Experiments on the Application of IOHMMs to Model Financial Returns Series

Yoshua Bengio    Vincent-Philippe Lauzon    Réjean Ducharme

Département d'informatique et recherche opérationnelle
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{bengioy,lauzon,ducharme}@iro.umontreal.ca

March 15th, 1999

Technical report #1146,
Département d'informatique et recherche opérationnelle
Université de Montréal

Abstract

Input/Output Hidden Markov Models (IOHMMs) are conditional hidden Markov models in which the emission (and possibly the transition) probabilities can be conditioned on an input sequence. For example, these conditional distributions can be linear, logistic, or non-linear (using for example multi-layer neural networks). We compare the generalization performance of several models which are special cases of Input/Output Hidden Markov Models on financial time-series prediction tasks: an unconditional Gaussian, a conditional linear Gaussian, a mixture of Gaussians, a mixture of conditional linear Gaussians, a hidden Markov model, and various IOHMMs. The experiments are performed on modeling the returns of market and sector indices. Note that the unconditional Gaussian estimates the first moment with the historical average. The results show that, although for the first moment the historical average gives the best results, for the higher moments, the IOHMMs yielded significantly better performance, as measured by the out-of-sample likelihood.
1 Introduction

Hidden Markov Models (HMMs) are statistical models of sequential data that have been used successfully in many machine learning applications, especially for speech recognition. Recently, HMMs have been applied to a variety of applications outside of speech recognition, such as handwriting recognition (Nag, Wong and Fallside, 1986; Kundu and Bahl, 1988; Matan et al., 1992; Ha et al., 1993; Schenkel et al., 1993; Schenkel, Guyon and Henderson, 1995; Bengio et al., 1995), pattern recognition in molecular biology (Krogh et al., 1994; Baldi et al., 1995; Chauvin and Baldi, 1995; Karplus et al., 1997; Baldi and Brunak, 1998), and fault-detection (Smyth, 1994). Input-Output Hidden Markov Models (IOHMMs) (Bengio and Frasconi, 1995) (or Conditional HMMs) are HMMs for which the emission and transition distributions are conditional on another sequence, called the input sequence. In that case, the observations modeled with the emission distributions are called outputs, and the model represents the conditional distribution of an output sequence given an input sequence. In this paper we will apply synchronous IOHMMs, for which input and output sequences have the same length. See (Bengio and Bengio, 1996) for a description of asynchronous IOHMMs, and see (Bengio, 1996) for a review of Markovian models in general (including HMMs and IOHMMs), and (Cacciatore and Nowlan, 1994) for a form of recurrent mixture of experts similar to IOHMMs.

An IOHMM is a probabilistic model with a certain fixed number of states corresponding to different conditional distributions of the output variables given the input variables, and with transition probabilities between states that can also depend on the input variables. In the unconditional case, we obtain a Hidden Markov Model (HMM). An IOHMM can be used to predict the expected value of the output variables given the current input variables and the past input/output pairs. The most likely state sequence corresponds to a segmentation of the past sequence into regimes (each one associated to one conditional distribution, i.e., to one state), which makes it attractive to model financial or economic data in which different regimes are believed to have existed.

Previous work on using Markov models to represent the non-stationarity in economic and financial time-series due to the business cycle are promising (Hamilton, 1989; Hamilton, 1988) and have generated a lot of interest and generalizations (Diebold, Lee and Weinbach, 1993; Garcia and Perron, 1995; Garcia and Schaller, 1995; Garcia, 1995; Sola and Driffield, 1994; Hamilton, 1996). In the experiments described here, the conditional dependency is not restricted to an affine form but includes non-linear models such as multi-layer artificial neural networks. Artificial neural networks have already been used to model some components of the business cycle (Bramson and Hoptroff, 1990; Moody, Levin and Rehfuss, 1993), but not using an IOHMM or conditional Markov-switching model.

2 Summary of Learning Algorithms for IOHMMs

The model of the multivariate discrete time-series data is a mixture of several models, each associated to one of several states, or regimes. In each of these regimes, the relation between
certain variables of interest may be different and is represented by a different conditional probability model. For each of these regime-specific models, we can use a multi-layer artificial neural network or another conditional distribution model (depending on the nature of the variables). The sequence of states (or regimes) is itself assumed to be a Markov chain, with transition probabilities that may change with time and depend on the currently observed (or input) variables. This model is an extension of Hidden Markov Models (HMMs) to the case of modeling the conditional distribution of an output sequence given an input sequence, and it is therefore called an Input/Output Hidden Markov Model (IOHMM). Whereas HMMs represent a distribution $P(y_T^T)$ of sequences of (output) observed variables $y_T^T = y_1, y_2, \ldots, y_T$, IOHMMs represent a conditional distribution $P(y_T^T|x_T^T)$, given an observed input sequence $x_T^T = x_1, x_2, \ldots, x_T$. In the asynchronous case, the lengths of the input and output sequences may be different. See (Bengio, 1996) for a more general discussion of Markovian models which include IOHMMs.

2.1 The Model and Its Likelihood

As in HMMs, that distribution is very much simplified, by introducing a discrete state variable, $q_T^T$, and a joint model $P(y_T^T, q_T^T|x_T^T)$, along with two crucial conditional independence assumptions:

$$P(y_t|q_t^t, y_{t-1}^t, x_t^t) = P(y_t|q_t, x_t)$$  \hspace{1cm} (1)

$$P(q_{t+1}|q_t^t, y_t^T, x_t^T) = P(q_{t+1}|q_t, x_t)$$  \hspace{1cm} (2)

In simple terms, the state variable $q_t$, jointly with the current input $x_t$, summarize all the relevant past values of the observed and hidden variables when one tries to predict the value of the observed variable $y_t$, or of the next state $q_{t+1}$.

Because of the above independence assumptions, the joint distribution of the hidden and observed variables can be much simplified, as follows:

$$P(y_T^T, q_T^T|x_T^T) = P(q_1) \prod_{t=1}^{T-1} P(q_{t+1}|q_t, x_t) \prod_{t=1}^{T} P(y_t|q_t, x_t)$$  \hspace{1cm} (3)

The joint distribution is therefore completely specified in terms of

1. the initial state probabilities $P(q_1)$,

2. the transition probabilities model $P(q_t|q_{t-1}, x_t)$ and,

3. the emission probabilities model $P(y_t|q_t, x_t)$.

In our experiments we have arbitrarily chosen one of the states (state 0) to be the “initial state”, i.e., $P(q_1 = 0) = 1$ and $P(q_1 > 0) = 0$.

