Trading Filter for Risk Management with Hidden Markov Models

Lavaneesh Sharma

ABSTRACT

Market Regimes are periods where modification of financial markets occurs due to market stress, regulations, government policy and other macroeconomic effects. Detection of such time periods is one of the major challenges in quantitative finance. Hidden Markov models, have offered a solution to this problem and we shall further explore the how such regimes can first be detected using Hidden Markov models, the mathematical framework and then used as a risk management technique for practical trading purposes.

INTRODUCTION

A regime is a period of changing financial markets due to extraneous factors. They may develop in times of stress, a government policy or other macroeconomic conditions. A detection of such changing periods becomes imperative because it leads to adjustment of asset returns via a shift in their metrics like mean, variance, correlations and variances. Particularly, they lead to varying correlation, excess kurtosis or fat tails, heteroscedasticity and skewed returns. As such regime detection particularly becomes useful for risk management (hedging), to change strategy or position sizing.

A statistical time series method known as Hidden Markov model is the chief method proposed to detect such regimes. The include interpretation of "hidden" generative process (underlying regime state) via a "noisy" indirect observations (asset returns). An overall audit will kick us off in the paper alongside a short portrayal of its numerical structure and its utilization as an exchanging channel for algorithmic trading. Then focus will be specific to architecture of HMM. Lastly, we will try to implement a HMM to S&P 500 series using a simple moving average strategy and compare its performance with and without regime detection.

MARKOV MODELS

In probability theory, a Markov model is a stochastic model used to model randomly changing systems. A jump occurs from one state to the other. The probability of such an event is dependent on current state only. It is "memoryless" and possesses Markov Property. Markov Models are classified as follows:

	Fully Observable	**Partially Observable**
Autonomous	Markov Chain	Hidden Markov Model
Controlled	Markov Decision Process	Partially Observable Markov Decision Process

If the data is somewhat visible ("hidden") but is completely autonomous we get a Hidden Markov Model. The probabilities of jumps and underlying latent states cannot be viewed directly. The observations (not necessarily have Markov property) are influenced by the latent states (possess Markov property).

MATHEMATICAL FRAMEWORK OF MARKOV MODEL:

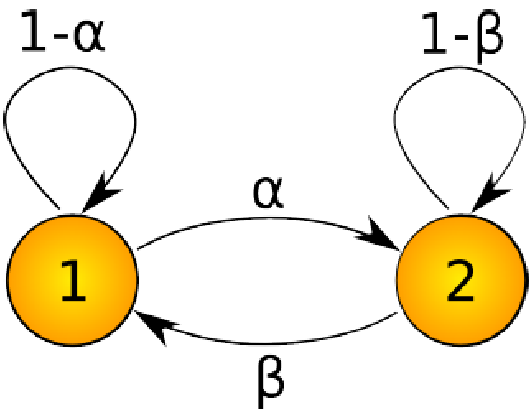
A time series consists of discrete observations $X_1, X_2, \dots, X_T$. For Markov Models, given any time instant t, all the characteristics to predict the future are possessed by the input X_t. The probability of seeing observations can then be written as:

$$
\begin{aligned}
p(X_{1:T}) &= p(X_1)p(X_2 \mid X_1)p(X_3 \mid X_2) \\
&= p(X_1)\prod_{t=2}^{T} p(X_t \mid X_{t-1})
\end{aligned}
$$

The market regimes models for financial markets will be considered to be in Discrete Time. Probability of jump from states per unit time is enclosed in a $K\ x\ K$ matrix given as:

$$A_{ij} = p(X_t = j \mid X_{t-1} = i)$$

For a two-state Markov Model for example:



The probabilities of making a jump from state 1 to state 2 is α while that of not making a jump is $1 - \alpha$. The cumulative sum of $\alpha + 1 - \alpha = 1$. The tranisition matrix of this system is given as:

$$A = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$

Generally speaking, for a n-steps of a Discrete-State Markov Chain model the transition matrix $A(n)$ is given as :

$$A_{ij}(n) := p(X_{t+n} = j \mid X_t = i)$$

HIDDEN MARKOV MODELS:

The states in HMMs are hidden from each other and can't be observed. In real markets, concealed stage is the change in market scenario while the changes in price of assets are outputs and unmasked. More specifically for a HMM we must:

1. Make a joint density function of observations
2. Define a Time Invariant transition matrix
3. Create a series of distinct states $z_t \in \{1,2,\dots,K\}$
4. Have a conditional probability to see a particular asset's return given the market scenario.

