Hidden Markov Models Applied To Intraday Momentum Trading With Side Information

Haochen Jiang

Stony Brook University

December 12, 2022

Outline

Introduction of Paper

Contents

Introduction of Paper

Title

Hidden Markov Models Applied To Intraday Momentum Trading With Side Information[1]

Authors and affiliations

Hugh L Christensen*

Signal Processing and Communications Laboratory, Engineering Department, Cambridge University, CB2 1PZ, UK

Richard E Turner

Computational and Biological Learning Laboratory, Engineering Department, Cambridge University, CB2 1PZ, UK

Simon J Godsill

Signal Processing and Communications Laboratory, Engineering Department, Cambridge University, CB2 1PZ, UK

Introduction of Paper

No code about this paper are found

Link to this paper: https://arxiv.org/abs/2006.08307?context=q-fin

Introduction of Paper - Abstract

A Hidden Markov Model for intraday momentum trading is presented which specifies a latent momentum state responsible for generating the observed securities' noisy returns.

Existing momentum trading models suffer from **time-lagging** caused by the delayed frequency response of digital filters. Time-lagging results in a momentum signal of the wrong sign, when the market changes trend direction. A key feature of this state space formulation, is no such lagging occurs, allowing for accurate shifts in signal sign at market change points.

The number of latent states in the model is estimated using three techniques, cross validation, **penalized likelihood criteria** and simulation based model selection for the marginal likelihood. All three techniques suggest either 2 or 3 hidden states. Model parameters are then found using **Baum-Welch** and **Markov Chain Monte Carlo**, whilst assuming a single (discretized) univariate Gaussian distribution for the emission matrix.

Introduction of Paper - Abstract

Often a momentum trader will want to condition their trading signals on additional information. To reflect this, learning is also carried out in the presence of **side information**. Two sets of side information are considered, namely a ratio of realized volatilities and intraday seasonality. It is shown that splines can be used to capture statistically significant relationships from this information, allowing returns to be predicted.

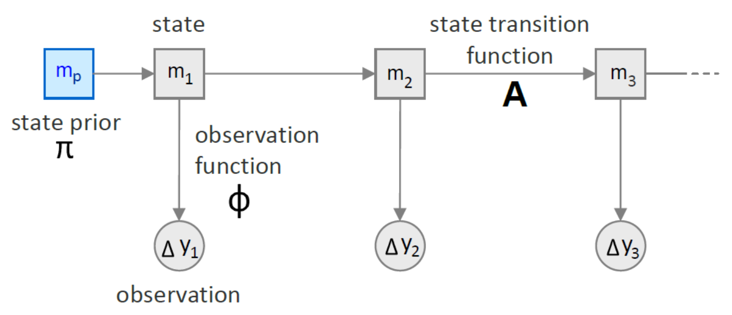
An **Input Output Hidden Markov Model** is used to incorporate these univariate predictive signals into the transition matrix, presenting a possible solution for dealing with the signal combination problem.

Bayesian inference is then carried out to predict the securities $t + 1$ return using the forward algorithm. The model is simulated on one year's worth of e-mini SP500 futures data at one minute sampling frequency, and it is shown that pre-cost the models have a Sharpe ratio in excess of 2.0.

Simple modifications to the current framework allow for a fully non-parametric model with asynchronous prediction.

Introduction of Paper - Main Results

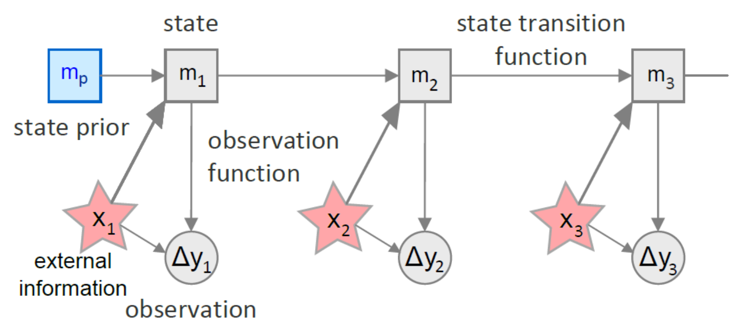
Structure of HMM



*A state space model for a discretized continuous observed state Δy (the change in price) and a discrete hidden state m (the trend). The relationship between the latent variables and the system parameters is shown.