Following equation 3, the computation of $P(y_T^T, q_T^T|x_T^T)$ is therefore straightforward (done in time $O(T)$). However, $q_1^T$ is not observed, and we are really interested in representing
the distribution \( P(y_T^1|x_T^1) \). Simply marginalizing the joint distribution yields an exponential number of terms:

\[
P(y_T^1) = \sum_{q_T^1} P(y_T^1, q_T^1)
\]

In the case of discrete states, there is fortunately an efficient recursive way to compute the above sum, based on a factorization of the probabilities that takes advantage of the Markov property of order 1. The recursion is not on \( P(y_T^1|x_T^1) \) itself but on \( P(y_T^1, q_T|x_T^1) \), i.e., the probability of observing a certain subsequence while the state takes a particular value at the end of that subsequence:

\[
P(y_T^1, q_T|x_T^1) = P(y_T^1|y_T^{t-1}, q_T, x_T^1) P(y_T^{t-1}, q_T, x_T^1)
= P(y_T^1|q_T, x_T) \sum_{q_{t-1}} P(y_T^{t-1}, q_{t-1}|x_T^1)
= P(y_T^1|q_T, x_T) \sum_{q_{t-1}} P(q_T|q_{t-1}, y_T^{t-1}, x_T^1) P(y_T^{t-1}, q_{t-1}|x_T^1)

\]

\[
P(y_T^1, q_T|x_T^1) = P(y_T^1|q_T, x_T) \sum_{q_{t-1}} P(q_T|q_{t-1}, x_T) P(y_T^{t-1}, q_{t-1}|x_T^1)
\]

where we used the two Markov assumptions (on the observed variable and on the state) respectively to obtain the second and last equation above. The recursion can be initialized with \( P(y_1, q_1|x_T^1) = P(y_1|q_1, x_1) P(q_1) \), using the initial state probabilities \( P(q_1) \). This recursion is central to many algorithms for HMMs, and is often called the forward phase. Let us note \( y_T^p(p) \) for the \( p \)-th sequence of a training data set, of length \( T_p \). The above recursion allows to compute the likelihood function \( l(\theta) = \prod_p P(y_T^p(p)|x_T^1, \theta) \), where \( \theta \) are parameters of the model which can be tuned in order to maximize the likelihood over the training sequences \( y_T^p(p) \). The computational cost of this recursion is \( O(Tm) \) when \( T \) is the length of a sequence and \( m \) is the number of non-zero transition probabilities at each time step, i.e., \( m \leq J^2 \) (where \( J \) is the number of values that the state variable \( q_T \) can take).

Once \( P(y_T^1, q_T|x_T^1, \theta) \) is obtained, one can readily compute the likelihood \( P(y_T^1|\theta) \) for each sequence as follows:

\[
P(y_T^1|\theta) = \sum_{q_T} P(y_T^1, q_T|x_T^1, \theta).
\]

Note that we sometimes drop the conditioning of probabilities on the parameters \( \theta \) unless the context would make that notation ambiguous.

### 2.2 EM and GEM for IOHMMs

In this section we will sketch the application of the EM (Expectation-Maximization) algorithm (Dempster, Laird and Rubin, 1977) to HMMs (Baum and Eagon, 1967; Baum et al., 1970; Baum, 1972) and IOHMMs. The papers by Baum et al. present a special case of the EM algorithm applied to discrete emissions HMMs, but were written before the general version of the EM algorithm was described (Dempster, Laird and Rubin, 1977).
The basic idea of the EM algorithm is to use a hidden variable whose joint distribution with the observed variable is “simpler” than the marginal distribution of the observed variable itself. In HMMs and IOHMMs, the hidden variable is the state path \( q_1^T \). We have already seen that \( P(y_1^T, q_1^T) \) is simpler to compute and represent than \( P(y_1^T) = \sum_{q_1^T} P(y_1^T, q_1^T) \). Because the hidden variable is not given, the EM algorithm looks at the expectation (over all values of the hidden variable) of the log-probability of the joint distribution. This expectation, called the auxiliary function, is conditioned on the previous values of the parameters, \( \theta^k \), and on the training observations. The E-Step of the algorithm consists in forming this conditional expectation:

\[
F(\theta|\theta^k) = E_Q[\log P(Y, Q|X, \theta) \mid Y, X, \theta^k]
\]  
(6)

where \( E_Q \) is the expectation over \( Q \), \( Y = \{y_1^T(1), \ldots, y_1^T(N)\} \) is the set of \( N \) output sequences, and similarly \( X \) and \( Q \) are respectively the sets of \( N \) input and \( N \) state sequences. The EM algorithm is an iterative algorithm successively applying the E-Step and the M-step. The M-Step consists in finding the parameters \( \theta \) which maximize the auxiliary function. At the \( k \)th iteration,

\[
\theta^{k+1} = \arg\max_\theta F(\theta|\theta^k).
\]  
(7)

It can be shown (Dempster, Laird and Rubin, 1977) that an increase of \( F \) brings an increase of the likelihood, and this algorithm converges to a local maximum of the likelihood, \( P(Y|X, \theta) \). When the above maximization cannot be done exactly (but \( F \) increases at each iteration), we have a GEM (Generalized EM) algorithm. The maximization can in general be done by solving the system of equations

\[
\frac{\partial F(\theta|\theta^k)}{\partial \theta} = 0
\]  
(8)

For HMMs, IOHMMs and state space models with simple enough emission and transition distributions, this can be done analytically. We will discuss here the case of discrete states, where the expectation in equation 6 corresponds to a sum over the values of the state variable, and the solution of equation 8 can be obtained efficiently with recursive algorithms. To see this, we will first rewrite the joint probability of states and observations by introducing indicator variables \( z_{i,t} \) with value 1 when \( q_t = i \) and 0 otherwise:

\[
\log P(y_1^T, q_1^T | x_1^T, \theta) = \sum_{t,i} z_{i,t} \log P(y_t|q_t=i, x_t, \theta) + \sum_{t,i,j} z_{i,t} z_{j,t-1} \log P(q_t=i|q_{t-1}=j, x_t, \theta)
\]

The overall joint log-probability for the whole training set is a sum over the training sequences of the above sum. Moving the expectation in equation 6 inside these sums, and ignoring the \( p \) indices for sequences within the training set (which would make the notation very heavy):

\[
F(\theta|\theta^k) = \sum_{p,t,i} E_Q[z_{i,t} | x_1^T, y_1^T, \theta^k] \log P(y_t|q_t=i, x_t, \theta)
\]

\[+ \sum_{p,t,i,j} E_Q[z_{i,t}, z_{j,t-1} | x_1^T, y_1^T, \theta^k] \log P(q_t=i|q_{t-1}=j, x_t, \theta)
\]

