mathcal{Y}^{T-1}, q^{T-1}|\theta) && \text{using Eq. 4} \\ &= \sum_{q^T} \underbrace{P(y^1|s^1, x^1, \theta)}_{b^1} \underbrace{P(s^1)}_{\text{initial}} \prod_{t=2}^{T} \underbrace{P(y^t|s^t, x^T, \theta)}_{=:b_j^t} \underbrace{P(s^t|s^{t-1})}_{=:a_{ij}} \quad . \end{aligned} \tag{5}$$

To obtain the probability $P(\mathcal{Y}^T|\theta)$, two probabilities need to be estimated. First, the emission probability given the current state, $P(y^t|s^t, x^T, \theta)$; it varies at each time step. Second, the transition probability $P(s^t|s^{t-1})$; it is a parameter of the model.

The product $b_j^t a_{ij}$ is at the heart of the hidden Markov framework. If there was no Markov assumption, the second term a_{ij} was absent, and the observation at time t would be attributed to state j with probability $b_j^t / \sum_i^M b_i^t$. Model based clustering is (without Markov assumption, no a_{ij}) the unconditional case (no input x). The presence of the second term, a_{ij}, however introduces the trade-off with the first term towards the entire likelihood. In most applications, the main diagonal elements a_{ii}, describing the self-transitions (i.e., the probability of staying in a state) typically have values above 0.9, corresponding to an average time of staying in the state of above ten steps. Only if the next observation in the sequence can be explained much better by a state different from the current state does the model switch to the next state.

2.3 Modeling the Conditional Emission Probabilities: The Experts

- **Independence.** Given the input of the emission model, the likelihood of observing y^t given the current state and given the current input is $b_j^t = P(y^t|s^t = j, x^t, \theta)$. They are independent for each t. We call each of the specified emission models an **expert**, and each individual expert corresponds to a state.

- **Density Function.** We can assume different forms for the distribution of the "noise". In the specific example of a Gaussian, the emission probability of expert j becomes

$$b_j^t = P(y^t|s^t = j, x^t, \theta)$$

$$= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{\left(y^t - \widehat{y}_j^t(x^t)\right)^2}{2\sigma_j^2}\right)$$

$\widehat{y}_j^t(x^t)$ is the conditional mean, and σ_j^2 is the variance of the predicted Gaussian density.

- **Architecture.** The functional dependence of the conditional mean $\widehat{y}_j^t(x^t)$ on its input (x^t) can potentially be nonlinear and is expressed through a feedforward neural network with nonlinear hidden units and a linear output transfer function. Removing the hidden units reduces the functional form of each expert to a linear relation. In the autoregressive case with a single lag, $\widehat{y}^t$ is given by $\widehat{y}^t = k_0 + k_1 x^t$. We can use linear autoregressive models as well as nonlinear neural networks as experts. The emission probability, $\mathbf{B}$, is determined by the set of parameters, θ_j, of expert j, according to the architecture of the emission model.

Different experts can have different sets of inputs. Typically, the number of inputs to each expert is a subset of the full set of inputs that would be useful. When different dynamics modeled by the different experts "live" on subsets of the full set of inputs, this approach can help reduce the "curse of dimensionality."

2.4 Computing the Likelihood: The Forward-Backward Procedure

Rather than computing $P(\mathcal{Y}|\theta)$ directly using Eq. 5, Baum (1972) proposed an elegant algorithm called the **forward-backward** procedure. Dempster, Laird and Rubin (1977) subsequently introduced the so-called "Expectation Maximization" or EM algorithm to maximize this probability. HME build on these powerful algorithms.

Let α_i^t be the *joint* probability of having observed y from time 1 to time t and of being in state i at time t,

$$\alpha_i^t = P(y^1, y^2, ..., y^t, s^t = i|\theta)$$

where $1 \leq t \leq T$ and θ denotes the model parameters. The probability of the *entire sequence* of observations is given by the sum over the states at the end of the sequence (at time T):

$$P(\mathcal{Y}|\theta) = \sum\nolimits_{i=1}^{M} \alpha_i^T \quad . \tag{6}$$

The breakthrough of this idea is the computational complexity. Rather than being exponential in T (as one might expect, given the consideration of paths), it is only linear in time: α_i^T can be computed recursively

$$\alpha_j^{t+1} = \left[\sum_{i=1}^{M} \alpha_i^t a_{ij}\right] .b_j^{t+1} \tag{7}$$

At the beginning of the sequence, the α's are initialized with probability $\alpha_i^1 = \pi_i b_i^1$. This recursion is called the *forward procedure.* Given initial estimates of π_i and b_i^1, Eq. 7 prescribes the computation of the probability $P(\mathcal{Y}|\theta)$, and, for $t = T$, the entire likelihood. Similarly, the backward variable β_i^t is defined as the *conditional* probability of observing y from $t+1$ to T given state i at time t (and, as always, the parameters):

$$\beta_i^t = P(y^{t+1}, y^{t+2}, ..., y^T|s^t = i, \theta) \quad .$$

The recursive induction for β starts at the end of the sequence with from $\beta_i^T = 1\ \forall i$,

$$\beta_i^t = \sum_{j=1}^{M} a_{ij} b_j^{t+1} \beta_j^{t+1} \quad . \tag{8}$$

With $t = T-1, T-2, \ldots, 2, 1$, we obtain the β's for all t. Combining α and β, we now obtain the important posterior probability of being in state i at time t given the entire set of observations and parameters

$$\begin{aligned}\gamma_i^t &= P(s^t = i|\mathcal{Y},\theta) \\ &= \frac{P(\mathcal{Y}, s^t = i|\theta)}{P(\mathcal{Y}|\theta)} \\ &= \frac{\alpha_i^t\beta_i^t}{\sum_{k=1}^M P(\mathcal{Y}, s^t = k|\theta)} \\ &= \frac{\alpha_i^t\beta_i^t}{\sum_{k=1}^M \alpha_k^t\beta_k^t} \quad . \end{aligned} \tag{9}$$

γ_i^t is a key quantity that will serves as the estimate for $P(s^t = i|\theta)$.

Finally, the joint probability of the conjunction, $\xi_{ij}^{t,t+1} = P(s^t = i, s^{t+1} = j|\mathcal{Y},\theta)$, is also computed from α and β:

$$\begin{aligned}\xi_{ij}^{t,t+1} &= \frac{P(s^t = i, s^{t+1} = j, \mathcal{Y}|\theta)}{P(\mathcal{Y}|\theta)} \\ &= \frac{\alpha_i^t a_{ij} b_j^{t+1} \beta_j^{t+1}}{\sum_{i=1}^M \sum_{j=1}^M \alpha_i^t a_{ij} b_j^{t+1} \beta_j^{t+1}} \quad . \end{aligned} \tag{10}$$

The sub-section defined the important *variable* γ_i^t, the probability of being in state I at time t, and showed how it can be computed from α_i^t and β_i^t capturing the likelihoods of the beginning of the sequence through t, and from t to the end of the sequence, respectively. The variable ξ will serve as an auxiliary quantity in the computation of the transition probabilities, discussed in the next section that discusses how the *parameters* of the model are estimated.

2.5 The Baum-Welch Algorithm: EM Algorithm for Hidden Markov Models

The likelihood as given by Eq. 5 cannot be maximized directly since the hidden states are not known. The solution of this problem goes back to Baum et al. (1970). For the sets of parameters θ and θ^{old}, an auxiliary Q-function is defined:

$$Q(\theta,\theta^{old}) = \sum_{\forall q^T} P(\mathcal{Y}^T, q^T|\theta^{old}) \log P(\mathcal{Y}^T, q^T|\theta) \quad . \tag{11}$$

It can be shown that $Q(\theta,\theta^{old}) > Q(\theta^{old},\theta^{old}) \implies P(\mathcal{Y}^T|\theta) > P(\mathcal{Y}^T|\theta^{old})$ (Baum et al. 1970, Liporace 1982, Juang 1984). This re-estimation algorithm is called the **Baum-Welch algorithm**. Its key idea is to go back and forth between two steps, the E-step and the M-step.

- The **E-step** ("Expectation Step") assumes that the parameters of the model are known, and computes for each time step t the likelihoods α_i^t and β_i^t, and in turn, the posterior probabilities γ_i^t and $\xi_{ij}^{t,t+1}$.
- The **M-step** ("Maximization Step") takes the variables computed in the E-step and updates the parameters of the model such that Eq. 11 is maximized under the constraints $\sum_{i=1}^M \pi_i = 1$ and $\sum_{j=1}^M a_{ij} = 1$.

The new **transition probabilities** are given by:

$$a_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i \text{ (to anywhere)}} = \frac{\sum_t \xi_{ij}^{t,t+1}}{\sum_t \gamma_i^t} \quad .$$

The new *initial probabilities* of state i are $\pi_i = \gamma_i^1$.