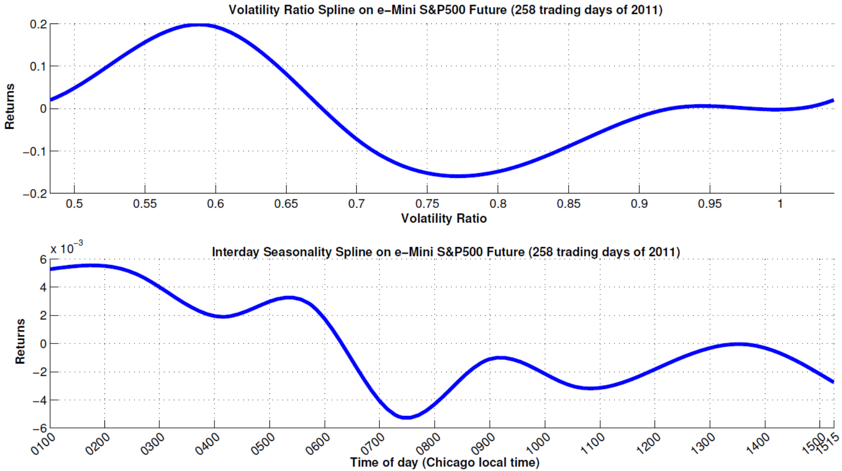
Introduction of Paper - Main Results

Structure of IOHMM



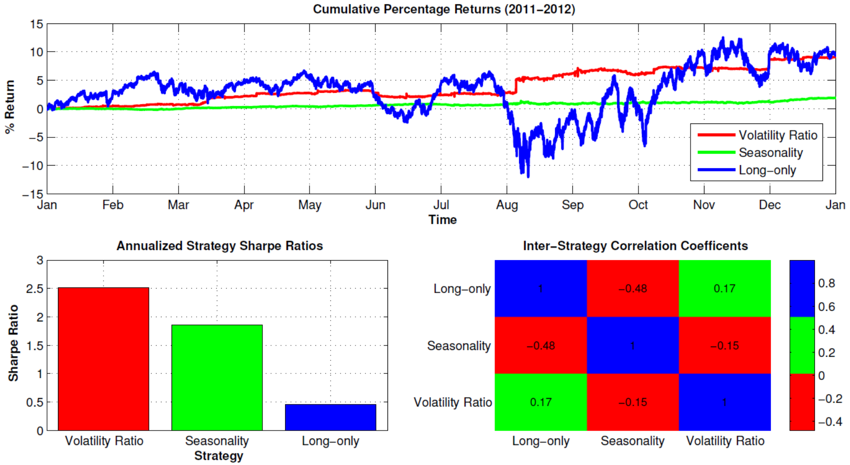
*Bayesian network showing the conditional independence assumption of a synchronous IOHMM. ΔY is an observable discrete output, X is an observable discrete input and M is an unobservable discrete variable. The model at time t is described by the latent state conditional on the observed state and some external information $p(m_t|\Delta y_{1:t}, x_t)$.

Introduction of Paper - Main Results



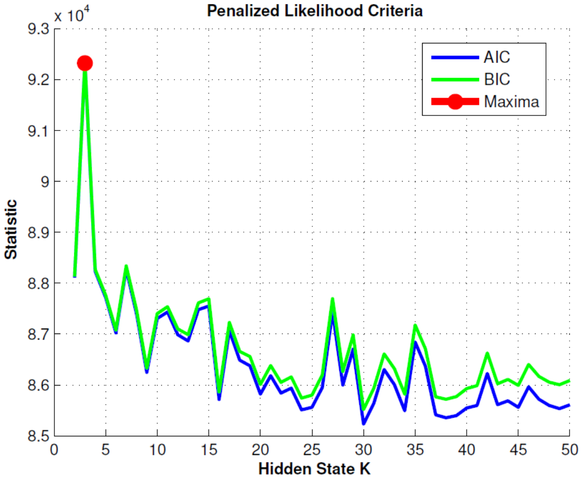
*Forecasting splines. Subplot one shows the spline generated by the volatility ratio predictor. Subplot two shows the spline generated by the seasonality predictor. This approach could be generalized when using $\mathcal{N}$ predictors, by generating an $\mathcal{N}$-dimensional spline.

Introduction of Paper - Main Results



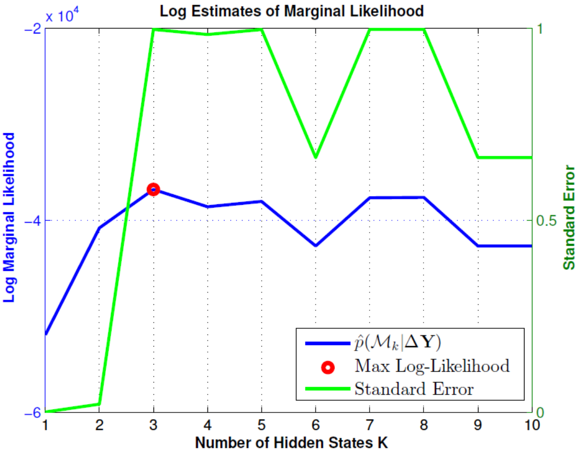
*Forecasting splines results. Subplot one shows the annual returns for the two strategies against a long-only portfolio for the 258 trading days of 2011. Subplot two shows the mean (pre-cost) annualized Sharpe ratios for the strategies. Subplot three shows the correlation coefficients between the strategies.