Note how in this expression the maximization of \( F \) with respect to the parameters \( \theta \) of the emission and transition probabilities have been completely decoupled in two separate sums.
To simplify the notation (and because they are often ignored in practice by forcing all state sequences to start from the same state) we have ignored the initial state probabilities. In the M-Step, the problem becomes one of simple likelihood maximization for each of the different types of distributions, but with weights for each the probabilities in the above sums. These weights are the state posterior probabilities

\[ P(q_t=i|x_1^T, y_1^T, \theta^k) = E_{Q}[z_{i,t}|x_1^T, y_1^T, \theta^k], \]

and the transition posterior probabilities

\[ P(q_t=i, q_{t-1}=j|x_1^T, y_1^T, \theta^k) = E_{Q}[z_{i,t}, z_{j,t-1}|x_1^T, y_1^T, \theta^k]. \]

Let us now see how these posterior probabilities, which we will note \( P(q_t|x_1^T, y_1^T) \) and \( P(q_t, q_{t-1}|x_1^T, y_1^T) \) to lighten the notation, can be computed with the Baum-Welch forward and backward recursions (Baum and Eagon, 1967; Baum et al., 1970; Baum, 1972).

We have already introduced the forward recursion (equation 5), which yields \( P(y_1^T, q_t|x_1^T) \) recursively. Note that \( P(y_1^T, q_t|x_1^T) \) can be normalized to perform the filtering operation, i.e., to obtain \( P(q_t|y_1^T, x_1^T) \).

Using the two Markov assumptions (the equivalent of equations 1 and 2 conditioned on the input sequence), the Baum-Welch backward recursion can be obtained:

\[
P(y_{T+1}^T|q_T, x_1^T) = \sum_{q_{T-1}} P(y_{t+1}^T, q_{t+1}, x_{t+1}^T)P(q_{t+1}|q_t, x_{t+1})P(y_{t+2}^T|q_{t+1}, x_1^T)
\]

By multiplying the results of the forward and backward recursion and normalizing by the output sequence probability, we obtain the state posteriors (i.e., the smoothed estimates of the state distribution):

\[
P(q_t|x_1^T, y_1^T) = \frac{P(y_1^T, q_t|x_1^T)P(y_{T+1}^T|q_T, x_1^T)}{P(y_1^T)}.
\]

Similarly, the transition posteriors can be obtained from these two recursions and from the emission and transition probabilities as follows:

\[
P(q_t, q_{t-1}|x_1^T, y_1^T) = \frac{P(y_1^T|q_t, x_1^T)P(y_{T+1}^T, q_{T+1}|x_1^T)P(y_{T+2}^T|q_{T+1}, x_1^T)P(q_{T+1}|q_{T-1}, x_1^T)}{P(y_1^T)}
\]

Some care must be taken in performing the forward and backward recursions in order to avoid numerical over or under flow (usually this is accomplished by performing the computation in a logarithmic scale with a small base).

The details of the parameter update algorithm depends on the particular form of the emission and transition distributions. If they are discrete, in the exponential family, or a mixture thereof, then exact (and simple) solutions for the M-Step exist (by using a weighted form of the maximum likelihood solutions for these distributions). See the appendices of (Lauzon, 1999) for detailed derivations of the updates for the kinds of emission distributions used in
the experiments of this paper. For other distributions such as those incorporating an artificial neural network to compute conditional discrete probabilities or the conditional mean of a Gaussian, one can use a GEM algorithm or the maximization of the observations likelihood by numerical methods such as gradient ascent. Note that maximizing \( F \) by gradient ascent is equivalent to maximizing the likelihood by gradient ascent (see (Lauzon, 1999) for a proof). This can be shown by noting that the quantities computed in the backward pass are in fact gradients of the likelihood with respect to the quantities computed in the forward pass:

\[
P(y_{t+1}^T | q_t, x_1^T) = \frac{\partial P(y_1^T)}{\partial P(y_1^T, q_t|x_1^T)}.
\]

In the experiments described in this paper, we have used for the emission distributions either Gaussians, conditional linear Gaussians, or multi-layer neural networks. In the first two cases we can use the EM algorithm whereas in the latter case we used a gradient-based numerical optimization method (the conjugate gradients method).

### 2.3 Using a Trained Model

Once the model is trained, it can be used in several ways. If inputs and outputs up to time \( t \) (or just inputs up to time \( t \)) are given, we can compute the probability of being in each one of the states (i.e., regimes). Using these probabilities, we can make predictions on the output (e.g., decision, classification, or prediction) variables for the current time step. This prediction is obtained by taking a linear combination of the predictions of the individual regime models, and the weights of this linear combination are simply the probabilities of being in each one of the states, given the past input/output sequence and the current input:

\[
P(q_t = i | x_1^t, y_1^{t-h}) = \frac{P(q_t = i, y_1^{t-h} | x_1^t)}{\sum_j P(q_t = i, y_1^{t-h} | x_1^t)}
\]

where we can compute recursively

\[
P(q_t = i, y_1^{t-h} | x_1^t) = \sum_j P(q_t = i, q_{t-1} = j | x_t) P(q_{t-1} = j, y_1^{t-h} | x_1^{t-1})
\]

starting from the \( P(q_{t-h} = i, y_1^{t-h} | x_1^{t-h}) \) computed in the forward phase (equation 5).

We can also find the most likely sequence of regimes up to the current time step, using the Viterbi algorithm (Viterbi, 1967) (see (Lauzon, 1999) in the context of IOHMMs). The model can also be used as an explanatory tool: given both past and future values of the variables, we can compute the probability of being in each one of the regimes at any given time step \( t \). If a model of the input variables is built, then we can also use the model to make predictions about the future expected state changes. What we will obtain is not just a prediction for the next expected state change, but a distribution over these changes, which also gives a measure of confidence in these predictions. Similarly for the output (prediction or decision) variables, we can train the model to compute the expected value of this variable (given the current state and input), but we can also train it to model the whole distribution (which contains information about the expected error in a particular prediction).
3 Experimental Setup

In this section, we describe the setup of the experiments performed on financial data, for modeling the future returns of Canadian stock market indices. Let us introduce some notation. Let

\[ r_t = \frac{\text{value}_t}{\text{value}_{t-1}} - 1 \]

be a discrete return series (the ratio of the value of an asset at time \( t \) over its value at time \( t - 1 \), which in the case of stocks includes dividends and capital gains). Let \( \bar{r}_{h,t} \) be a moving average of \( h \) successive values of \( r_t \):

\[ \bar{r}_{h,t} = \frac{1}{h} \sum_{s=t-h+1}^{t} r_s. \]

In the experiments, we will measure performance in two ways: looking at time \( t \) at how well the conditional expectation (first moment) of

\[ y_t = \bar{r}_{h,t+h} \]

is modeled, and looking at how well the overall conditional distribution of \( y_t \) is modeled. The distribution is conditioned on the input series \( x_t \). We call the offset \( h \) the horizon because it is the number of time steps from a prediction to the corresponding observation. The horizon in the various experiments were 1, 5, and 12.