While the original work by Baum et al. (1970) estimated only the *unconditional* density for each state, one of the important points of this paper is that it allows for *conditional* densities. The formulae for the re-estimation of the *emission parameters* depend both on the specific noise model and the specific functional form for the parameters of the noise model (e.g., linear dependence, neural network) of the experts. For each expert, Eq. 11 is maximized when the following G-function is maximized (cf., Fraser and Dimitriadis, 1994):

$$G = \sum_{t=1}^{T} \sum_{j=1}^{M} \gamma_j^t \, \log P(y^t | x^t, s^t = j, \theta_j) \quad . \tag{12}$$

θ_j represents the parameters of the emission model of state j. Equation 12 can be interpreted as the negative of a cost function for the emission model. The estimation of the parameter θ_j depends on the specific form of the emission model. To be able to write down specific formulae for updating the parameters, the errors are assumed to be Gaussian. We first discuss the update for the variance of expert j, σ_j^2. Assuming that σ_j^2 depends only depends only on the expert and not on any inputs, the llikelihood is maximized when

$$\frac{\partial G}{\partial \sigma_j^2} = \sum_{t=1}^{T} \gamma_j^t \, \frac{1}{P(y^t | x^t, s^t = j, \theta_j)} \frac{\partial P(y^t | x^t, s^t = j, \theta_j)}{\partial \sigma_j^2}$$

takes the value of zero, yielding

$$\sigma_j^2 = \frac{\sum_{t=1}^{T} \gamma_j^t \left(y^t - \widehat{y}_j^t\right)^2}{\sum_{t=1}^{T} \gamma_j^t} \quad . \tag{13}$$

This is the γ_j^t-weighted squared error between observation y^t and prediction $\widehat{y}_j^t$. It describes the "local" noise level for expert j.[7]

The mean of expert j, $\widehat{y}_j^t(x_j^t)$ is a function of the inputs into the expert, x_j^t. This (linear or nonlinear) dependence is parameterized with θ_j. To maximize Eq. 12, its partial derivative with respect to the parameters θ_j has to vanish:

$$\begin{aligned}\frac{\partial G}{\partial \theta_j} &= \sum_{t=1}^{T} \gamma_j^t \, \frac{1}{P(y^t | x^t, s^t = j, \theta_j)} \frac{\partial P(y^t | x^t, s^t = j, \theta_j)}{\partial \theta_j} \\ &= \sum_{t=1}^{T} \gamma_j^t \, \frac{y^t - \widehat{y}_j^t}{\sigma_j^2} \, \frac{\partial \widehat{y}_j^t}{\partial \theta_j}\end{aligned} \tag{14}$$

where the mean of the Gaussian of the jth expert is given by $\widehat{y}_j^t = \widehat{y}_j^t(x_j^t, \theta_j)$. In the special case where this functional form is linear, the parameters for expert j can be estimated directly by regressing $\sqrt{\gamma_j^t}\, y_j^t$ onto $\sqrt{\gamma_j^t}\, x_j^t$. In the general nonlinear case, each pattern still has the importance γ_j^t, but the parameters are to be estimated iteratively, as an additional inner loop within each M-step.[8] Interpreting it as a cost function for a neural network, each expert minimizes the weighted squared error

$$\sum_{t=1}^{T} \gamma_j^t \left(y^t - \widehat{y}_j^t(x_j^t, \theta_j)\right)^2 \quad .$$

[7] The corresponding formulae for the case of vector-valued predictions are the γ_j^t-weighted covariances for dimension m and n:

$$\frac{1}{\sum_{t=1}^{T} \gamma_j^t} \sum_{t=1}^{T} \gamma_j^t \left(y^t(m) - \widehat{y}_j^t(m)\right) \left(y^t(n) - \widehat{y}_j^t(n)\right) \quad .$$

In many applications it is reasonable to consider only a diagonal covariance matrix. This implies that the noise is drawn independently for the different outputs, and can often be interpreted more easily than the general case that allows for a rotation.

[8] The nonlinear case is sometimes called the *generalized* EM (or GEM) algorithm.

The parameters θ_j can be estimated through weighted error backpropagation (local linearization around θ_j and taking a small step towards a better solution). This justifies viewing γ_j^t as an *effective learning rate.*

3 PREDICTING AND EVALUATING DENSITIES

The previous section emphasized the underlying assumptions and algorithms for estimating the model. This section discusses how density predictions are generated and evaluated.

3.1 Generating the Density Forecasts

To generate a predictive density from a given HME model, one might be tempted to use the state as estimated by Eq. 9. This, however, is cheating: γ_j^t is estimated using the entire sequence of observations, including future information. However, given the sequence of observations through time t, we can estimate the predictive probability of a state in terms of the transition probabilities a_{ij} and the joint α_i^t probability of state $s = j$ at time $t+1$ and the observations through time t, as

$$\begin{aligned} P(s^{t+1} = j|\mathcal{Y}^t, \theta) &= \frac{P(\mathcal{Y}^t, s^{t+1} = j|\theta)}{P(\mathcal{Y}^t|\theta)} \\ &= \frac{\sum_{i=1}^{M} \alpha_i^t a_{ij}}{\sum_{j=1}^{M}(\sum_{i=1}^{M} \alpha_i^t a_{ij})} =: g_j^{t+1} \quad . \end{aligned} \tag{15}$$

Using the same notation for HME as for GE (Weigend et al. 1995), we use g_j^{t+1} as an abbreviation for $P(s^{t+1} = j|\mathcal{Y}^t, \theta)$. Note that g is a *causal* version of the γ—it is based only on past information (through α) but does not use any future information (that enters γ through β).

The density for y^{t+1} is the linear g_j-weighted superposition of the densities of the individual experts:

$$\begin{aligned} P(y^{t+1}|\mathcal{X}^t, \mathcal{Y}^t, \theta) &= \sum_{j=1}^{M} P(y^{t+1}|\mathcal{X}^t, s^{t+1} = j, \theta_j) P(s^{t+1}|\mathcal{Y}^t, \theta) \quad (16) \\ &= \sum_{j=1}^{M} g_j^{t+1} P(y^{t+1}|\mathcal{X}^t, s^{t+1} = j, \theta_j) \quad . \end{aligned}$$

$\mathcal{X}^t$ summarizes the set of exogenous variables that are available at time t. For the specific case of Gaussian distributions for the individual noise models, the individual densities are described by their conditional means $\widehat{y}_j^{t+1}$ and the variances σ_j^2. This completes the discussion of the ingredients needed to generate the full distribution for y^{t+1}.

Should one be interested in the overall mean of the predicted density at time $t+1$, due to its linearity, it is the g_j-weighted superposition of each individual mean:

$$\widehat{y}^{t+1} = \sum_{j=1}^{M} g_j^{t+1} \widehat{y}_j^{\,t+1} \quad . \tag{17}$$

However, recall that the key goal is to generate a forecast of the density, and not just its mean. The emphasis on densities requires special care in evaluating the forecasts. The next subsection presents different evaluation methods.

3.2 Evaluating the Density Forecasts

We use two different methods to evaluate density forecasts that complement each other well. While the first method is based on the probability density function itself (pdf), the second method is based on the integral of the pdf, i.e., the cumulative distribution function (cdf).

- For each time step in the test set, the **pdf-based evaluation** records $\log P(y^{t+1}|\mathcal{X}^t,\mathcal{Y}^t,\theta)$, the value of the logarithm of the predicted density at the corresponding observation y^{t+1}. The average of these $\log P$ over the test set is used as measure for evaluation.[9] This average (or "per-pattern") likelihood of out-of-sample data given the density predictions of the model allows direct comparison between different model classes. Since the value on a specific test set is only an estimate for the true value, it is important to use identical training and test sets in the comparisons.[10]

- The **cdf-based evaluation** computes for each time step the cumulative probability distribution from the predicted density and records the value of the cdf at the observed data point for each day. This probability integral transform was proposed by Diebold et al. (1998). Z^{t+1} denotes the value that the predicted cdf takes at the observation y^{t+1}:

  $$Z^{t+1} = \int_{-\infty}^{y^{t+1}} P(\eta^{t+1}|\mathcal{X}^t,\mathcal{Y}^t,\theta)d\eta$$

  η is the integration variable. The key idea is that these values of Z should be uniformly distributed. Diebold et al. (1998) point out that standard procedures (e.g., Kolmogorov-Smirnov) test jointly for uniformity and independence. If the test is rejected, it is not clear what conclusions should be drawn. We follow their suggestion and first evaluate unconditional uniformity using a simple histogram. Second, to evaluate whether Z is iid, we show the correlogram of the centered $(Z-\overline{Z})$, where $\overline{Z}$ is the mean of Z. To explore dependencies beyond linearity, we also show the correlogram of the powers of $(Z-\overline{Z})$.

