HIDDEN MARKOV MODELS FOR VIX INDEX PREDICTION

Author : Paul Serre

github.com/PdePazanan/Hidden-Markov-Model-for-VIX-index-prediction-trading

pserre76@gmail.com

Contents

1 Introduction

Markov chains are significantly used today in the market to benefit investors and traders. In this paper, we shall learn to analyze how a Hidden Markov Model would help us identify different hidden states in the market and use it to develop a trading strategy.

The model estimates three hidden states—normal trading, top formation, and bottom formation—using a number of market observations—returns, price levels, volatility, and momentum. It tracks shifting market structures while being sensitive to different volatility and regime stability by dynamically updating state probabilities using Bayesian inference and adaptive transition matrices.

Risk-adjusted performance is constructed to be confronted, emission probabilities capturing the probability of price behavior in different regimes, with transition probabilities adapting to new information. Signals are validated by outlier detection and sensitivity thresholds to reduce false alarms and coordinate trades with statistically significant patterns.

Lastly, this model enables the proper identification of the stages of exhaustion and capitulation, which facilitates better timing of entry and exit. For a hedge fund, it provides a mathematical decision-making model that balances mathematization with trading judgment to improve both alpha generation and risk management.

2 Hidden Markov Model

2.1 Definition

2.2 Markov chains

In order to understand the HMM, we need first to go througth the markov chains. A Markov chain is a stochastic model hat represents a sequence of events or states. Let S be a finite or countable set and (X_n) a stochastic process with values in S, then X is a Markov's chain if for $n \in \mathbb{N}$, and for all $x_0, x_1, ..., x_n+_1) \in S, P\big(X_0 = i_0, \dots, X_n = i_n\big) > 0$,

$$P\big(X_{n+1} = i_{n+1} \,\big|\, X_0 = i_0, \dots, X_n = i_n\big) = P\big(X_{n+1} = i_{n+1} \,\big|\, X_n = i_n\big) \tag{1}$$

This means that the probability of transitioning to a certain state at time step n + 1 only depends on the current state at time step m and not on any previous states. A Markov chain is defined by the set of states S and a transition probability matrix $A = [a_{ij}]$,a_{ij} represents the probability of transitioning from state S_i to state S_j in one step

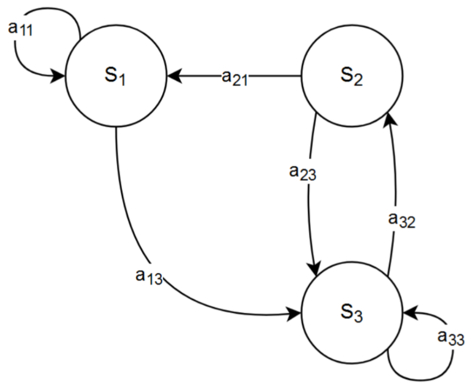

Figure 1: Markov chain with 3 states

Let $x_n \in \mathbb{R}^{\mathbb{N}}$,be the vector containing the probabilities of being in each state at time n, the system evolves according to

$$x_{n+1}^T = x_n^T A \tag{2}$$

2.3 Hidden Makov Model

In opposition of a markov chain where the states are seen and observable, in an HMM, the states are hidden and we deduct them thanks to observations. The hidden state process of a HMM is a Markov

chain.
Let $O = \{O_k \mid O_k \in O,\ k = 1, \ldots, T\}$ be an observed sequence where O is the set of possible observations. The hidden states are evolving according to (2)

The structure of an HMM is defined by his differents characteristics

- λ is a parametric set
- N is the number of hidden states in the model. The set of the hidden states is $S = (S_1, ..., S_N)$ and the state at time t by y_t
- M is the number of distinct symbols observable per state. Let these symbols be o_k where k=1,2,...,M and the observation at time t by O_t
- $A = \{a_{ij}\}$ is the transition matric for the hidden states where :

$$a_{ij} = P\big(Y_{t+1} = S_j \,\big|\, Y_t = S_i\big), \quad i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, N$$

- $B = \{b_s; (o_k)\}$ is the probability matrix of observations k in the state S_i where :

$$b_s(o_k) = P(O_t = o_k \mid Y_t = S_i), \quad i = 1, 2, \ldots, N,\ k = 1, 2, \ldots, M$$

- $\mu = \{\mu_i\}$, the distribution of the initial state of the model where :

$$\mu_i = P(y_1 = S_i),\ i = 1, 2, \ldots, N \text{ and } k = 1, 2, ..., M$$

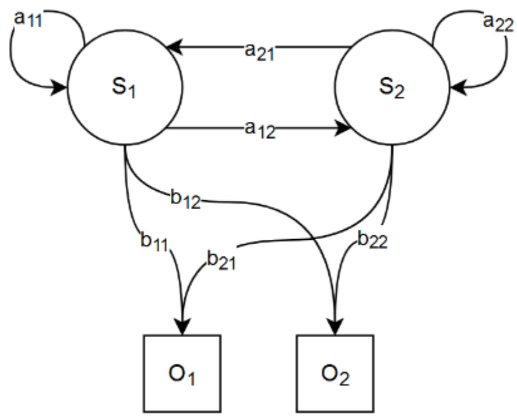

Figure 2: Hidden Markov model chain with 2 states and 2 possible outputs

2.4 Theory of HMM in finance

Markov chains are an important mathematical tool in stochastic processes. The underlying idea is the Markov Property, in order words, that some predictions about stochastic processes can be simplified by viewing the future as independent of the past, given the present state of the process. This is used to simplify predictions about the future state of a stochastic process. A Hidden Markov Model is simply a Markov chains but with hidden states. In a financial market, the transition matrix represents the likelihood of the future evolution of the prices. The transition matrix will describe the probability that a stock or index will either remain in his current state, or transition into a new state.