Introduction of Paper - Main Results



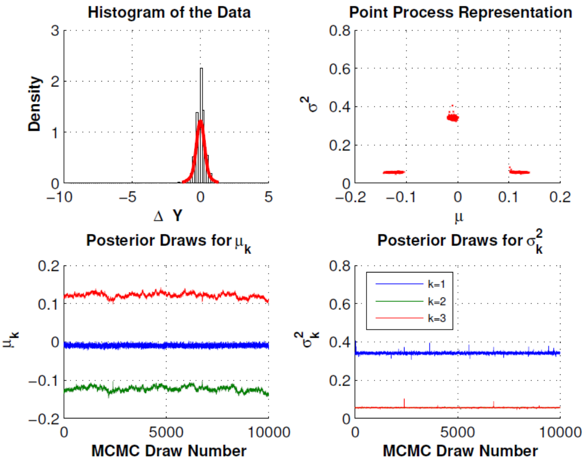
*Penalized likelihood criteria. Finding the number of hidden states using Baum-Welch. The optimal model of $K = 3$ is shown by a red dot.

Introduction of Paper - Main Results



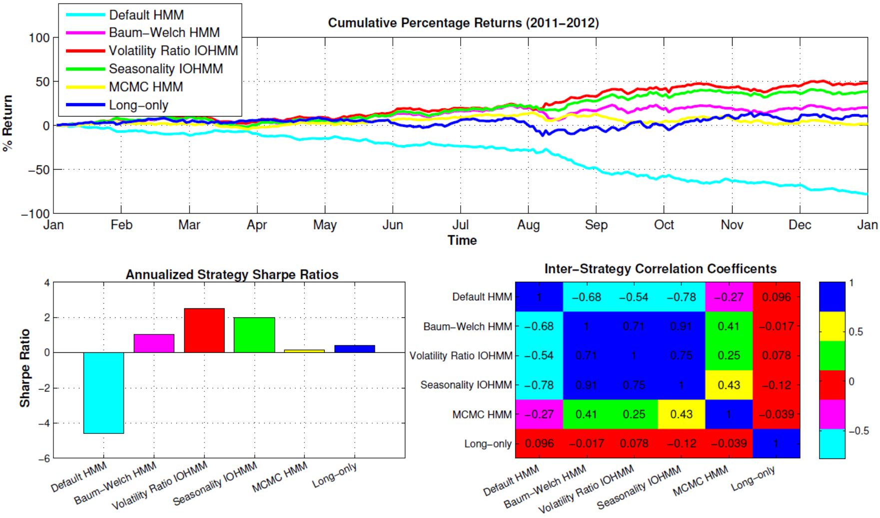
*Log of the bridge sampling estimator of the marginal likelihood $\hat{p}(\Delta Y|\mathcal{M}_k)$ under the default prior for $K = 1, \cdots, 10$. The maximum is at $K = 3$. On the right-hand axis the standard error is shown for each model.

Introduction of Paper - Main Results



*Markov Chain Monte Carlo by the Metropolis-Hastings algorithm. Subplot one, histogram of the data in comparison to the fitted 3 component Gaussian mixture distribution. Subplot two, a point process representation for $K = 3$. Subplot three, MCMC posterior draws for μ_k. Subplot four, MCMC posterior draws for σ_k^2.

Introduction of Paper - Main Results



*Simulation results for the five variations of the HMM intraday momentum trading strategy, with $K = 2$ or 3, plus the long-only case.

Introduction of Paper - Data and Simulation

Data from the CME GLOBEX e-mini SP500 (ES) future is used, one of the most liquid securities' in the world. Tick data is used for the period 01/01/2011 to 31/12/2011, giving 258 days data. The synchronous form of the algorithm is implemented and the tick data pre-processed by aggregating to periodic spacing on a one minute grid, giving 856 observations per day. Only 0100-1515 Chicago time is considered, Monday-Friday, corresponding to the **most liquid trading hours**. 1515 Chicago time is when the GLOBEX server closes down for its maintenance break and when the exchange officially defines the end of the trading day. Only use the front month contract is used, with contract rolling carried out 12 days before expiry. Only GLOBEX (electronic) trades are considered, with pit (human) trades being excluded. No additional cleaning beyond what the data provider has done is carried out.

CME GLOBEX e-mini SP500 (ES) future Trading time

Introduction of Paper - Conclusions

This paper has presented a viable framework for intraday trading of the momentum effect using both Bayesian sampling and maximum likelihood for parameters, and Bayesian inference for state. The framework is intended to be a practical proposition to augment momentum trading systems based on low-pass filters, which have been use since the 1970's. A key advantage of our state space formulation is that **it does not suffer from the delayed frequency response that digital filters do**. It is this time lag which is the biggest cause of predictive failure in digital filter based momentum systems, due their poor ability to detect reversals in trend at market change points.