For completeness, we also give the normalized mean squared error defined as

$$E_{\text{NMS}} = \frac{\sum_t \left(\text{observation}^t - \text{prediction}^t\right)^2}{\sum_t \left(\text{observation}^t - \text{mean}_{\text{train}}\right)^2} = \frac{\sum_t \left(y^t - \widehat{y}^t\right)^2}{\sum_t \left(y^t - \text{mean}_{\text{train}}\right)^2} \tag{18}$$

where t usually enumerates the points in the withheld test set. E_{NMS} compares the model's point predictions to simply predicting the mean of the training set. Note, however, that the normalized mean squared error only evaluates the point prediction and thus requires that we collapse the density prediction for each time step onto its mean. When predicting financial returns, many people do not expect a significant improvement over predicting the mean of past data. When the mean of the time series shifts, E_{NMS} can actually be larger than unity. This is also the case when the daily S&P500 forecasts are reduced to the mean and evaluated with E_{NMS}. However, the first two methods, using the pdf and the cdf, both evaluating the density, exhibit strong differences between the model classes.

4 EXAMPLE 1: COMPUTER GENERATED DATA

For complicated model classes, it is important to understand the behavior of the model and to build up some intuitions about what happens when the model assumptions deviate from those

[9] To avoid possible confusion, it might be worth pointing out that there are two very different likelihood functions in estimation and in evaluation. The likelihood function maximized in the model estimation or search, Eq. 6, considers the likelihood of the *sequence*—this includes the trade-off between staying in a regime and allowing for somewhat worse predictions vs. changing regimes and obtaining better predictions. Note that this likelihood includes the transition matrix (Eq. 7). In contrast, the likelihood function used for evaluation does not take transitions into account but only measures for each time step how well the observation was predicted by the pdf. It is important that this likelihood does not contain aspects of the sequence or the transition probabilities, but only the predicted densities. This allows for clean comparisons between approaches to density prediction.

[10] We thank Art Owen for pointing out that average log-likelihood can be very sensitive to a few extreme values. We computed trimmed means, but it turns out that outliers in log-likelihood are not a problem in the experiments reported here.

of the generating process. Since mixture models contain an unsupervised part in learning, this section investigates whether the states found by the model actually correspond to the true hidden states.[11] We generate data from a hidden Markov model with two states, and analyze these data with HME, GE and, as a naive sanity check, an unconditional Gaussian.

4.1 Generation and Recognition Models

The **data generation** consists of two distinct and different processes: the Markov chain of the hidden states, and the dynamics of the individual experts.

- **Dynamics of the Markov model**. The transition probabilities are given by the matrix

$$\mathbf{A} = \begin{pmatrix} 0.98 & 0.02 \\ 0.03 & 0.97 \end{pmatrix} .$$

This allows us to generate a realization for the time series of the hidden states.

- **Dynamics of the individual experts**. With financial processes in mind, we pick the first process as trending, and the second process as mean reverting:

$$y^{t+1} = \begin{cases} 0.5\ y^t + 0.8\ \varepsilon^{t+1} & \text{if in state 1} \\ -0.3\ y^t + 0.5\ \zeta^{t+1} & \text{if in state 2} \end{cases} .$$

ε and ζ are $N(0,1)$ iid.

We first generated a sequence of length 15,000 of the (eventually hidden) states. This sequence determined which of the two processes was used for each time step to generate an "observation". From the generated data, we use the first 10,000 points as the training set, and the remaining 5,000 points as the test set.

The **recognition models** are HME, GE and the unconditional Gaussian, defined by the mean and variance of the training set. In the case of HME, it is possible to choose the recognition process to perfectly match the data generating process by using two experts that are linear autoregressive models with one lag, AR(1).

The GE model used for the comparison also has two linear AR(1) experts, chosen to be as similar to the HME model as possible. The difference is that the probability of being in state j at time t in the GE model is learned as a feed-forward function of some variables, as opposed to recursively from the series itself using the hidden Markov assumption. One of the two gate inputs is the input that is also used in the experts, i.e., the current value of the time series, y^t. The other gate-input is an exponential moving average of the squared values of the observations $\left(y^t\right)^2$

$$\xi^t = \lambda\ \xi^{t-1} + (1-\lambda)\ \left(y^t\right)^2 \tag{19}$$

where the decay constant $\lambda = 0.95$. The gate is implemented as a nonlinear neural network with three hidden units (tanh transfer function) and two outputs. The outputs are constrained to be positive and to sum to unity, using the "softmax" architecture as discussed in Weigend et al. (1995).

[11] In the real world, such as when modeling S&P500 densities (Section 5), we do not know the true model. This problem is particularly serious in finance for two reasons. First, while in the sciences experiments are usually carried out under carefully controlled conditions, finance does not allow for carefully controlled experiments. Second, the high amount of noise tends to mask subtle differences between competing models. This is again quite different to, say, physics, where some predictions are made with incredible accuracy and the data can distinguish between two models that make almost the same predictions.

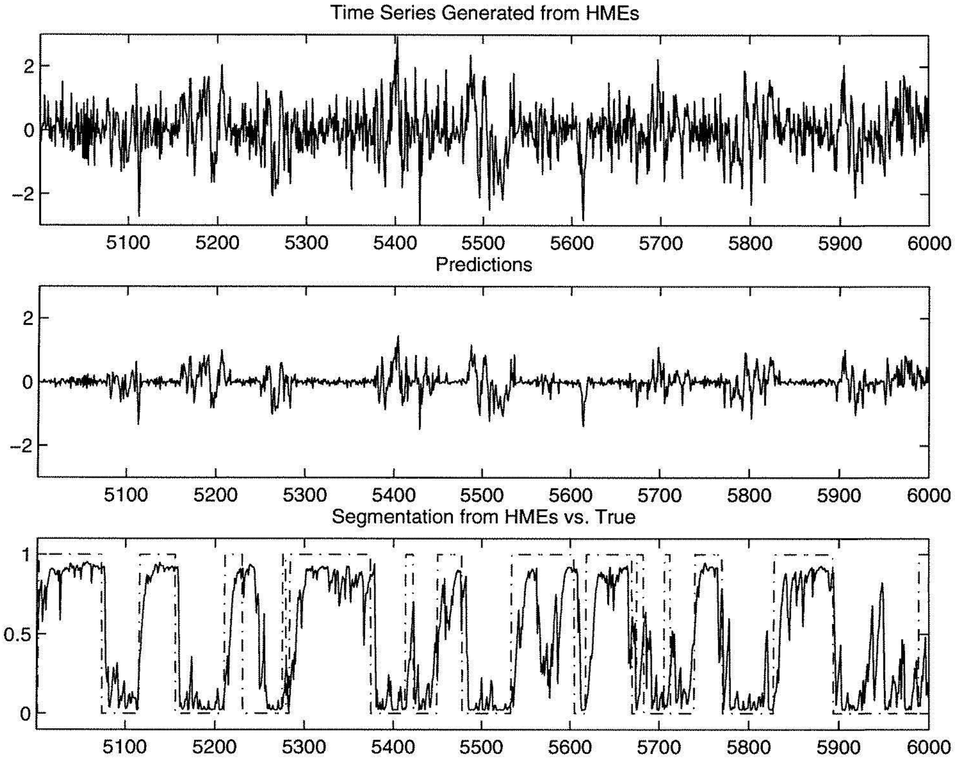


Figure 1: Time series of the computer generated data modeled with HME. From top to bottom, the panels display 1,000 true values of the out-of-sample data, the point forecasts, and the probability of one expert, g_1^t. It sums with the other expert (not shown) to unity for each time step. The true regime used in the generation of the test data is indicated in the bottom panel as dash-dotted line.