In the experiments all the models \( \hat{P} \) are trained at each time step \( t \) to maximize the conditional likelihood of the training data, \( \hat{P}(y_{1:t-h}^{t-h}, x_{1:t-h}, \theta) \), yielding parameters \( \theta_t \). We then use the particular form of the model to infer \( \hat{P}(y_t | y_{1:t-h}^{t-h}, x_t, \theta_t) \), and we measure out-of-sample

- **SE**, the squared error: \( \frac{1}{t} \sum_{t-h+1}^{t} (y_t - \hat{E}[y_t | y_{1:t-h}^{t-h}, x_t, \theta_t])^2 \), and
- **NLL**, the negative log-likelihood: \( -\log \hat{P}(y_t | y_{1:t-h}^{t-h}, x_t, \theta_t) \),

where \( \hat{E}[] | \theta \) is the expectation under the model distribution \( \hat{P}([] | \theta) \). The above logarithm is for making an additive quantity, and the minus is for getting a quantity that should be minimized, like the squared error.

In the experiments we have performed experiments on three types of return series:

1. **Daily Market index returns**: we used daily returns data from the TSE300 stock index, from January 1977 to April 1998, for a total of 4154 days.

2. **Monthly Market index returns**: we used 479 monthly returns from the TSE300 stock index, from February 1956 to January 1996. We also used 24 economic and financial time-series as explanatory (input) variables in some of the models.

3. **Sector returns**: we used monthly returns data for the main 14 sectors of the TSE, from January 1956 to June 1997 inclusively, for a total of at most 497 months (some sectors started later).
3.1 Models Compared

In the experiments, we have compared the following models. All can be considered special cases of IOHMMs, although in some cases an analytic solution to the estimation exists, in other cases the EM algorithm can be applied, while in others only GEM or gradient-based optimization can be performed.

- **Gaussian** model: we have used a diagonal Gaussian model (i.e., not modeling the covariances among the assets). The number of free parameters is $2n$ for $n$ assets. In the experiments $n = 1$ (TSE300) or $n = 14$ (sector indices). There is an analytical solution to the estimation problem. This model is the basic "reference" model to which the other models will be compared.

$$
P_g(Y = y | x, \theta) = \prod_{i=1}^{n} N(y_i; \mu_i, \sigma_i^2)
$$

where $N(x; e, v)$ is the Normal probability of observation $x$ with expectation $e$ and variance $v$, $\theta = (\mu, \sigma)$, and $\mu = (\mu_1, \ldots, \mu_n)$, $\sigma = (\sigma_1, \ldots, \sigma_n)$. This is like an "unconditional" IOHMM with a single state.

- **Mixture of Gaussians** model: we have used a 2-component mixture in the experiments, whose emissions are diagonal Gaussians:

$$
P_m(Y = y | x) = \sum_{j=1}^{J} w_j P_g(Y = y | x, p_j)
$$

where $\theta = (w, p_1, p_2, \ldots, p_J)$, $p_j$ is the vector with the means and deviations for the $j$-th component, and $\sum_j w_j = 1$, $w_j \geq 0$. $J = 2$ in the experiments, to avoid overfitting. The number of free parameters is $J - 1 + 2nJ$. This is like an "unconditional" IOHMM with $J$ states and shared transition probabilities (all the states share the same transition probability distribution to the next states). This means that the model has no "dynamics": the probability of being in a particular "state" does not depend on what the previous state was.

- **Conditional Linear Gaussian** model: this is basically an ordinary regression, i.e., a Gaussian whose variance is unconditional while its expectation is conditional:

$$
P_l(Y = y | x, \theta) = \prod_{i=1}^{n} N(y_i; b_i + \sum_{k=1}^{K} A_{i,k} x_k, \sigma_i^2)
$$

where $\theta = (A, b, \sigma)$, $x_k$ denotes the $k$-th element of the input vector. The number of free parameters is $n(2 + K)$. In the experiments, $K = 1, 2, 4, 6, 14, 24$ have been tried for various input features. See more details below in section 4.
Mixture of Conditional Linear Gaussians: this combines the ideas from the previous two models, i.e., we have a mixture, but the expectations are linearly conditional on the inputs. We have used separate inputs for each sector prediction:

\[ P_{lm}(Y = y|x, \theta) = \sum_{j=1}^{J} w_j P_l(Y = y|x, p_j) \]

where \( \theta = (w, p_1, p_2, \ldots, p_J) \), \( p_j \) is the parameter vector for a conditional linear Gaussian model, and as usual \( w_j \geq 0 \), \( \sum_j w_j = 1 \). The number of free parameters is \( J \) times the number of free parameters of the conditional linear Gaussian model, i.e., \( nJ(2 + K) \) in the experiments. \( J = 2 \) in the experiments, and \( K = 1, 2, 4, 6, 14, 24 \) have been tried. This is like an IOHMM whose transition probabilities are shared across all states. Like for the mixture of Gaussians, this means that the model has no “dynamics”: the probability of being in a particular “state” does not depend on what the previous state was. Note that this model is also called a mixture of experts (Jacobs et al., 1991).

HMM: this is like the Gaussian mixture except that the model has a dynamics, modeled by the transition probabilities. In the experiments, there are \( J = 2 \) states. The number of free parameters is \( 2nJ + J(J - 1) \). This is like an “unconditional” IOHMM. Each emission distribution is an unconditional multinomial: \( P(q_t = i|q_{t-1} = j, x_t) = A_{i,j} \). Each emission distribution is a diagonal Gaussian: \( P(y_t = i|x_t) = \prod_{j=1}^{n} N(y_t; \mu_{i,j}, \sigma^2_{i,j}) \).