Often a trend-following system will want to incorporate external information, in addition to the momentum signal, leading to the **signal combination problem**. An IOHMM is formulated as possible solution to this problem. In an IOHMM, the transition distribution is conditioned not only on the current state, but also on an observed external signal. Two such external signals are generated, seasonality and volatility ratio, both with positive Sharpe ratios, and are incorporated into the IOHMM.

Introduction of Paper - Conclusions

The performance of the IOHMM can be seen to be improved over the HMM, suggesting the IOHMM methodology used is a possible solution to the signal combination problem.

In addition to presenting novel applications of HMMs, this paper provides additional support for the **momentum effect** being profitable, pre- and post-cost, and adds to the substantial body of evidence on the effect. While much of the existing literature shows that the momentum effect is strongest at the 1-3 month period, we have shown the effect is viable at higher trading frequencies too.

Finally it is noted that this work is an instance of **unsupervised learning** under a single basic generative model. As such it can be linked to other work in the field by noting when the state variables presented in this model become continuous and Gaussian, the problem can be solved by a Kalman filter and when continuous and non-Gaussian the problem can be solved by a particle filter, for example (Christensen et al., 2012).

Introduction of Paper - Future Work

Asynchronous Price Data

The popularity of such synchronous methodologies in dealing with financial data arises from the computational challenge of dealing with the huge amounts of data generated by the markets. In reality, financial data is asynchronous due to trades clustering together (Dufour and Engle, 2000).

Aggregation is the process of moving from asynchronous to synchronous data and this acts as a zero-one filter. Such a rough down-sampling procedure means potentially useful high-frequency information is thrown away. The Bayesian approach to this problem is to keep as much information as possible and then let the model decide how what parts are/are not needed.

Gaussian Assumption of Emission Distributions

The Gaussian assumption could be replaced with any other parametric distribution (for example, fat-tailed Cauchy) or a non-parametric approach (for example, kernel density estimation).

Hidden Markov Model Basics

Contents

Main Reference

Statistical Learning Methods

Hang Li (2019) [2]

A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition

Lawrence R. Rabiner (1989) [3]

Hidden Markov Models for Time Series: An Introduction using R

Zucchini, Walter and MacDonald, Iain L (2009) [5]

Hidden Markov Model Basics

EM Algorithm [2]

EM Algorithm

Introduction

In statistics, an expectation–maximization (EM) algorithm is an iterative method to find (local) maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

EM Algorithm

Introduction

As a convention, we use Y to represent our observations, while Z is the latent or missing data, so that (Y;Z) is called complete-data. And observations are also called incomplete-data.

Assume we are given the observations Y with probabilistic distribution $P(Y|\theta)$, where θ is the model parameters to be estimated. Then the likelihood function of incomplete data Y is $P(Y|\theta)$, whose log-likelihood function $LL(\theta) = \log P(Y|\theta)$; And assume the joint distribution of (Y,Z) is $P(Y, Z|\theta)$, then the complete data log-likelihood function is $CDLL(\theta) = \log P(Y, Z|\theta)$.

EM algorithm uses iterations to estimate the maximum likelihood estimation of $LL(\theta) = logP(Y|\theta)$.

EM Algorithm

Algorithm (EM Algorithm)

Input: observations Y, latent data Z, joint distribution $P(Y, Z|\theta)$, *conditional distribution* $P(Z|Y, \theta)$;
Output: model parameters θ.

1. Start with initial guesses for the parameters $\theta^{(0)}$;

2. Expectation Step: In (i+1)th step, compute:

$$\begin{aligned} Q\left(\theta, \theta^{(i)}\right) &= E_Z\left[\log P(Y, Z|\theta)|Y, \theta^{(i)}\right] \\ &= \sum_Z \log P(Y, Z|\theta) \cdot P(Z|Y, \theta^{(i)}) \end{aligned}$$

where $\theta^{(i)}$ *is the estimate of the ith step, and* $P(Z|Y, \theta^{(i)})$ *is the conditional distribution of latent data Z given observations Y and current estimate of parameters* $\theta^{(i)}$.

EM Algorithm

Algorithm (EM Algorithm)

3. Maximization Step: determine the maximizer of $Q\left(\theta, \theta^{(i)}\right)$ *over* θ *as the estimate of (i+1)th step of model parameters* $\theta^{(i+1)}$*.*

$$\theta^{(i+1)} = \arg\max_{\theta} Q\left(\theta, \theta^{(i)}\right)$$

4. Iterate steps 2 and 3 until convergence

Function $Q\left(\theta, \theta^{(i)}\right)$ *is the core concept in the EM algorithm, which is called Q function.*

EM Algorithm

Definition (Q Function)

The expectation of complete-data log-likelihood $CCDL(\theta) = \log P(Y, Z|\theta)$ given the conditional probabilistic distribution $P(Z|Y, \theta^{(i)})$ of unobserved data Z conditioning on the observations Y and current estimate of parameters $\theta^{(i)}$ is called **Q Function**, i.e.,

$$Q\left(\theta, \theta^{(i)}\right) = E_Z\left[\log P(Y, Z|\theta)|Y, \theta^{(i)}\right]$$

EM Algorithm

Remarks

Here are some remarks for EM algorithm:

1. The initial values can be randomly chose, but we have to notice that EM algorithm is sensitive to initial values, which means that it is likely to reach the local maxima finally.