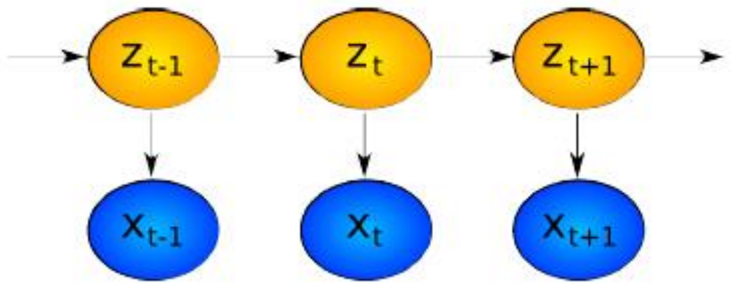
The probabilistic model interchanges between states It is supposed to stay in a "jumped" state for a while. This exactly the behavior expected from a model applied to financial market regimes.

$$
\begin{aligned}
p(\mathbf{z}_{1:T} \mid \mathbf{x}_{1:T}) &= p(\mathbf{z}_{1:T}) p(\mathbf{x}_{1:T} \mid \mathbf{z}_{1:T}) \\
&= \left[p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1}) \right] \left[\prod_{t=1}^{T} p(\mathbf{x}_t \mid z_t) \right]
\end{aligned}
$$

The primary equation expresses that joint likelihood of seeing the full arrangement of concealed states and perceptions is equivalent to the likelihood of seeing the shrouded states duplicated by the likelihood of seeing the perceptions, contingent on the states. This bodes well as the perceptions can't influence the states, yet the concealed states do by implication influence the perceptions.

Two transition functions are obtained in second line; one each for the states and observations respectively.

Here for continuous time asset returns a conditional Multivariate Gaussian distribution is chosen with mean μ_k and covariance σ_k. Below is evolution of states z(t) and their evolution to observations x(t):



REGIME DETECTION WITH HIDDEN MARKOV MODELS

In our paper HMM based filter will be analyzed for equities traded in US, particularly S&P500. It will be implemented as a risk managing regime filter. It will prohibit trades to occur when the standard deviation of returns is high. The aim is that unprofitable trades will be removed and thereby volatility will be removed to ultimately increase the Sharpe Ratio. The trading strategy used is also quite simple. It is as follows.

TRADING STRATEGY:

We will use a relatively simple strategy for this paper, because we need to focus the its use for risk management perspectives. We use classic moving average strategy with rules:

- We compute 10-day and 30-day moving average
- If the 10-day average exceeds the 30-day average and security isn't bought, then buy
- Else if the 30-day average exceeds the 10-day average and security is bought, then short.

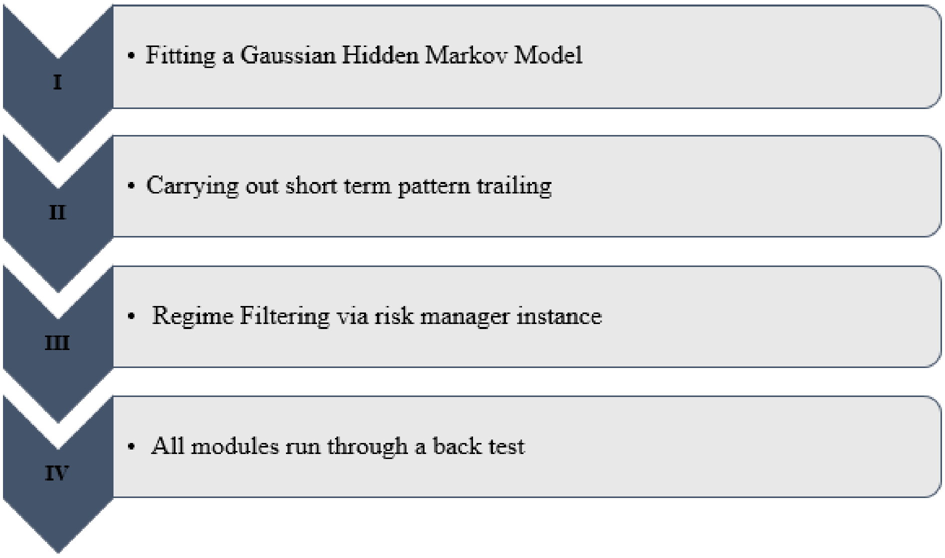
The above strategy won't perform much better than simple buy and hold, but combining it with a "blocking filter" we hope it becomes much more effective especially in periods of high standard deviation of returns.