Linear IOHMM: this is like the mixture of experts, but with dynamics (modeled by the transition probabilities), or this is like an HMM in which the Gaussian expectations are affine functions of the input vector. We have not used conditional transition probabilities in the experiments. Again we have used \( J = 2 \) states, and the number of inputs \( K \) varies. The number of free parameters is \( nJ(2 + K) + J(J - 1) \).

MLP IOHMM: this is like the linear IOHMM except that the expectations are non-linear functions of the inputs, using a Multi-Layer Perceptron with \( K \) inputs and a single hidden layer with \( H \) hidden units (\( H = 2, 3, \ldots \) depending on the experiment). In one set of experiments we have used the same MLP for all the \( n \) assets (i.e., a single network per state is used, with \( n \) outputs, with \( n \) inputs associated to the \( n \) assets). In the other experiments, a separate network (with 1 output and \( K \) inputs) is used for each asset, so there \( n \) MLPs per state. In the first case the number of free parameters is \( J(J - 1) + J(2n + (1 + H)n + H(n + 1)) \), and in the second case it is \( J(J - 1) + Jn(2 + H + H(K + 1)) \).

3.2 Performance Measurements

In the experiments on sectors, there are several assets, with corresponding return series. Different models are trained separately on each of the assets, and the results reported below concern the average performance over all the assets. We have measured the average squared
error (MSE) and the average negative log-likelihood (NLL). We have also estimated the variance of these averages, and the variance of the difference between the performance for one model and the performance for a reference model, as described below. Using the latter, we have made hypothesis tests on the hypothesis that the two compared models have identical true generalization error (over the unknown data distribution).

For this purpose, we have used an estimate of variance that takes into account the dependencies among the errors at successive time steps. Let \( e_t \) be a series of errors (or error differences), which are maybe not i.i.d. Their average is

\[
\bar{e} = \frac{1}{n} \sum_{t=1}^{n} e_t.
\]

We are interested in estimating its variance

\[
Var[\bar{e}] = \frac{1}{n^2} \sum_{t=1}^{n} \sum_{t'=1}^{n} Cov(e_t, e_{t'}).
\]

Since we are dealing with a time-series, and because we do not know how to estimate independently and reliably all the above covariances, we will assume that the error series is covariance-stationary and that the covariance dies out as \(|t - t'|\) increases. This can be verified empirically by drawing the autocorrelation function of the \( e_t \) series. The covariance stationarity implies that

\[
Cov(e_t, e_{t'}) = \gamma_{|t-t'|}.
\]

An unbiased and convergent estimator (Priestley, 1981) of the above variance is then the following:

\[
\hat{\gamma} = \frac{1}{n} \sum_{k=-[\Delta_n]}^{[\Delta_n]} (1 - |k|/\Delta_n) \gamma_k,
\]

where \( \lim_{n \to \infty} \Delta_n = \infty \) and \( \lim_{n \to \infty} \Delta_n/n = 0 \); we have used \( \Delta_n = \sqrt{n} \).

The \( \gamma's \) are estimated from the sample covariances (here for \( k > 0 \)):

\[
\gamma_k = \frac{1}{n-k} \sum_{t=1}^{n-k} e_t e_{t+k}.
\]

Because there are generally strong dependencies between the errors of different models, we have found that much better estimates of variance were obtained by analyzing the differences of squared errors, rather than computing a variance separately for each average:

\[
Var[\bar{e}_i^A - \bar{e}_i^R] = Var[\bar{e}_i^A] + Var[\bar{e}_i^R] - 2Cov[\bar{e}_i^A, \bar{e}_i^R]
\]

where \( \bar{e}_i^A \) is the average error of model \( A \) on asset \( i \), and \( R \) is the reference model.

The average over assets of the performance measure is simply the average of the average errors for each asset:

\[
\bar{e}^A = \frac{1}{n} \sum_{i=1}^{n} \bar{e}_i^A.
\]
To combine the variances obtained as above for each of the assets, we decompose the variance of the average over assets as follows:

\[
Var\left[\frac{1}{n} \sum_{i=1}^{n} (\bar{\varepsilon}_i^A - \bar{\varepsilon}_i^R)\right] = \frac{1}{n^2} \sum_{i=1}^{n} Var[\bar{\varepsilon}_i^A - \bar{\varepsilon}_i^R] + \frac{2}{n^2} \sum_{i=1}^{n} \sum_{j<i} Cov[\bar{\varepsilon}_i^A - \bar{\varepsilon}_i^R, \bar{\varepsilon}_j^A - \bar{\varepsilon}_j^R]
\]

where the covariances are estimated by the sample covariances of the averages of the error differences:

\[
Cov[\bar{\varepsilon}_i^A - \bar{\varepsilon}_i^R, \bar{\varepsilon}_j^A - \bar{\varepsilon}_j^R] \approx \frac{1}{T(T-1)} \sum_{t=1}^{T} (\bar{\varepsilon}_{t,i}^A - \bar{\varepsilon}_{t,i}^R - \bar{\varepsilon}_{t,j}^A + \bar{\varepsilon}_{t,j}^R)(\bar{\varepsilon}_{t,i}^A - \bar{\varepsilon}_{t,i}^R + \bar{\varepsilon}_{t,j}^A - \bar{\varepsilon}_{t,j}^R)
\]

In the tables below we give the p-value of the null hypothesis that model A is not better than model B (where model B is a reference model, usually the Gaussian model). In the case that \(\bar{\varepsilon}_A < \bar{\varepsilon}_B\), the alternative hypothesis is that A is better than B (so we use a single-sided test, with p-value = \(P(E[\varepsilon_B - \varepsilon_A] \geq \bar{\varepsilon}_B - \bar{\varepsilon}_A)\)). In the opposite case, we consider the converse test (with the alternative hypothesis being that B is better than A). To compute the p-values we assume that the average or the average difference is a normal variate, whose variance is estimated as explained above using the auto-covariances of the series.