2. In the maximization in M-step, when $\theta^{(i+1)}$ is estimated, we finish one step. It will be illustrated that log-likelihood function will increase or reach local maxima after each step.

3. As for the condition that we stop iterating, we commonly use

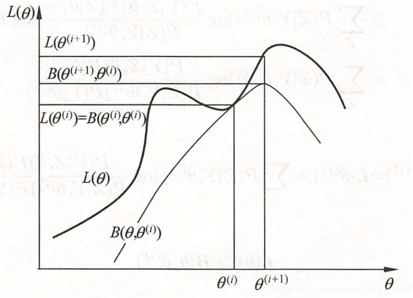
$$||\theta^{(i+1)} - \theta^{(i)}|| < \epsilon_1 \quad or \quad ||Q\left(\theta^{(i+1)}, \theta^{(i)}\right) - Q\left(\theta^{(i)}, \theta^{(i)}\right)|| < \epsilon_2$$

   where ϵ_1 and ϵ_2 are very small positive numbers. When the inequality is satisfied, we will stop iterating.

EM Algorithm

Remarks

It is obvious that this equation is equivalent to one step in EM algorithm. We can conclude that the essence of EM algorithm is to take steps to maximize its lower bound to approximate the maxima of itself. The figure below illustrates this idea. And we can also know that EM algorithm is not guaranteed to find the global maxima. Thus, the choice of initial values is very important.



*Geometric Explanation of EM Algorithm.

EM Algorithm

Finally, here are two theorems about the monotonicity and convergence of EM algorithm.

Theorem (Monotonicity)

If $P(Z|\theta)$ is the likelihood function of observations, $\{\theta^{(i)}\}(i = 1, 2, ...)$ is the parameter sequence estimated from EM algorithm, and $\{P(Z|\theta^{(i)})\}(i = 1, 2, ...)$ is the corresponding likelihood function sequence. Then $P(Z|\theta^{(i)})$ is monotonically increasing, i.e.,

$$P(Z|\theta^{(i+1)}) > P(Z|\theta^{(i)})$$

Proof.

See Hang Li (2019) [2]. ∎

EM Algorithm

Theorem (Convergence)

If $LL(\theta) = \log P(Y|\theta)$ is the likelihood function of observations, $\{\theta^{(i)}\}(i = 1, 2, ...)$ is the parameter sequence estimated from EM algorithm, and $\{LL(\theta^{(i)})\}(i = 1, 2, ...)$ is the corresponding log-likelihood function sequence.

(1) *If $P(Z|\theta)$ has a upper bound, then $LL(\theta) = \log P(Y|\theta)$ will converge to a value LL^*;*

(2) *When the functions $Q\left(\theta, \theta^{(i)}\right)$ and $LL(\theta)$ are under certain circumstances, convergence value θ^* of parameter sequence estimated from EM algorithm $\{\theta^{(i)}\}(i = 1, 2, ...)$ is a stationary point of $LL(\theta)$.*

Proof.

See Wu (1983) [4]. ■

Side Information

Contents

Side Information - Introduction

In “classical” trading models, the $t + 1$ return of a security is forecast by a “signal” which is a univariate time series, typically synchronous and continuous between -1 and 1. When this signal is > 0 the trader will go “long” and when the signal is at < 0 the trader will go “short”.

In the simple case of a portfolio consisting of only one security, the number of lots of security to be traded is directly proportional to the product of the **signal magnitude** and **available capital**. Historically such predictive trading signals are generated from either “technicals” or “fundamentals”. Technicals are signals based on the prior behaviour of the security (Schwager, 1995b). Fundamentals are signals based on upon extrinsic factors (such as economic data) (Schwager, 1995a).

Side Information - Introduction

Predictive signals are generated and shown to have statistical traction against security returns. In Input-Output Hidden Markov Model, the information held in signals is used when learning transition matrix A. This methodology is quite general and as such could be applied to any technical or fundamental predictor.

Momentum traders often want to combine their momentum signal with one or more extrinsic predictive signals to give a single forecast. This is called the **signal combination problem** for which a variety of different solutions exist.

It is noted that our approach of biasing the transition dynamics of an HMM momentum trading system using external predictors seems to be another possible solution to this problem.