We prepare the model on information of S&P500 from the 29/01/93 to 31/01/2004. The filter first checks, for each exchange sent, regardless of whether the present status is a low or high variance or standard deviation of returns. The following table summarizes the return strategy:

Volatility	Action
Low	Buying Permitted, Trade Executed
High	Close Out open positions, Block New Trades

This has the ideal impact of taking out pattern following exchanges times of high change where they are probably going to lose cash because of wrong ID of "pattern". The back trial of this technique is completed from 01/01/2005 to 31/12/2014, dropping the Hidden Markov Model, which implies HMM used for hold-out set.

There are four components of the sub-parts that we are about to implement. They go as:



1. Training the model. The HMM is fit to set of returns data. A python based *hmmlearn* library is used, along with other necessary imports. The data downloaded from S&P500 is cleaned to be fed into the model. The missing values are dropped. A plot of hidden states is used as a sanity check to see if HMM is producing "sensible" states. The output of it is given as:

The regime detection does capture highly volatile periods especially during 2008-2009 (financial crisis), where a Hidden State #1 nicely shows that. The *GaussianHMM* object requires an input of *n_components* which basically is number of states we wish to model. We chose two. Model is fit and output showing the hidden close prices is drawn. Model is saved to a path to be used by Risk Manager.

2. In the next step we actually implement the MA trading strategy discussed, that will be filtered by $RiskManager$ module. An instance of subclass is defined that essentially implements our trading strategy. The QSTrader has inbuilt functionality to generate signals. The first check in our case will be to check the price threshold (via OHLCV). A prices are added on a moving forward basis and if enough prices are obtained to do so, they are calculated. As a result, the Trading Strategy discussed above is implemented.
3. The $AbstractRiskManager$ is the actual object that will block trades during periods of high volatility. The only input it requires is for the HMM model file and also keeps a flag variable that tracks if a strategy is invested or not. We get a full rundown of changed shutting returns and a yield of anticipated system states. A conditional black checks for identification of a regime. If the volatility is low (State #0) it checks if order is "BOT" (long) or "SLD" (close). If it's the former, then keeps a tab if the long position is actually available. Else, it simply cancels the order. It also checks for State #1. No buying is allowed in this stage. Only closing of an order is allowed. Lastly a back test is conducted.

RESULTS

The strategy is used in conjunction with actual transaction costs provided from interactive brokers. It can be closely used as a real time, live trading strategy. The no filter strategy obtained is as follows:

(MIJ) 2020, Vol. No. 6, Jan-Dec **e-ISSN: 2454-924X; p-ISSN: 2454-8103**

Monthly Returns (%)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2005	0.0	-1.2	-5.1	0.0	7.5	-0.6	3.5	-3.0	-5.3	0.0	6.6	-1.8
2006	6.0	-5.8	3.2	4.6	-9.8	0.0	-6.7	6.2	6.1	8.3	5.0	1.9
2007	3.6	-4.6	-3.6	11.2	8.0	-3.0	-7.1	0.0	5.7	-1.2	0.0	-9.2
2008	-8.7	-8.6	-5.7	3.6	4.4	-3.0	0.0	0.9	-9.1	0.0	0.0	-3.8
2009	-23.0	0.0	-8.3	26.6	13.6	-1.1	7.7	8.0	6.4	-3.9	2.8	2.9
2010	-3.0	0.1	12.1	3.1	-9.8	-11.8	2.8	-4.9	3.0	9.1	0.6	6.1
2011	5.3	7.6	-3.7	5.4	-7.0	0.0	-7.5	-5.7	0.2	5.6	-11.5	-1.1
2012	12.1	10.6	6.4	-5.9	-2.3	0.7	2.7	5.7	4.4	2.8	0.0	1.4
2013	10.5	2.5	6.4	3.6	4.4	-1.9	4.0	-4.3	-2.6	5.6	5.7	3.8
2014	-7.4	2.4	0.8	-2.8	4.7	3.1	-2.6	2.1	-3.7	2.4	5.7	-10.1

Curve vs. Benchmark

Total Return	86%	73%
CAGR	6.41%	5.62%
Sharpe Ratio	0.37	0.37
Sortino Ratio	0.38	0.46
Annual Volatility	26.01%	20.37%
R-Squared	0.51	0.30
Max Daily Drawdown	57.70%	56.47%
Max Drawdown Duration	1299	1365
Trades per Year	4.1	

Trade

Trade Winning %	46%
Average Trade %	0.91%
Average Win %	5.24%
Average Loss %	-2.83%
Best Trade %	14.35%
Worst Trade %	-12.31%
Worst Trade Date	TBD
Avg Days in Trade	0.0
Trades	41