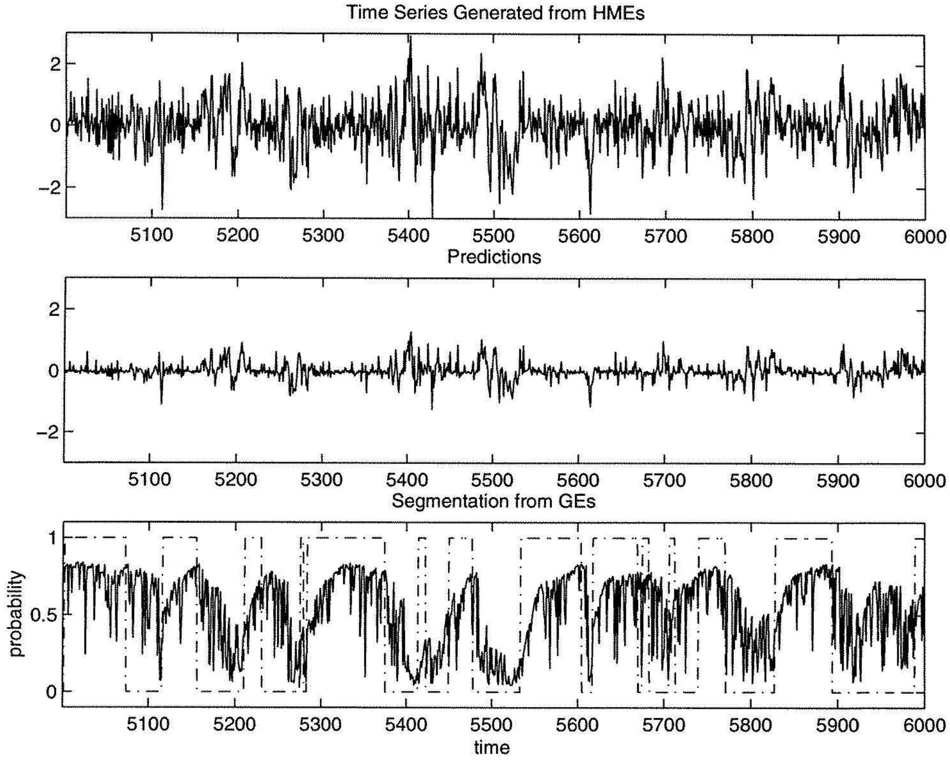
4.2 Results and Interpretation

We present selected results on the computer generated data for several purposes:

- Illustrate to what degree HME recover the hidden regimes on these fairly noisy time series. Note that true regimes are known in the computer generated example, but not in real world examples.
- Show how the partitioning of the gate-input space performed by the GE differs from the segmentation of the HME. For real applications, it is important to recognize signatures that indicate the wrong model class.
- Build up some intuitions for interpreting results of the analysis based on the probability integral evaluation proposed by Diebold et al. (1998).

Figures 1 and 2 show in the time domain the same 1,000 points of the test set for HME and for GE, respectively. In both figures, the top panel shows the true data. The bottom panel shows the probability that the model predicts for one of the two experts. The probability of the other expert is not shown but corresponds to the difference between unity and the probability shown. The dash-dotted line indicates the true regimes used in the generation of the test data. Despite the high noise level in both training and test data, HME discover the regimes adequately.

The corresponding results for GE are shown in Fig. 2. The training and test data are identical to those used for the HME. The GE architecture is chosen to be as similar to the HME architecture as possible, as discussed in Section 4.1. The main difference is in the segmentation. Comparing


Time Series Generated from HMEs

Predictions

Segmentation from GEs

Figure 2: Time series of the computer generated data modeled with GE. The top panel shows 1,000 points of the test set, the middle panel the one-step-ahead point predictions of the GE for the same period, and the bottom panel the probability g_1^t. These probabilities are not as clean as those found by HME since GE ignore the difference between adjacent and distant patterns in time.

the bottom panels of Figs. 1 and 2, note that the regime assignments are cleaner for HME than for GE. This can be explained by the different likelihood functions: while GE represent a feedforward architecture that necessarily produces solutions that are invariant under re-shuffling of the input-output patterns, HME "know" about the sequence of the patterns through the assumption of the hidden Markov structure. HME can be said to trade-off the switching with the likelihood of the observation.

Although this paper focuses on density prediction, we included the mean of each one-step-ahead prediction as the middle panels. For these point predictions, the normalized mean squared errors (defined in Eq. 18) on a 5,000 point test set are for the two model classes $E_{\mathrm{NMS}}(\mathrm{HME}) = 0.826$ and $E_{\mathrm{NMS}}(\mathrm{GE}) = 0.886$ with the ratio (squared error(HME))/(squared error(GE)) = 0.93 .

We now turn to the evaluation of the densities, first using the predicted probability densities directly. On the same test set as above, the log-likelihood ratios are:

$$\frac{\text{log-likelihood(HME)}}{\text{log-likelihood(GE)}} = 0.96 \quad \text{and} \quad \frac{\text{log-likelihood(HME)}}{\text{log-likelihood(Gaussian)}} = 0.57 \quad .$$

While there is a clear improvement of the conditional mixtures over the unconditional Gaussian, the difference between the mixtures is not significant.

This second approach uses the cdf-based integral transform (Diebold et al. 1998). This analysis focuses on Z^t, the area of the pdf to the left of the observation, i.e., the probability that a value below the observation was predicted. The qualitative aspects of the density forecasts are exposed in Figs. 3, 4, and 5 for HME, GE and the naive unconditional Gaussian, respectively. In these figures, the top panels give the histogram of Z on the test set. As discussed in Section 3.2, Z should be uniformly distributed between 0 and 1. The remaining four panels in each figure show the correlograms of powers of the mean-subtracted Z-series, i.e., the empirical autocorrelations of $(Z - \overline{Z})$, $(Z - \overline{Z})^2$, $(Z - \overline{Z})^3$, and $(Z - \overline{Z})^4$.

Analyzing the density predictions obtained with HME, Fig. 3 indicates that the histogram of the Z series is consistent with a uniform distribution. Furthermore, there are no significant autocorrelations in the powers of the mean-subtraced Z-series. These good density forecasts are reassuring—but not surprising since the structure of the HME recognition model was chosen to be identical to that of the generating process.

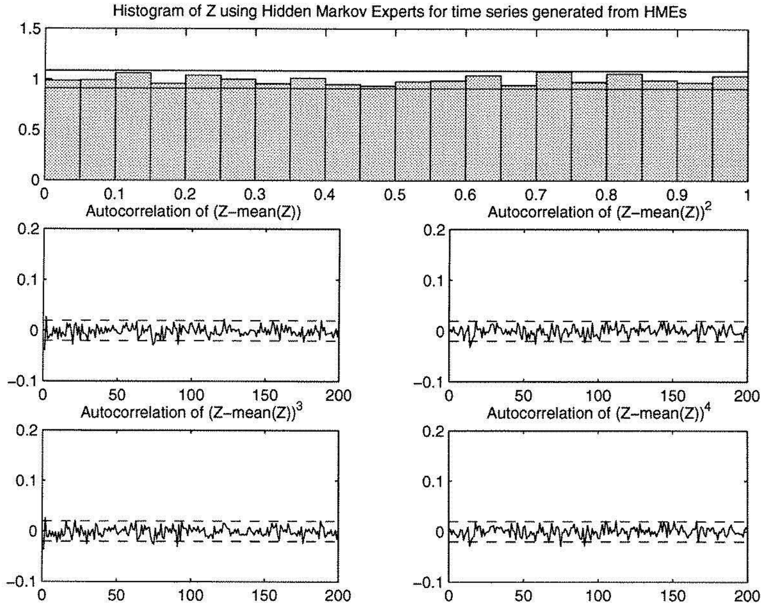


Figure 3: Evaluation using the integral transform, Z, of the probability density predictions generated by HME. The histogram of Z indicates that the distribution of Z is uniformly distributed between 0 and 1, indicating good density predictions. The absence of autocorrelations indicates that there is no residual time structure in the mean corrected Z and its powers. The horizontal lines indicate two standard deviations.

Figure 4 shows the effect of a misspecified model. While the structure of the emission models (the experts) is still identical to the data generating process, GE cannot model correctly the underlying Markov structure of the sequence. In comparison to HME (Fig. 3), the histogram for GE is less uniform, and there are some short but significant autocorrelations in $(Z - \overline{Z})$ and $(Z - \overline{Z})^3$.

To put the qualitative aspects of the HME and GE predictions into perspective, Fig. 5 presents the histogram and the correlograms of Z when the model is a single unconditional Gaussian. In this model, more observations occur than were predicted in the central region of the histogram of about one standard deviation, and fewer observations in the areas around the 10 and 90 percentiles.[12] Furthermore, there are long autocorrelation dependencies in the Z-series. The non-uniformity of the histogram and the Z-autocorrelations are consistent with the poor performance on the quantitative measures of squared errors and of the log-likelihood.

Knowing the true model in this first example of computer generated data allows us to compare the estimated parameters with the true parameters: the diagonal elements of transition probability matrix $\mathbf{A}$, the autoregressive coefficients k_i, and the noise level σ_i. Table 1 gives the true values

[12] The histogram focuses on the central part of the distribution since each bin has roughly the same number of points. The histogram can be viewed as expanding the center and compressing the tails. To focus on the tails of the distribution, a quantile-quantile plot (qq-plot) is more appropriate. It shows that there are too many observations in the extreme tails.