Special care is taken to ensure that the simulation is a truly out-of-sample simulation. Specifically, each one of the data points used to evaluate a trade had not been used in any of the previous stages of model identification, learning and estimation.

Side Information - Predictor I: Volatility Ratio

An extensive body of empirical research exists showing that realized volatility has predictive power against security returns.

We choose to implement the ratio of current volatility to historical volatility termed the **volatility ratio** as designed by Chande in 1992 (Chande, 1992) which requires estimating conditional volatilities for “now” and in the “past” (Colby, 2002; investopedia.com, 2016; quantshare.com, 2016).

We parameter sweep the ratio and select values $\psi_{fast} = 50$ and $\psi_{slow} = 100$ based on stability and predictive performance. The input X to Spline Forecasting Algorithm is given by

$$X_i = \left\{ \frac{\sigma_{i,t+1|t}(\psi_{fast})}{\sigma_{i,t+1|t}(\psi_{slow})} \right\}_{t=1}^{T}$$

where ψ represents the window size of EWMA (Exponential Weighted Moving Average) methodology, known as the J.P. Morgan Risk Metrics EWMA model (JPM, 1996; Pafka and Kondor, 2001).

Side Information - Predictor I: Volatility Ratio

EWMA (Exponential Weighted Moving Average) methodology

The EWMA methodology exponentially weights the observations, representing the finite memory of the market, as per following equation

$$\sigma_{i,t+1|t} = \sqrt{(1-\lambda)\sum_{\tau=0}^{\psi} \lambda^{\tau} R^2_{i,t-\tau}}$$

where $R_{i,t}$ represents the log return during time $t-1$ and t at day i. The model has two parameters (window size) and λ (variance decay factor where $0 < \lambda < 1$) which are fixed a-priori with a trade-off between λ and ψ, with a small λ yielding similar results to a small ψ.

The original J.P. Morgan documentation suggests using $\lambda = 0.94$ with daily frequency data, though we increase the reactivity of the term to fit our one minute frequency data and set $\lambda = 0.79$ (Pesaran and Pesaran, 2007; Patton, 2010). However, we set $\lambda = 0.94$ since it decays too fast if $\lambda = 0.79$.

Side Information - Predictor II: Seasonality

Intra-daily seasonality is where returns vary conditionally on the location within the trading day. A wide range of methodologies for extracting seasonality signals from financial data exist in the literature.

As we know the size of the cycle a-priori, believe the effect to be non-linear and prefer to work in the time domain, splines are chosen to estimate the relationship between **time of day** and **security return**. The use of splines seems to be a well accepted way of capturing seasonality.

Side Information - Predictor II: Seasonality

Following the approach of Martin et al a seasonal index is used to quantify the cycle (Martín Rodríguez and Cáceres Hernández, 2010). Here the author constructs an index by defining the period of time under consideration and then partitioning it into a periodic grid between 1 and T and then assigning observations to buckets on this grid. The authors then capture the seasonal variation by fitting a spline to the seasonal index and bucketed-data.

In the case of our one-minute frequency data the size of the period is $T = 390$. The input X to Spline Forecasting Algorithm is given by $X_i = [1, \cdots, T]$.

Side Information - Forecasting with Splines

Splines are now introduced as the methodology by which we condition learning of the transition matrix. Splines are a way of estimating a noisy relationship between dependent and independent variables, while allowing for subsequent interpolation and evaluation (Reinsch, 1967).

We use the **B-spline** as a way of capturing a stationary, non-linear relationship between predictor and security return.

Furthermore, **Polynomial Regression Spline** (short for **P-spline** later) will be used with the same parameters except for the degree of polynomials to get the benchmark, so that we can make a comparison.

Side Information - Forecasting with Splines

Remark

Moreover, the choice of the number of knots for B-spline is important. Too many knots means the spline will be very tightly fitted to the data, while too few knots may fail to capture the relationship of interest. The problem with over-fitting the relationship being that the in-sample performance will be great, but the out-of-sample performance will be poor. Hence it is a matter of balance which is decided upon by intuition about the variability of the underlying economic relationship. 6 knots are chosen for the volatility predictor and 10 knots for the seasonality predictor.

Side Information - Forecasting with Splines

For a predictor $X = \{X_1, \cdots, X_{N-1}\}$, where $X_i = \{X_{i,1}, \cdots, X_{i,T}\}$ and $i \in \mathbb{N}^+ \cap [1, N-1]$. The learning and subsequent forecasting procedure is shown in the next slide.

In that algorithm, $t = 1, \cdots, T$ is intra-day time and $n = 1, \cdots, n^*, \cdots, N$ is inter-day time. Spline evaluation is intra-day while spline learning is inter-day where the spline is "grown" over time, allowing it to capture new information and forget old information.