### 4 Experimental Results

<table>
<thead>
<tr>
<th>Model</th>
<th>(\mu) (MSE) \times 10^{-6}</th>
<th>(\sigma) (MSE) \times 10^{-6}</th>
<th>(\mu) (NLL) \times 10^{-2}</th>
<th>(\sigma) (NLL) \times 10^{-2}</th>
<th>p-value (%) vs Gaussian</th>
<th>p-value(%) vs IOHMM (MLP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>6.733</td>
<td>1.29</td>
<td>-4.171</td>
<td>6.38</td>
<td>&lt; 1e-7</td>
<td></td>
</tr>
<tr>
<td>Gaussian Mixture</td>
<td>6.733</td>
<td>1.29</td>
<td>-4.241</td>
<td>4.41</td>
<td>0.41</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>HMM</td>
<td>6.741</td>
<td>1.30</td>
<td>-4.257</td>
<td>4.60</td>
<td>0.02</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Linear (K=4)</td>
<td>6.867</td>
<td>1.41</td>
<td>-4.180</td>
<td>6.23</td>
<td>1.47</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture of Linear (K=4)</td>
<td>6.794</td>
<td>1.34</td>
<td>-4.261</td>
<td>4.14</td>
<td>0.25</td>
<td>1.31e-3</td>
</tr>
<tr>
<td>IOHMM (Linear K=4)</td>
<td>6.937</td>
<td>1.39</td>
<td>-4.277</td>
<td>5.85</td>
<td>&lt; 1e-7</td>
<td>5.80e-3</td>
</tr>
<tr>
<td>IOHMM (MLP H=2,K=4)</td>
<td>6.957</td>
<td>1.35</td>
<td>-4.332</td>
<td>5.21</td>
<td>&lt; 1e-7</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Results on modeling the next **TSE300 5-day** returns every day.

Each of the tables in this section gives the result of a series of experiments in which the output variable is the same, and we vary the choice of model and input feature vectors (when they are used). We used measures of return variation in the input features:

\[
v_t = |r_t - \bar{r}_{t,t}|
\]

where \(\bar{r}_{t,t}\) is the historical average of the returns up to time \(t\), and \(\bar{r}_{d,t}\) is a \(d\)-time steps moving average of \(v_t\) available at time \(t\).

The following target output variables have been considered:
Table 2: Results on modeling the next TSE300 12-month returns every month.

- TSE300 5-day return: we want to predict the distribution of the 5-day average future market index return $y_t = \bar{r}_{5,t+5}$ at time $t$ ($t$ on a daily scale); horizon $h = 5$.

- TSE300 12-month return: we want to predict the distribution of the 1-year average future market index return $y_t = \bar{r}_{12,t+12}$ at time $t$ ($t$ on a monthly scale); horizon $h = 12$.

- TSE300 1-month return: we want to predict the distribution of next month’s market index return $y_t = r_{t+1}$ at time $t$ ($t$ on a monthly scale); horizon $h = 1$.

- 14 sectors 1-month return: we want to predict the individual distributions of next month’s return for each of the 14 sector indices of the Toronto Stock Exchange, $y_t = r_{t+1}$, at time $t$ ($t$ on a monthly scale); horizon $h = 1$.

The following input variables have been considered in the experiments:

- 1 input: the current return $r_t$ associated to the output series.

- 2 inputs: the current return $r_t$ and current variation $v_t$.

- 4 inputs: the current return $r_t$, current variation $v_t$, and the average past 4 returns and variations $\bar{r}_{4,t}$ and $\bar{v}_{4,t}$.

- 12 inputs: the current return $r_t$, current variation $v_t$, the average past 4 returns and variations $\bar{r}_{4,t}$ and $\bar{v}_{4,t}$, and the average past 12 returns and variations $\bar{r}_{12,t}$ and $\bar{v}_{12,t}$.

- 14 inputs: the 14 current sector returns.

- 24 inputs: 24 economic and financial indicators of the Canadian financial markets.

In the tables, the column entitled “$\mu$ (MSE)” gives the average squared error, the column entitled “$\sigma$ (MSE)” gives the estimated variance of that average, and similarly for the negative log-likelihood (NLL). The performance figure in bold is the lowest in the column. The column

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu$ (MSE) $\times 10^{-5}$</th>
<th>$\sigma$ (MSE) $\times 10^{-5}$</th>
<th>$\mu$ (NLL)</th>
<th>$\sigma$ (NLL)</th>
<th>p-value (%)</th>
<th>p-value (%) vs Gaussian</th>
<th>IOHMM (MLP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>9.694</td>
<td>2.13</td>
<td>-2.886</td>
<td>0.12</td>
<td>&lt; 1e-7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixture Gaussian</td>
<td>9.694</td>
<td>2.13</td>
<td>-2.888</td>
<td>0.12</td>
<td>28.14</td>
<td>&lt; 1e-7</td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>10.00</td>
<td>2.17</td>
<td>-2.971</td>
<td>0.12</td>
<td>0.0043</td>
<td>5.40e-3</td>
<td></td>
</tr>
<tr>
<td>Cond. Linear (K=1)</td>
<td>10.13</td>
<td>2.54</td>
<td>-2.901</td>
<td>0.14</td>
<td>38.71</td>
<td>2.51e-3</td>
<td></td>
</tr>
<tr>
<td>Mixture of Linear (K=1)</td>
<td>10.04</td>
<td>2.49</td>
<td>-2.912</td>
<td>0.13</td>
<td>27.93</td>
<td>4.17e-4</td>
<td></td>
</tr>
<tr>
<td>IOHMM (Linear K=1)</td>
<td>10.15</td>
<td>2.54</td>
<td>-2.936</td>
<td>0.14</td>
<td>10.53</td>
<td>2.95e-4</td>
<td></td>
</tr>
<tr>
<td>IOHMM (MLP H=2, K=1)</td>
<td>10.42</td>
<td>2.35</td>
<td>-3.171</td>
<td>0.12</td>
<td>&lt; 1e-7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3: Results on modeling the next TSE300 1-month returns.