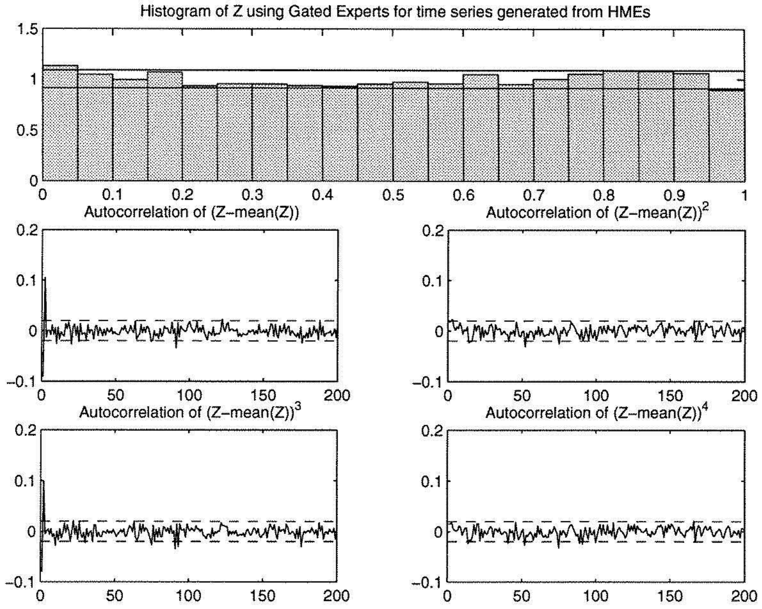


Figure 4: Evaluation of density predictions using GE on the computer generated data. Note the appearance of significant autocorrelations for the odd powers of $(Z - \overline{Z})$ compared to the correctly specified HME.

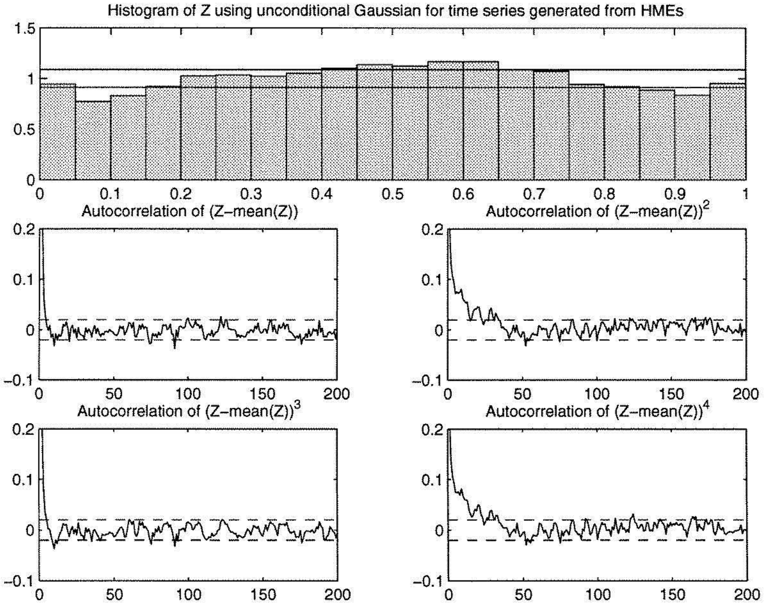


Figure 5: Evaluation of density predictions using an unconditional Gaussian on the computer generated data. The model mismatch is indicated by both the non-uniformity of the histogram and the significant autocorrelations in the correlograms.

of the parameter and the estimates of the models. The correctly specified HME found the correct parameters. In contrast, the estimation of the corresponding GE experts is significantly worse than that of HME. In this specific run, the second expert does not even learn the mean-reverting dynamics but predicts an essentially unconditional Gaussian.

Table 1: Estimated parameters for HME and GE along with the true values used in the data generating process. a_{ii} denotes the self-transition probabilities of staying in regime i; the off-diagonal terms are the complements to unity. k_i denotes the autoregressive coefficients of the individual experts, and σ_i the noise levels of the individual experts.

	a_{11}	a_{22}	k_1	k_2	σ_1	σ_2
true value	0.980	0.970	0.500	-0.300	0.800	0.500
HME	0.976	0.969	0.507	-0.269	0.808	0.492
GE	N/A	N/A	0.466	0.003	0.867	0.528

5 EXAMPLE 2: S&P500 RETURNS

This section applies HME to the real-world problem of forecasting the density of daily S&P500 returns. To provide a perspective and a deeper understanding of HME, comparisons are carried out to several other model classes: an unconditional Gaussian, an unconditional mixture of Gaussians, a generalized autoregressive conditional heteroskedastic GARCH(1,1) model, and the GE model that is as similar as possible to the HME model.

We first describe the data and models and analyze the estimated HME model. We then present the segmentation obtained by HME and by GE and explain the difference. Among the performance comparisons, the most important metric is the direct evaluation of the out-of-sample likelihood of the test data given each of the models. We also include the graphs of the probability integral transform evaluation of the density forecasts.

5.1 Data and Model Classes

For the data, we start with 21 years of daily S&P500 prices, p^t, and compute the series we try to predict, y^t, by taking the difference between the logarithms of the prices at adjacent days

$$y^t = \log p^t - \log p^{t-1} = \log \frac{p^t}{p^{t-1}} \approx \frac{p^t - p^{t-1}}{p^{t-1}} \quad .$$

The Taylor expansion used in the last step, $\log(1 + \epsilon) \approx \epsilon$, gives the interpretation of y^t as the relative price change, i.e., as the difference between today's and yesterday's price with respect to yesterday's price. In finance, this series is referred to as *continuously compounded returns.*

We use the first ten years (from 3 Jan 1977 to 31 Dec 1986) of the data as the training set, and the last ten years (from 2 Mar 1988 to 31 Dec 1997) as the test set. To avoid possible artifacts of the Oct 1987 crash, we do not use the data from 3 Jan 1987 to 1 Mar 1988 in this study. (When using HME for risk management, rare events and crashes are not excluded.) No further transformation or preprocessing is performed.

Both HME and GE have four experts. They are liner autoregressive models that predict the mean based on the values of the previous seven lagged returns.

For GE, we need to also specify the structure of the gate: we use a nonlinear neural network with five tanh hidden units and four "softmax" outputs. The inputs into the gate include the seven lagged values of the returns given to the experts, in addition to seven lagged values of the exponential moving average of the squared returns (Eq. 19).

5.2 Results

After discussing the data and the models, we now turn to the results. We first inspect the segmentation obtained with HME and GE, then discuss the estimated parameters and their meaning, and finally turn to the evaluation of the densities, using the pdf and the cdf methods.

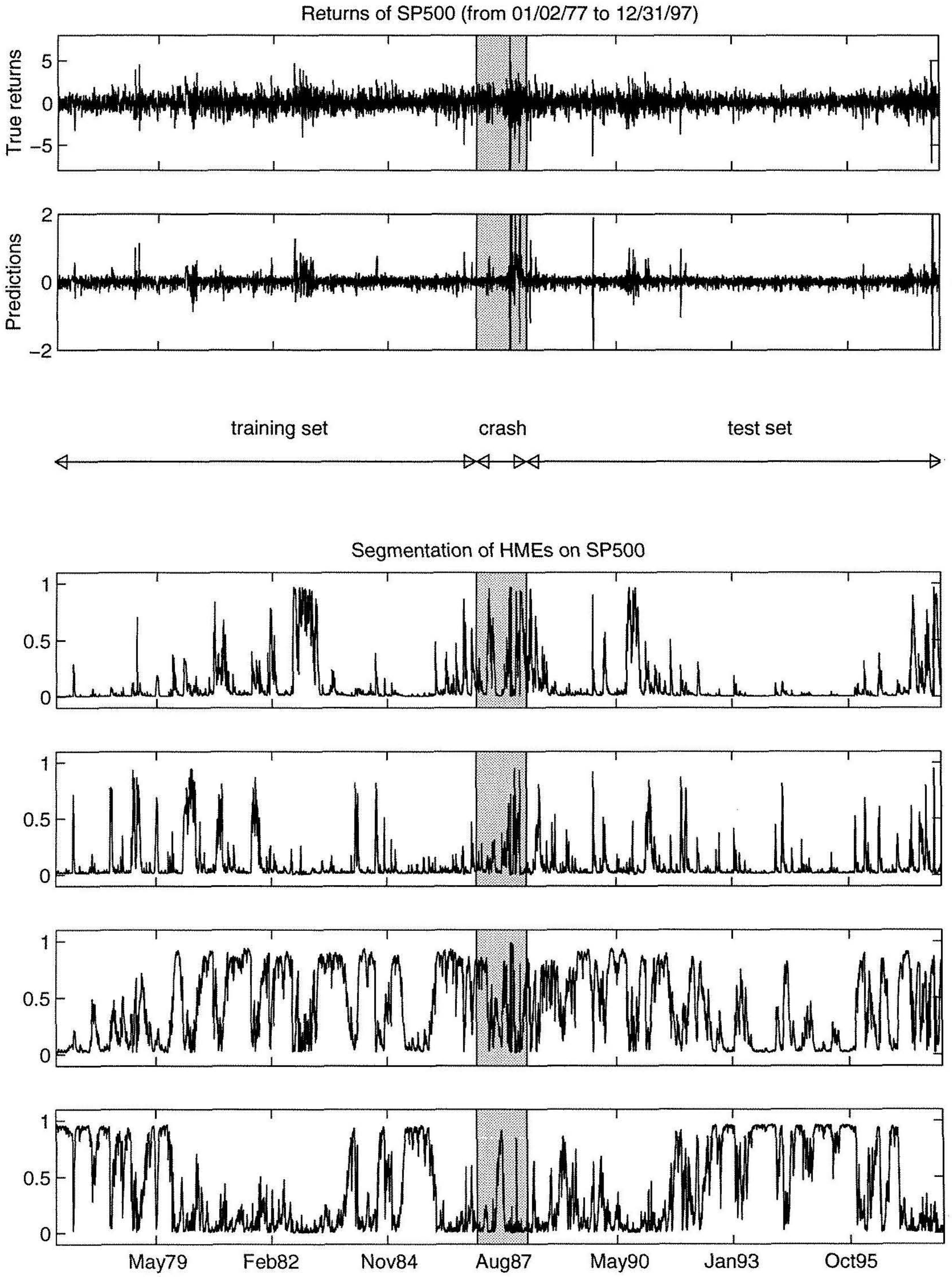


Figure 6: Time series of S&P500 returns modeled with HME. The returns (top panel) have been normalized to zero mean and unit variance. The four plots at the bottom show the probabilities of the experts for each time step. The experts are arranged by decreasing noise level: the expert with the lowest noise level is at the bottom of the figure. (For completeness, the mean of each day's density is shown in the remaining panel, labeled "Predictions".)