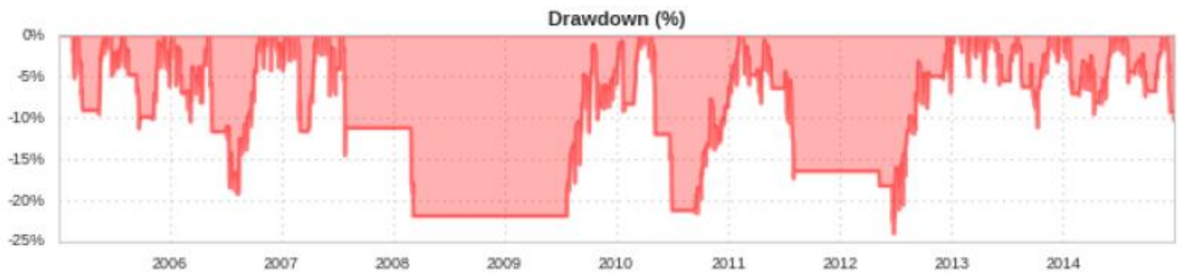
Time

Winning Months %	62%
Average Winning Month %	4.53%
Average Losing Month %	-5.62%
Best Month %	26.56%
Worst Month %	-22.97%
Winning Years %	50%
Best Year %	44.18%
Worst Year %	-27.09%

A Sharpe Ratio of 0.37 is obtained, implying it incorporates a lot of volatility to generate these returns. The CAGR show a 0.79% increase.

The same strategy when implemented with HHM regime detection filter is:

Trend Following Regime Detection with HMM



Monthly Returns (%)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2005	0.0	-1.2	-5.1	0.0	7.5	-0.6	3.5	-3.0	-5.3	0.0	6.6	-1.8
2006	6.0	-5.8	3.2	4.6	-9.8	0.0	-6.7	6.2	6.1	8.3	5.0	1.9
2007	3.6	-4.6	-3.6	11.2	8.0	-3.0	-7.1	0.0	0.0	0.0	0.0	0.0
2008	0.0	-7.5	-4.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2009	0.0	0.0	0.0	0.0	0.0	0.0	6.1	6.4	5.2	-3.2	2.3	2.4
2010	-2.5	0.1	9.9	2.6	-8.3	-9.8	-0.8	0.0	2.4	7.3	0.5	4.9
2011	4.3	6.3	-3.1	4.5	-5.9	0.0	-6.3	-4.7	0.0	0.0	0.0	0.0
2012	0.0	0.0	0.0	0.0	-2.1	0.6	2.6	5.4	4.1	2.6	0.0	1.3
2013	9.9	2.4	6.1	3.4	4.2	-1.8	3.8	-4.1	-2.5	5.3	5.5	3.7
2014	-7.1	2.3	0.7	-2.7	4.5	3.0	-2.5	2.0	-3.6	2.3	5.4	-9.7

Curve vs. Benchmark

Total Return	94%	73%
CAGR	6.88%	5.62%
Sharpe Ratio	0.48	0.37
Sortino Ratio	0.45	0.46
Annual Volatility	16.82%	20.37%
R-Squared	0.74	0.30
Max Daily Drawdown	23.98%	56.47%
Max Drawdown Duration	668	1365
Trades per Year	3.1	

Trade

Trade Winning %	45%
Average Trade %	1.25%
Average Win %	5.37%
Average Loss %	-2.15%
Best Trade %	14.35%
Worst Trade %	-6.22%
Worst Trade Date	TBD
Avg Days in Trade	0.0
Trades	31

Time

Winning Months %	72%
Average Winning Month %	2.62%
Average Losing Month %	-4.56%
Best Month %	11.19%
Worst Month %	-9.79%
Winning Years %	60%
Best Year %	41.24%
Worst Year %	-12.00%

This strategy's maximum daily draw-down is 32% less than the benchmark, a reduction in "risk". The Sharpe ratio is 0.48 because of the relative simplicity. However, the major metric is the reduction in number of trades to 31 from 41 previously. The strategy eliminated trades with large negative returns, implying less volatility. No trades were made from early 2008 to mid-2009. The benefit is clearly visible here. The strategy did not lose money when other would have in the same time period.

CONCLUSION:

An overview of Markov Models their mathematical framework and Hidden Markov Models and a simple application was reviewed in this paper. A simple MA strategy was devised and run with and without the HMM regime detection filter. The strategy that incorporated the risk management filter outperformed that without a regime detection filter. Applications of HMM to regime detection are slightly tricky as there is no labelled data to train. Also we need to guess on number of regimes present, which in turn depends on underlying asset-class, the time period and quality of information. Therefore, using HMM requires extensive research and substantial knowledge of asset class being modelled. The model above predicts only state transitions given it has seen prior distributions. The model needs to be trained again if the distribution or regime changes. How frequently the retraining is done, is subject of future research.