<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu$ (MSE) $\times 10^{-3}$</th>
<th>$\sigma$ (MSE) $\times 10^{-4}$</th>
<th>$\mu$ (NLL) $\times 10^{-2}$</th>
<th>$\sigma$ (NLL) $\times 10^{-2}$</th>
<th>p-value (%) vs Gaussian</th>
<th>p-value (%) vs IOHMM (K=24)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>1.218</td>
<td>2.04</td>
<td>-1.603</td>
<td>7.91</td>
<td>3.49e-3</td>
<td></td>
</tr>
<tr>
<td>Mixture Gaussian</td>
<td>1.218</td>
<td>2.04</td>
<td>-1.675</td>
<td>6.81</td>
<td>0.28</td>
<td>0.046</td>
</tr>
<tr>
<td>HMM</td>
<td>1.230</td>
<td>2.06</td>
<td>-1.685</td>
<td>6.85</td>
<td>0.059</td>
<td>0.52</td>
</tr>
<tr>
<td>Cond. Linear (K=4)</td>
<td>1.245</td>
<td>1.97</td>
<td>-1.631</td>
<td>7.04</td>
<td>6.51</td>
<td>0.086</td>
</tr>
<tr>
<td>Cond. Linear (K=6)</td>
<td>1.285</td>
<td>2.06</td>
<td>-1.634</td>
<td>6.97</td>
<td>7.62</td>
<td>0.084</td>
</tr>
<tr>
<td>Mixture of Linear (K=4)</td>
<td>2.120</td>
<td>4.51</td>
<td>-1.728</td>
<td>6.91</td>
<td>0.18</td>
<td>7.69e-3</td>
</tr>
<tr>
<td>Mixture of Linear (K=6)</td>
<td>1.248</td>
<td>1.97</td>
<td>-1.740</td>
<td>5.78</td>
<td>0.025</td>
<td>0.72</td>
</tr>
<tr>
<td>Mixture of Linear (K=24)</td>
<td>1.325</td>
<td>2.18</td>
<td>-1.736</td>
<td>5.55</td>
<td>0.096</td>
<td>0.45</td>
</tr>
<tr>
<td>IOHMM (Linear K=4)</td>
<td>2.972</td>
<td>7.03</td>
<td>-1.815</td>
<td>8.74</td>
<td>0.57</td>
<td>0.17</td>
</tr>
<tr>
<td>IOHMM (Linear K=6)</td>
<td>1.313</td>
<td>2.22</td>
<td>-1.732</td>
<td>6.13</td>
<td>0.013</td>
<td>0.74</td>
</tr>
<tr>
<td>IOHMM (Linear K=24)</td>
<td>1.386</td>
<td>2.39</td>
<td>-1.736</td>
<td>5.44</td>
<td>0.13</td>
<td>0.32</td>
</tr>
<tr>
<td>IOHMM (MLP H=3, K=4)</td>
<td>2.267</td>
<td>4.77</td>
<td>-2.075</td>
<td>8.75</td>
<td>3.49e-3</td>
<td></td>
</tr>
<tr>
<td>IOHMM (MLP H=3, K=6)</td>
<td>1.251</td>
<td>1.99</td>
<td>-1.748</td>
<td>6.06</td>
<td>4.86e-3</td>
<td>0.090</td>
</tr>
<tr>
<td>IOHMM (MLP H=2, K=24)</td>
<td>1.360</td>
<td>2.18</td>
<td>-1.759</td>
<td>5.86</td>
<td>2.56e-3</td>
<td>0.25</td>
</tr>
</tbody>
</table>

entitled “p-value (%) vs Gaussian” gives the p-value (in percent) of a single-sided test for the null hypothesis that the model is not better than the Gaussian reference. If there is no “W” before the percentage the alternative hypothesis is that the model is better than the reference; if there is a “W” than the test is with respect to the alternative hypothesis that the reference is better than the model. Similarly, the column entitled “p-value (%) vs M” compares each of the model with model M (which is the one with the lowest NLL). For this column there never is a “W” since the comparison is always against the “best” model, so we are testing with respect to the alternative hypothesis that the given model is worse than the “best” model. For both p-value columns, bold figures underline the fact that the p-value is less than 5%, i.e., that we consider that the null hypothesis of no difference should be rejected. Since in all comparisons no model significantly improved on the Gaussian for predicting the first moment (i.e., with respect to mean squared error), we have only given the p-values for comparing the negative log-likelihoods. Note that the p-values take into account not only the variances of each model’s performance (in the $\sigma$ columns) but also their covariance, which is generally positive (hence the variance of their difference is smaller than the sum of their variance).

In Table 1 the results on predicting the 5-day return of the TSE300 are presented. All the conditional models use the 4 inputs variables. No model outperforms the Gaussian for predicting the first moment, but all of them outperform the Gaussian for predicting the distribution. As in all of the other experiments, the Gaussian model gives the lowest MSE
<table>
<thead>
<tr>
<th>Model</th>
<th>$\mu$ (MSE) $\times 10^{-3}$</th>
<th>$\sigma$ (MSE) $\times 10^{-4}$</th>
<th>$\mu$ (NLL) $\times 10^{-2}$</th>
<th>$\sigma$ (NLL) $\times 10^{-2}$</th>
<th>p-value (%) vs Gaussian</th>
<th>p-value (%) vs IOHMM (MLP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>2.017</td>
<td>1.93</td>
<td>-1.407</td>
<td>5.12</td>
<td>&lt; 1e-7</td>
<td></td>
</tr>
<tr>
<td>Mixture Gaussian</td>
<td>2.017</td>
<td>1.93</td>
<td>-1.455</td>
<td>4.58</td>
<td>&lt; 1e-7</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>HMM</td>
<td>2.015</td>
<td>1.93</td>
<td>-1.474</td>
<td>4.30</td>
<td>9.53e-5</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Cond. Linear (K=14)</td>
<td>2.490</td>
<td>1.96</td>
<td>-1.132</td>
<td>13.8</td>
<td>W4.32e-4</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture of Linear (K=14)</td>
<td>2.183</td>
<td>1.94</td>
<td>-1.585</td>
<td>4.07</td>
<td>2.98e-6</td>
<td>6.29</td>
</tr>
<tr>
<td>IOHMM (Linear K=14)</td>
<td>2.286</td>
<td>2.17</td>
<td>-1.402</td>
<td>10.2</td>
<td>W4756</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>IOHMM (MLP H=2, K=14)</td>
<td>2.088</td>
<td>1.82</td>
<td>-1.616</td>
<td>4.43</td>
<td>&lt; 1e-7</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Results on jointly modeling the next 14 sectors 1-month returns.

(error in prediction of the first moment), but the difference with the other models is not significant. For predicting the whole distribution, the best model is the non-linear IOHMM (with a MLP), and the difference with each of the other models is significant.

In table 2, the results of experiments on the monthly TSE300 data are presented, with a horizon of $h = 12$ months, i.e., the models predict every month the distribution of the next 12-month TSE300 market index return. Note that only the non-linear IOHMM and the HMM give a significantly better model than the Gaussian, and note that the non-linear IOHMM is significantly better than all the others.

In table 3, the results of experiments on the monthly TSE300 data are presented. Almost all the models (except two of the linear models) are significantly better than the Gaussian for modeling the density of the next month return. The best model is the linear IOHMM, and its performance is significantly significantly better than the performance of the other models. Note also that the models with the 24 inputs (which include not only technical but also economic variables) outperform the models with only technical variables in input.