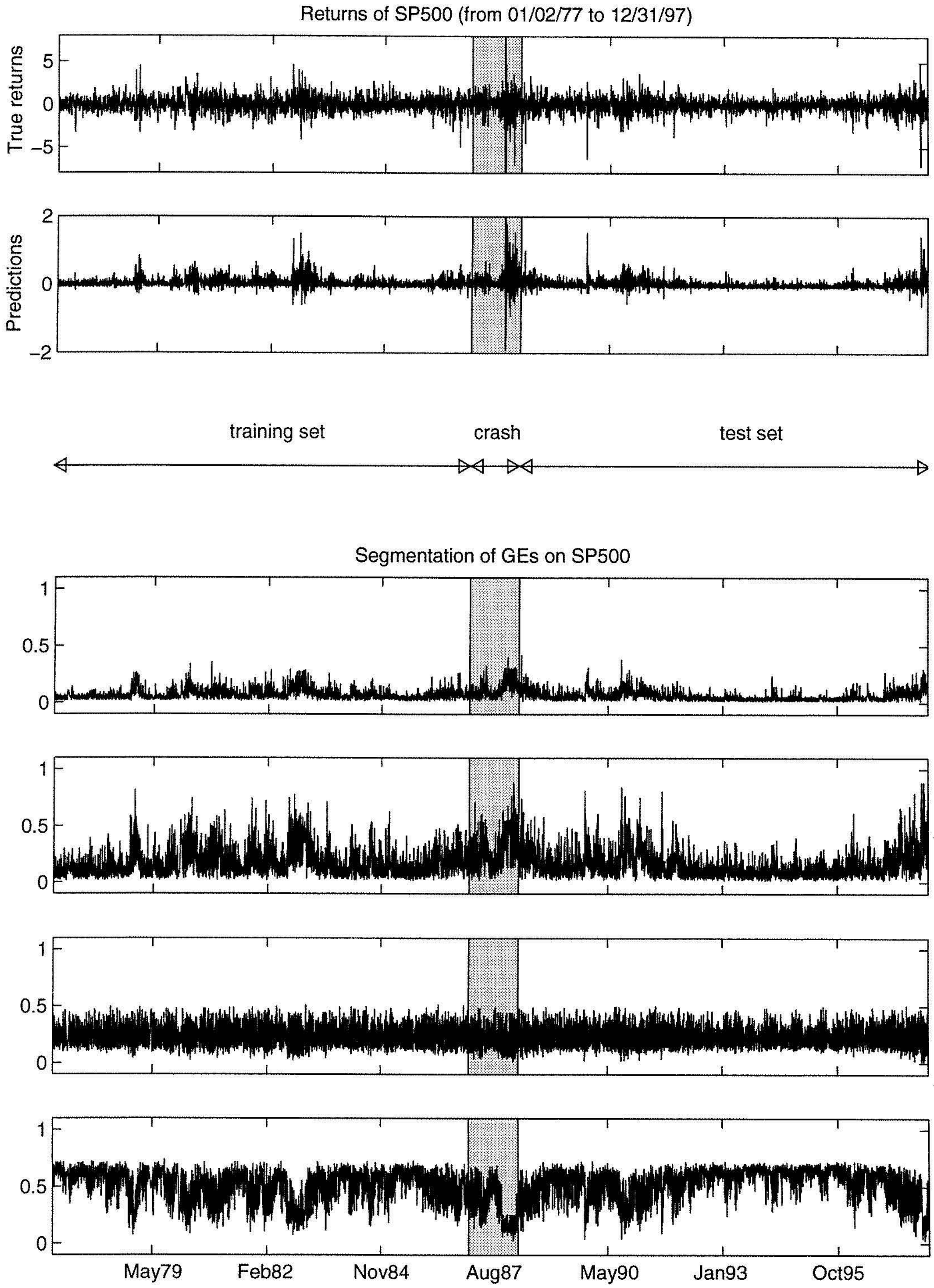


Figure 7: Time series of S&P500 returns modeled with GE. The description of the panels is the same as in the preceding figure. Note the poorer and more noisy segmentation in comparison with the previous figure.

5.2.1 Segmentation of the S&P500 Series

Figures 6 and 7 respectively show HME and GE models for daily S&P500 returns . Training and test periods are indicated by the arrows in the center of the figure. The lower half characterizes the importance of the individual experts for each day. The top panel shows the time series y^t, the daily S&P500 returns for the period from 1977 through 1997. The bottom four panels give the probability g_i^t for each expert $i (i = 1 \ldots 4)$. The experts are ordered in terms of decreasing σ_i. The expert with the lowest noise level corresponds to the lowest panel.

Figure 7 indicates that GE cannot generate clear regimes. Note, for example, that the probability of the expert with the second smallest variance (the second plot from the bottom) hardly ever leaves the range between 0.1 and 0.4. One reason for this poor segmentation is that the smoothed squared returns (Eq. 19) as gate-inputs do not characterize volatility as well as the recursively computed HME variances, σ_j^2. Another interpretation for the very noisy nature of the regimes is the absence of regime information from the neighboring pattern for GE, in contrast to HME.

5.2.2 Estimated Parameters and Interpretation

The dynamics of the hidden Markov process is characterized by the matrix of transition probabilities between the states. For the four states assumed in our model of the S&P returns, we obtain

$$\mathbf{A} = \begin{pmatrix} 0.904 & 0.031 & 0.023 & 0.042 \\ 0.014 & 0.950 & 0.029 & 0.007 \\ 0.011 & 0.014 & 0.969 & 0.007 \\ 0.011 & 0.002 & 0.004 & 0.983 \end{pmatrix} .$$

The elements of this matrix are averages over 200 initializations. For each run, the states are sorted by decreasing noise levels, σ_i, in order to make the averaging meaningful. Note that the expert with the largest noise level has the smallest self-transition probability, $a_{11} = 0.904$: on average, the system stays in this state for only ten days. Looking back to Fig. 6, we can see that this expert takes responsibility for some of the large returns in the training set, as well as for the region of high volatility in late 1982.

Table 2 lists the noise levels of the experts for both HME and GE. For each run, the experts were ordered in terms of decreasing noise levels, and means and standard deviations of the square roots of the variances of the Gaussians are shown.

Table 2: The average noise levels σ_i of the individual experts for HME and GE for the S&P500 density predictions. In each run, i.e., for each set of initial conditions, the expert with the largest variance is assigned the label "Expert 1", etc. The table gives the means of the square roots of the variances of the Gaussians. The standard deviations are indicated in parentheses. High-noise experts have more relative variation in the noise levels than the low-noise experts than in those of high-noise experts. Furthermore, GEs are more sensitive to initial conditions than HMEs.

σ_i	Expert 1	Expert 2	Expert 3	Expert 4
HME	1.37 (0.05)	0.92 (0.12)	0.74 (0.04)	0.61 (0.01)
GE	2.18 (1.10)	0.98 (0.44)	0.52 (0.16)	0.33 (0.07)

5.2.3 Evaluation of the S&P500 Density Predictions

The function optimized in training is significantly different for HME and GE—we have emphasized that HMEs include the transitions between states, whereas GEs do not. For prediction, we are ultimately interested in how well the next day's density is predicted. This function can be different from the one optimized in the estimation of the model. All architectures are compared on an equal footing: how likely are the observations of the test set given the predicted densities? We do not take any uncertainty for the observed value into account, i.e., we assume a delta distribution whose integral is unity on infinitesimally small support. We compute the log-likelihood for each pattern, take the average over the test set, and plot the results in Fig. 8. For GE and HME, we show the cumulative probability distribution of 200 runs that differ in their initializations. The main source of variation for HME are the initial emission probabilities, and for GE the initial weights of the gating network.