In our code $n^* = 66$ days with $N = 253$ days (the first day is set to be the warm-up data in volatility ratio case, thus $N = 252$ days) and $T = 390$ observations per day. In this way the spline is estimated using the previous 66 trading days worth of data, on a rolling basis.

Side Information - Learning and Forecasting With Splines

Algorithm (Learning and Forecasting With Splines)

Input Log returns $R = \{R_1, \cdots, R_N\}$, where $R_j = \{R_{j,1}, \cdots, R_{j,T}\}$ and $j \in \mathbb{N}^+ \cap [1, N]$;

Output Trading signals $S = \{S_{n^*+1}, \cdots, S_N\}$;
Spline forecasts of log returns $\widehat{R} = \left\{\widehat{R_{n^*+1}}, \cdots, \widehat{R_N}\right\}$.

1. Generate the mean μ_k and standard deviation σ_k of the current batch $B_k = \{R_k, \cdots, R_{k-1+n^*}\}$, where $k \in \mathbb{N}^+ \cap [1, N - n^*]$;
2. Normalize the log returns of the current batch, $\overline{B_k} = \dfrac{B_k - \mu_k}{\sigma_k}$;
3. Generate the inputs $X_{B_k} = \{X_k, \cdots, X_{k-1+n^*}\}$ and X_{k+n^*};
   - Volatility Ratio case

$$X_k^v = \left\{ \sqrt{(1-\lambda) \sum_{\tau=0}^{\psi_{fast}} \lambda^\tau R_{k,t-\tau}^2} \Bigg/ \sqrt{(1-\lambda) \sum_{\tau=0}^{\psi_{slow}} \lambda^\tau R_{k,t-\tau}^2} \right\}_{t=1}^{T}$$

Side Information - Learning and Forecasting With Splines

Algorithm (Learning and Forecasting With Splines) (Cont.)

3
- Seasonality case
  $X_k^s = [1, \cdots, T]$
- Combined case
  $X_k^c = (X_k^v, X_k^s)$

4 Generate the B-spline G_k of the current batch;

- Volatility Ratio case
  $G_k^v = spline(X_{B_k}^v, \overline{B_k}, knots = 6)$
- Seasonality case
  $G_k^s = spline(X_{B_k}^s, \overline{B_k}, knots = 10)$
- Combined case
  $G_k^c = spline(X_{B_k}^c, \overline{B_k}, knots = 6 \times 10)$

5 Evaluate the splines and get

- $S_{k+n^*} = limit_{-1}^{1}(G_k\left(X_{k+n^*}\right))$
- $\widehat{R_{k+n^*}} = \sigma_k \cdot G_k\left(X_{k+n^*}\right) + \mu_k$

6 Reset k to $k+1$, and iterate steps 1 to 6 until k exceed $N - n^*$.

7 Gather the all of the results, finally.

Input Output Hidden Markov Models

Contents

Input Output Hidden Markov Models

The HMM is re-specified by incorporating the side information held in the splines, such that the **transition distribution** is given by $p\left(i_t | i_{t-1} = q_j, X_{\cdot,t}\right)$.

The belief behind this new model is that the extrinsic data is of value to predicting the change in price of the security. Essentially we are saying that not all of the securities' variance can be explained by the momentum effect, even though we believe it to be the dominant factor.

In Input Output Hidden Markov Models (IOHMMs) the observed distributions are referred to as inputs and the emission distributions as outputs (Bengio and Frasconi, 1995). Like regular HMMs, IOHMMs have a fixed number of hidden states, however the output and transition distributions are **not only** conditioned on the current state, but are also conditioned on an observed discrete input value X.

Input Output Hidden Markov Models

We consider the simplifying case where the input and output sequences are **synchronous** (Bengio et al., 1999). Such a system can be represented with discrete state space distributions for emission $p\left(R_{\cdot,t}|i_t, X_{\cdot,t}\right)$ and transition $p\left(i_t|i_{t-1}, X_{\cdot,t}\right)$. When the extrinsic predictor and the HMM momentum predictor have different time stamps, or are of different sampling frequencies, an asynchronous setup is required, adding computational complexity to the forward-backward recursion (Bengio and Bengio, 1996). It is noted such a technique could allow signals of a lower frequency to be used in a high-frequency inference problem, for example, low-frequency macro-economic data could be used to bias intraday trading.

The literature suggests three main approaches to learning in IOHMM: **Artificial neural networks** (Bengio and Frasconi, 1995), **partially observable Markov decision processes** (Bäuerle and Rieder, 2011) and **EM** (Bengio et al., 1999). As Baum-Welch was used for learning in the HMM case, in order to be consistent we opt to learn by EM for the IOHMM case too. In terms of the Algorithm, the only changes required to deal with the IOHMM case are the induction formulas for α's (forward variables) and β's (backward variables).