Table 4 shows the results of experiments on jointly modeling the distribution of the next month’s return for the 14 sector indices. All the conditional models have 14 inputs, for the current return of each of the 14 sectors. The non-linear IOHMM yields the best results, significantly above all the other models except the mixture of linear experts (but the p-value is close to our threshold, at 6.3%).

In table 5, the results of the other experiments on the monthly 14 sectors of the TSE300 data are presented, where each sector has been trained individually. For predicting the distribution, the Gaussian is beaten significantly by all the other models. The best-performing model is one of the linear IOHMMs. It outperforms significantly all the models except with with an IOHMM with MLP (5 hidden units, 1 input) and the mixture of linear experts (with 6 inputs).
<table>
<thead>
<tr>
<th></th>
<th>$\mu$ (MSE) $\times 10^{-3}$</th>
<th>$\sigma$ (MSE) $\times 10^{-5}$</th>
<th>$\mu$ (NLL) $\times 10^{-2}$</th>
<th>$\sigma$ (NLL) $\times 10^{-2}$</th>
<th>p-value (%) vs Gaussian</th>
<th>p-value (%) vs Lin. IOHMM (K=6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>2.017</td>
<td>8.02</td>
<td>-1.408</td>
<td>1.87</td>
<td></td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture Gaussian</td>
<td>2.017</td>
<td>8.02</td>
<td>-1.419</td>
<td>1.85</td>
<td>2.53e-4</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>HMM</td>
<td>2.017</td>
<td>8.03</td>
<td>-1.426</td>
<td>1.83</td>
<td>7.74e-5</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Cond. Linear (K=1)</td>
<td><strong>2.013</strong></td>
<td>8.05</td>
<td>-1.417</td>
<td>1.85</td>
<td>0.31</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Cond. Linear (K=2)</td>
<td>2.030</td>
<td>8.11</td>
<td>-1.421</td>
<td>1.81</td>
<td>0.17</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Cond. Linear (K=4)</td>
<td>2.049</td>
<td>8.17</td>
<td>-1.425</td>
<td>1.79</td>
<td>0.073</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Cond. Linear (K=6)</td>
<td>2.064</td>
<td>8.31</td>
<td>-1.423</td>
<td>1.79</td>
<td>0.097</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture of Linear (K=1)</td>
<td>2.014</td>
<td>8.04</td>
<td>-1.430</td>
<td>1.81</td>
<td>5.96e-6</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture of Linear (K=2)</td>
<td>2.030</td>
<td>8.10</td>
<td>-1.436</td>
<td>1.77</td>
<td>5.96e-6</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>Mixture of Linear (K=4)</td>
<td>2.048</td>
<td>8.12</td>
<td>-1.458</td>
<td>1.65</td>
<td>&lt; 1e-7</td>
<td>0.024</td>
</tr>
<tr>
<td>Mixture of Linear (K=6)</td>
<td>2.061</td>
<td>8.20</td>
<td>-1.467</td>
<td>1.63</td>
<td>&lt; 1e-7</td>
<td>20.48</td>
</tr>
<tr>
<td>IOHMM (Linear K=1)</td>
<td>2.015</td>
<td>8.07</td>
<td>-1.431</td>
<td>1.82</td>
<td>&lt; 1e-7</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>IOHMM (Linear K=2)</td>
<td>2.031</td>
<td>8.10</td>
<td>-1.439</td>
<td>1.76</td>
<td>5.96e-6</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>IOHMM (Linear K=4)</td>
<td>2.050</td>
<td>8.10</td>
<td>-1.461</td>
<td>1.64</td>
<td>&lt; 1e-7</td>
<td>1.06</td>
</tr>
<tr>
<td>IOHMM (Linear K=6)</td>
<td>2.073</td>
<td>8.33</td>
<td>-1.470</td>
<td>1.63</td>
<td>&lt; 1e-7</td>
<td>&lt; 1e-7</td>
</tr>
<tr>
<td>IOHMM (MLP H=5, K=1)</td>
<td>2.022</td>
<td>8.05</td>
<td>-1.464</td>
<td>1.70</td>
<td>&lt; 1e-7</td>
<td>5.26</td>
</tr>
<tr>
<td>IOHMM (MLP H=5, K=2)</td>
<td>2.051</td>
<td>8.08</td>
<td>-1.446</td>
<td>1.64</td>
<td>&lt; 1e-7</td>
<td>8.94e-6</td>
</tr>
<tr>
<td>IOHMM (MLP H=3, K=4)</td>
<td>2.032</td>
<td>7.95</td>
<td>-1.445</td>
<td>1.69</td>
<td>&lt; 1e-7</td>
<td>2.68e-5</td>
</tr>
<tr>
<td>IOHMM (MLP H=2, K=6)</td>
<td>2.035</td>
<td>8.09</td>
<td>-1.443</td>
<td>1.69</td>
<td>&lt; 1e-7</td>
<td>&lt; 1e-7</td>
</tr>
</tbody>
</table>

Table 5: Results on modeling the TSE300 14 sectors (1-month) returns.

## 5 Conclusions

In this paper we have applied Input/Output Hidden Markov Models to financial time-series data in a number of comparative experiments aimed at measuring the expected generalization error of different types of models. The main conclusions from the experiments are the following:

- For predicting the expected future returns, none of the models we have tried performs better than the simple-minded historical average.

- For predicting the future distribution of the returns at a fixed horizon, we have found IOHMMs to perform significantly better than other models, more specifically:
  - For predicting the next 5-day return of the TSE300, the non-linear IOHMM (with MLPs) is significantly better than all the other models.
For predicting the next 12-month return of the TSE300, the non-linear IOHMM (with MLPs) is significantly better than all the other tested models.

For predicting the next 1-month return of the TSE300, the linear IOHMM with fundamental economic input variables is significantly better than all the other models.

For predicting the next 1-month return of the 14 sector indices,  
* when predicting all 14 sectors simultaneously, the non-linear IOHMM (with MLPs) is significantly better than all the other tested models (except the mixture of linear expert, against which the p-value is 6.3%).
* when predicting all 14 sectors simultaneously, one of the linear IOHMMs is significantly better than most other models except a non-linear IOHMM and a mixture of linear experts.

There are many more experiments and analyses that should be done in order to pursue in the direction explored here. Most interestingly, we would like to ascertain whether the improvements in likelihood brought by the various mixture models and in particular the IOHMMs can be used in order to improve financial decision-taking, e.g., for assessing risk in portfolio management or for trading options.

Acknowledgments

The authors would like to thank François Gingras, Éric Couture, as well as the NSERC Canadian funding agency.

References


18