For comparison, Fig. 8 also shows the log-likelihoods (i) of a single Gaussian, (ii) of an unconditional mixture of four Gaussians, and (iii) of a GARCH(1,1) model, all estimated on the same training set as HME and GE, and averaged over the same test set as HME and GE. The log-likelihood of the GARCH(1,1) model is slightly better than the unconditional mixture. The log-likelihood of HME tends to be better than the unconditional mixture and the GARCH(1,1) model, indicating that the combination of the conditional variance and the mixture aspect is needed for the improvement of the quality of the density predictions.

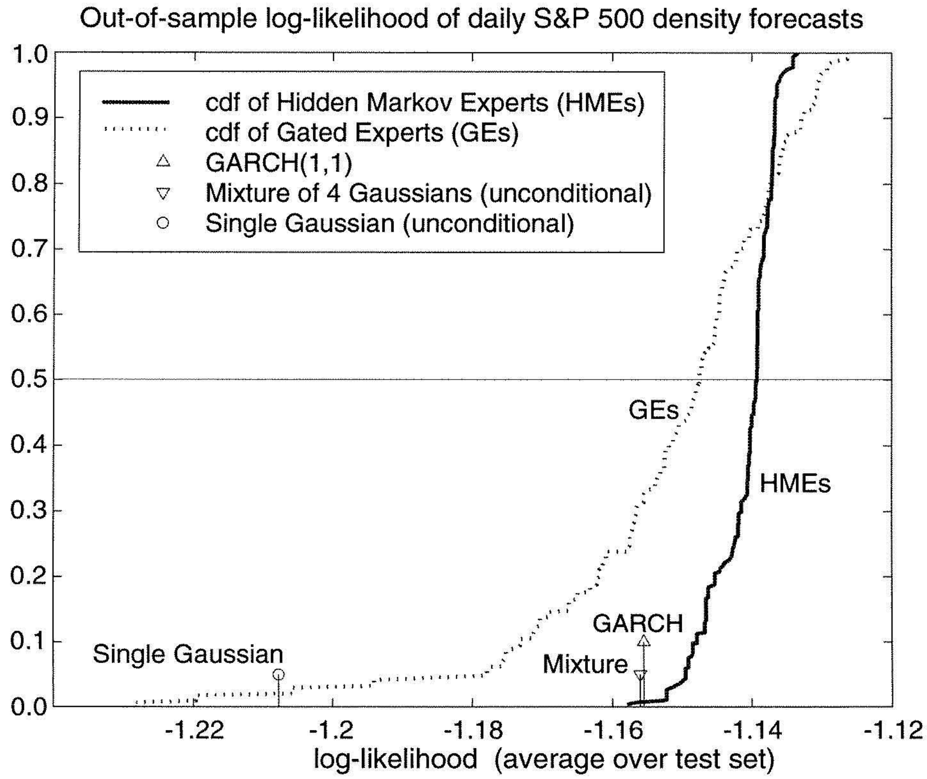


Figure 8: S&P500 density forecasts evaluated for several models based on the predicted pdf. The horizontal axis gives the log-likelihood averaged over the test set. For HME and GE the empirical cumulative distribution of 200 runs each is plotted. For comparison, we also indicate the log-likelihood averaged over the same test set for a single Gaussian, and unconditional mixture of four Gaussians, and a GARCH(1,1) model. The GE solutions are not an appropriate model class for this hard learning problem since the resulting models show a large variance in performance. In contrast, the distribution of quality of the HME is relatively sharp. Considering only the uncertainty stemming from the initialization, about 98 percent of the HME have a better out-of-sample likelihood than the GARCH model and than the unconditional mixture model. This indicates that all of HME's aspects (conditional model *and* mixture model *and* hidden Markov model) are needed for the improvement.

To rule out the possibility that the results are due to a few outliers, we analyzed the trimmed means of the log-likelihood. The ranking of the different methods remains the same when the means are trimmed; we removed up to two percent on each side. This establishes that HME give better predictions than the alternatives we considered when comparing the out-of-sample likelihood of new data.

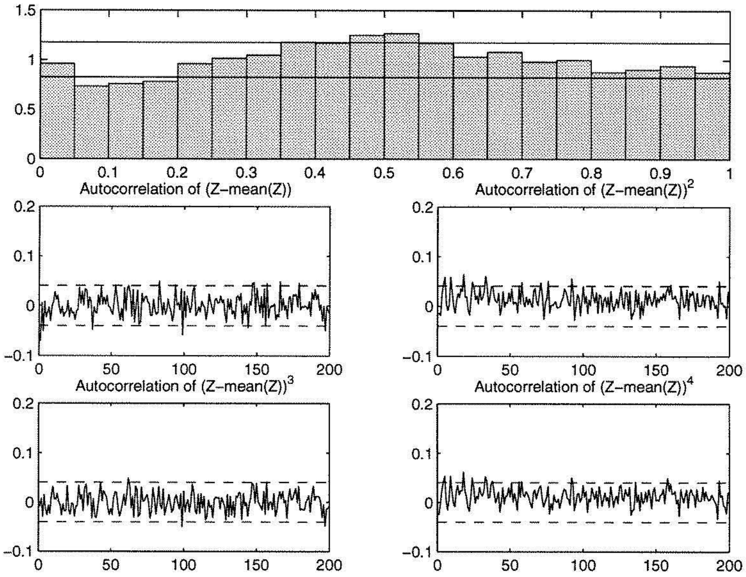


Figure 9: Evaluating the probability density predictions of HME for S&P500 returns. The top panel plots the histogram of the probability integral transform on S&P 500: the Z series is reasonably close to uniform. The four bottom panels show the correlograms: there are not many significant auto-correlations in the Z series and its powers. The dashed lines correspond to two standard deviations.

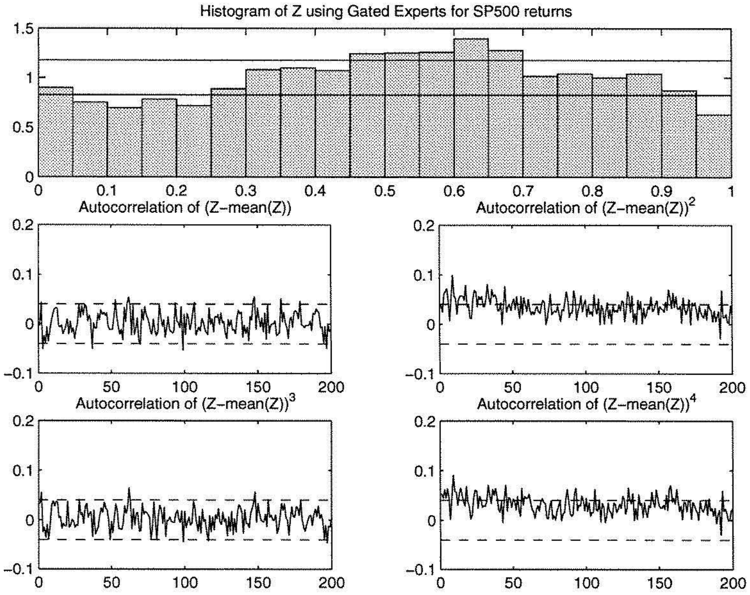


Figure 10: Evaluating the probability density predictions of GE for S&P500 returns. The Z series is less uniformly distributed as in the previous figure (HME), and auto-correlations remain in the Z series.

We now turn from the analysis based on the predicted probability distributions to an analysis based on the predicted cumulative distributions. Figures 9, 10 and 11 show the results for HME, GE, and unconditional Gaussian, respectively. In all cases, the top panel shows the histogram of the probability integral transform Z, and the four bottom panels the correlograms of the Z series and its powers. The results are acceptable for HME, slightly worse for GE and, as expected, a lot worse for the unconditional Gaussian.

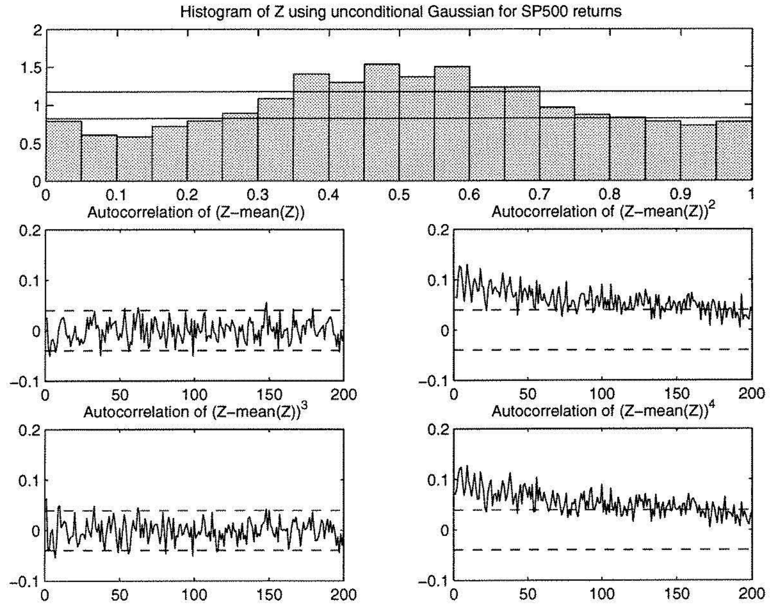


Figure 11: Evaluating the probability density predictions of an unconditional Gaussian for S&P500 returns. The Z series is far from uniformly distributed, and the auto-correlations are large.