Input Output Hidden Markov Models

Recurrence Relation for α's and β's in HMM

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(R_{\cdot,t+1}),\ i = 1, \cdots, N,\ t = 1, \cdots, T-1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(R_{\cdot,t+1}) \beta_{t+1}(j),\ i = 1, \cdots, N,\ t = T-1, \cdots, 1$$

To implement this methodology a different A is trained for every unique value of $X_{\cdot,t}$. Such an approach has the drawbacks of over parameterization and requiring large amounts of data. This is solved by discretizing the spline according to its roots, with $\mathcal{L}-1$ roots giving $\mathcal{L}$ "buckets" of spline. $X_{\cdot,t}$ is then aligned with $R_{\cdot,t}$, and $R_{\cdot,t}$ assigned to one of the $\mathcal{L}$ buckets, the contents of each bucket being concatenated to give a data vector. Baum-Welch learning algorithm is then carried out on each of these vectors, as before.

Input Output Hidden Markov Models

Recurrence Relation for α's and β's in IOHMM

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}^{\mathcal{L}^* \leftarrow X_{\cdot,t}}\right] b_i^{\mathcal{L}^* \leftarrow X_{\cdot,t}}(R_{\cdot,t+1}),$$

$$i = 1, \cdots, N,\ t = 1, \cdots, T-1$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}^{\mathcal{L}^* \leftarrow X_{\cdot,t}} b_j^{\mathcal{L}^* \leftarrow X_{\cdot,t}}(R_{\cdot,t+1}) \beta_{t+1}(j),$$

$$i = 1, \cdots, N,\ t = T-1, \cdots, 1$$

As the transition distribution $p\left(i_t | i_{t-1}, X_{\cdot,t}\right)$ is time sequential, concatenating the bucketed data is **strictly incorrect** as occasionally $p\left(i_t | i_{t-\tau}, X_{\cdot,t}\right)$ occurs, where $\tau > 1$.

Given the smoothness of the splines, concatenation is rare and so the resulting small loss of Markovian structure can be ignored. The obvious advantage of discretizing by roots is that parameters $\{A_1, \cdots, A_{\mathcal{L}}\}$ map to **signed** returns.

Input Output Hidden Markov Models

Remarks

Using the methodology described before, two independent predictions are generated for each of the two IOHMM models, one for the volatility ratio and one for seasonality. However, it maybe the case we wish to combine the two predictors into a single prediction. In this case of more than one predictor, X is treated as multivariate and a multi-dimensional spline is generated. Subject to some appropriate discretization of the spline, the same algorithm can then be applied to solve $p\left(i_t | i_{t-1}, \overrightarrow{X_{\cdot,t}}\right)$, where $\overrightarrow{X_{\cdot,t}}$ is a vector.

Furthermore, in my implementation, I will generate multi-dimensional splines by combine these two predictions, and finally apply them to IOHMM model to get signals.

Input Output Hidden Markov Models

Remarks

The major issue when learning is the transient nature of the latent state and how stable its estimated means are. In order to ensure the most accurate estimation possible, the means of the K Gaussian distributions (the trends) are efficiently estimated using short windows of data. This is implemented using a rolling window of data that consists of **23 trading days (one month)**. This window size approximately agrees with the lowest frequency information we are trying to exploit in our system.

Input Output Hidden Markov Models

Criticisms

If we only use side information $X_{\cdot,t}$ to divide the training data into different "buckets" by spline roots, there are two main problems. One is from the generated splines, we can see that the training data with different side information might yield exactly the same signals or return forecasts. The other one is this method does not take advantages of the signals generated in the step of spline forecasting, which are more valuable in my view.

Thus, I will train IOHMM with **spline signals**, which is different from the author

Splines are not always guaranteed to have roots from my observations, in this case, IOHMM will **degenerate** to HMM.

Input Output Hidden Markov Models

Criticisms

When there are very large differences among the **distribution parameters** of the buckets, the transition probability will become extremely inaccurate, then very rough estimation of return predictions will be given.

Moreover, when implementing IOHMM, we have to make sure that the distributions and states in different "buckets" are the same or at least similar to a certain extent. In other words, we have to guarantee that the states in different "buckets" refer to **similar status**. Only in this way will the models be meaningful.

In the author's implementation, maybe for simplicity, where a huge amount of **computation cost** will be consumed, or to avoid some issues like **overfitting**. Each set of parameters will be used to predict returns in every time point in one day, i.e, 390 time points.

Maybe the difference of the two algorithms can be figured out by numerical experiments.

Numerical Results

Contents

References

Contents

References I

[1] Christensen, H., Godsill, S., and Turner, R. E. (2020). Hidden markov models applied to intraday momentum trading with side information.

[2] Hang Li (2019). *Statistical Learning Methods*. Tsinghua University Press, 2 edition.

[3] Lawrence R. Rabiner (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286.

[4] Wu, C. J. (1983). On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103.

[5] Zucchini, Walter and MacDonald, Iain L (2009). *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC.

Thank you!