For completeness, we close by reporting the normalized mean squared errors that can be computed by collapsing the daily density predictions to their means: $E_{\mathrm{NMS}}(\mathrm{HME}) = 1.014$ (standard deviation for 200 runs with different initial emission probabilities is 0.002); $E_{\mathrm{NMS}}(\mathrm{GE}) = 1.043$ (standard deviation for 200 runs with different initial weights is 0.083). Values larger than unity indicate a drift in the mean.

6 CONCLUSIONS

This paper started out by discussing different tasks for prediction, and proceeded by presenting hidden Markov experts (HME) in detail. The main focus is the prediction of the full conditional density distribution. This is in contrast to the literature on Markov switching models that focuses on point predictions and segmentation, and on the literature on stochastic volatility and GARCH models that focuses on conditional variances. The density predictions we obtained as mixture models were evaluated in comparison to these standard approaches using several methods, including Diebold et al. (1998).

The approach was illustrated with two time series. Section 4 showed the results of a computer generated example where the true regimes are known. This helped us obtain intuitions for model misspecification, e.g., by revealing the signature of misapplying GE to data generated by HME. When the right model class is used (HME), the parameters are estimated correctly and the density is predicted well.

Section 5 applies the approach to the density of daily S&P500. On the test set, about 98 percent of the HMEs estimated (they differed by their initial conditions) outperformed a GARCH(1,1) model. While HME found a solution rather reliably, GE showed a large dispersion for two reasons:

(i) in any task with very high noise levels it is very difficult for the gate to learn a mapping from some exogenous variables to the expected probabilities of the experts, and (ii) in the specific case of financial returns, volatility is often estimated better recursively (as in GARCH and stochastic volatility models) than with a feedforward architecture without memory, such as GE, see Timmer and Weigend (1997).

This paper focused on introducing hidden Markov experts. The examples were chosen to communicate some intuitions and illustrate several methods to evaluate the performance of density predictions. An identical set of inputs, consisting of lags of the time series, was used to facilitate the comparisons between the methods. When using this architecture in trading, we find that carefully selected exogenous inputs lead to better predictions than autoregressive models. In addition to trading applications, we have also used HME in risk management in combination with Independent Component Analysis (Back and Weigend 1997) to capture non-Gaussian tails and compute Value-at-Risk, as discussed in Chin and Weigend (1998).

Acknowledgments

We thank Jens Timmer for sharing his insights into hidden Markov models during his visits to CU Boulder and NYU. Andreas Weigend would like to thank Frank Diebold, Jerry Friedman, Clive Granger, Helmut Lütkepohl, Art Owen, Morio Yoda, and the participants of the 1997 NBER Summer Institute for their valuable feedback, in particular Bruce Hanson, Blake LeBaron and Jim Stock. Shanming Shi is gratefully acknowledges the hospitality of Stern's Information Systems Department while finishing his thesis and would in particular like to thank Fei Chen, Elion Chin, Mike Henderson, Christian Pirkner, Satinder Singh and Ashok Srivastava.

References

Back, A. D. and Weigend, A. S. (1997). A first application of independent component analysis to extracting structure from stock returns, *International Journal of Neural Systems* **8**: 473–484.

Bates, J. M. and Granger, C. W. J. (1969). The combination of forecasts, *Operations Research Quarterly* **20**: 451–468.

Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov process, *Inequalities* **3**: 1–8.

Baum, L. E. and Eagon, J. A. (1963). An inequality with applications to statistical prediction for functions of Markov processes and to a model for ecology, *Bull. Amer. Math. Soc.* **73**: 360–363.

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

Baxter, R. (1996). *Minimum Message Length Inference: Theory and Applications*, PhD thesis, Department of Computer Science, Monash University, Clayton, Australia. `www.ultimode.com/rohan/thesis.html`.

Bengio, S. and Bengio, Y. (1996). Input-output HMM's for sequence processing, *IEEE Transactions on Neural Networks* **7**(5): 1231–1249.

Cesa-Bianchi, N., Freund, Y., Helmbold, D. P., Haussler, D., Schapire, R. E. and Warmuth, M. K. (1997). How to use expert advice, *Journal of the ACM* **44**(3): 427–485.

Chatfield, C. (1993). Calculating interval forecasts, *Journal of Business and Economics Statistics* **11**: 121–135.

Chin, E. and Weigend, A. S. (1998). Market risk using hidden Markov density predictions, *Technical report*, Information Systems Department, Leonard N. Stern School of Business, New York University.

Choey, M. and Weigend, A. S. (1997). Nonlinear trading models through Sharpe Ratio maximization, *International Journal of Neural Systems* **8**: 417–431.

Christoffersen, P. F. (1997). Evaluating interval forecasts, *International Economic Review, Forthcoming.*

Clemen, R. T., Murphy, A. H. and Winkler, R. L. (1995). Screening probability forecasts: Contrasts between choosing and combining, *International Journal of Forecasting* **11**: 133–146.

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

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

Durland, J. M. and McCurdy, T. H. (1994). Duration-dependent transitions in a Markov model of U.S. GNP growth, *Journal of Business and Economic Statistics* **12**: 279–288.

Engel, C. and Hamilton, J. D. (1990). Long swings in the dollar: Are they in the data and do markets know it?, *The American Economic Review* **80**: 689–713.

Ferguson, J. D. (1980). Hidden Markov analysis: An introduction, *in* J. D. Ferguson (ed.), *The Symposium on the Applications of Hidden Markov Models to Text and Speech*, Princeton, NJ, pp. 143–179.

Fraser, A. M. and Dimitriadis, A. (1994). Forecasting probability densities by using hidden Markov models, *in* A. S. Weigend and N. A. Gershenfeld (eds), *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA, pp. 265–282.

Gray, S. F. (1996). Modeling the conditional distribution of interest rates as a regime-switching process, *Journal of Financial Economics* **42**: 27–62.

Hamilton, J. D. (1990). Analysis of time series subject to changes in regime, *Journal of Econometrics* **45**: 39–70.

Hamilton, J. D. (1994). *Time Series Analysis*, Princeton University Press, Princeton.

Hamilton, J. D. and Susmel, R. (1994). Autoregressive conditional heteroskedasticity and changes in regime, *Journal of Econometrics* **64**: 307–333.

Hansen, B. E. (1992). The likelihood ratio test under nonstandard conditions: Testing the Markov switching model of GNP, *Journal of Applied Economics* **7**: s61–s82.

Harvey, A. C. (1989). *Forecasting Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.

Huang, X. D., Ariki, Y. and Jack, M. A. (1990). *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, Edinburgh.

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

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm, *Neural Computation* **6**: 181–214.

Juang, B. H. (1984). On hidden Markov model and dynamic time warping for speech recognition-a unified view, *AT&T BLTJ* **63**: 1213–1243.

Lahiri, K. and Wang, J. G. (1994). Predicting cyclical turning points with leading index in a Markov switching model, *Journal of Forecasting* **13**: 245–263.

Liporace, L. A. (1982). Maximum likelihood estimation for multivariate observations of Markov sources, *IEEE Transactions on Information Theory* **IT-28**: 729–734.

Murphy, A. H. and Winkler, R. L. (1992). Diagnostic verification of probability forecasts, *International Journal of Forecasting* **7**: 435–455.

Pearson, K. (1894). Contributions to the mathematical theory of evolution, *Phil. Trans. Royal Soc.* **185**: 71–110. See also V. 185A, p. 195.

Poritz, A. B. (1982). Linear predictive hidden Markov models and the speech signal, *Proc. ICASSP'82*, Paris, France, pp. 1291–1294.

Poritz, A. B. (1988). Hidden Markov models: A guided tour, *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7–13.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ.

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

Timmer, J. and Weigend, A. S. (1997). Modeling volatility using state space models, *International Journal of Neural Systems* **8**: 385–398.

Wallace, C. S. and Boulton, D. M. (1968). An information measure for classification, *Comp. J.* **11**: 185–195.

Weigend, A. S. and Gershenfeld, N. A. (eds) (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA.

Weigend, A. S., Mangeas, M. and Srivastava, A. N. (1995). Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting, *International Journal of Neural Systems* **6**: 373–399.