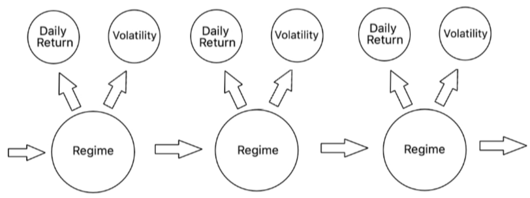

Figure 3: Applying the Hidden Markov Model to market regimes

3 HMM applied to the VIX index

3.1 Details of our HMM

Now that we understand the theory, we will now dive into the model i built on the basis of a Hidden Markov Model. The model continuously analyzes market behavior using multiple dimensions: price returns, volatility, momentum, and intraday patterns. It classifies the market into three distinct states: normal trading, top formation, and bottom formation. These states are determined dynamically through Bayesian updating, combining prior probabilities with real-time observations to adapt to evolving market conditions. The HMM will identify the underlying market structure. This forward-looking approach can signal regime changes before they become apparent in price action,

3.2 Observations of the market

As we saw precedently, to build our model we need observations from the market that will allow us to find hidden states. To do this, we chose the following features of the market :

Features	**definition**	**Formula**
Logarithmic returns	it is a way of measuring the percentage change in the value of an asset over a period of time, it measure the relative change in the value of an asset.	$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$
Rate of return	determine the percentage change from the beginning of the period until the end	$\lvert R_t \rvert = \left\lvert \frac{P_t - P_{t-1}}{P_{t-1}} \right\rvert$
absolute price change		$v_t = P_t - P_{t-1}$
Price acceleration	It is the discrete second derivative	$a_t = R_t - R_{t-1} = \frac{P_t - P_{t-1}}{P_{t-1}} - \frac{P_{t-1} - P_{t-2}}{P_{t-2}}$
Price jerk	discrete third derivative	$j_t = a_t - a_{t-1} = P_t - 3P_{t-1} + 3P_{t-2} - P_{t-3}$
local price position	where the current price stands relative to the highest and lowest prices within our specified look-back period.	$\frac{P_t - \min(P_{t-n+1}, ., P_t)}{\max(P_{t-n+1}, ., P_t) - \min(P_{t-n+1}, ., P_t)}$
local high position	normalized position of the highest peak within a specified window of high-frequency data	$\frac{H_t - \min(H_{t-n+1}, ., H_t)}{\max(H_{t-n+1}, ., H_t) - \min(H_{t-n+1}, ., H_t)}$
local low position	normalized position of the lowest peak within a specified window of high-frequency data	$\frac{L_t - \min(L_{t-n+1}, ., L_t)}{\max(L_{t-n+1}, ., L_t) - \min(L_{t-n+1}, ., L_t)}$
price gap	represents the calculated price gap for the current period.	$\text{price\_gap}_t = \frac{\lvert \text{Open}_t - \text{Close}_{t-1} \rvert}{\text{Close}_{t-1}}$
intraday range	show the price movement within a single trading day	$\text{Intraday\_Range} = \frac{\text{High} - \text{Low}}{\text{Close}}$
body ratio	represents the proportion of the candlestick's "body" (the difference between open and close) relative to its "full range" (the difference between high and low).	$\text{body\_ratio} = \frac{\lvert\text{Close} - \text{Open}\rvert}{\text{High} - \text{Low} + 10^{-6}}$
recent volatility	Mesure of recent volatility	$\text{recent\_vol}_t = \frac{1}{10}\sum_{i=0}^{9}\left\lvert \frac{P_{t-i} - P_{t-i-1}}{P_{t-i-1}} \right\rvert$
mean price	rolling (moving) average of the close price at time t	$\text{mean\_price}_t = \frac{1}{L}\sum_{i=0}^{L-1}[\text{close}]_{t-i}$

Algorithm 1 Feature for Price Data

Compute log returns and magnitude:
$df[log\_return] \leftarrow \log\left(\frac{df[close]}{df[close].shift(1)}\right)$
$df[return\_magnitude] \leftarrow |df[log\_return]|$
Compute price derivatives:
$df[price\_velocity] \leftarrow df[close].diff()$
$df[price\_acceleration] \leftarrow df[price\_velocity].diff()$
$df[price\_jerk] \leftarrow df[price\_acceleration].diff()$
Compute local normalized positions:
$df[local\_price\_position] \leftarrow \_normalize\_price(df[close], lookback\_period)$
$df[local\_high\_position] \leftarrow \_normalize\_price(df[high], lookback\_period)$
$df[local\_low\_position] \leftarrow \_normalize\_price(df[low], lookback\_period)$
Compute gaps and intraday metrics:
$df[price\_gap] \leftarrow \frac{|df[open]-df[close].shift(1)|}{df[close].shift(1)}$
$df[intraday\_range] \leftarrow \frac{df[high]-df[low]}{df[close]}$
$df[body\_ratio] \leftarrow \frac{|df[close]-df[open]|}{df[high]-df[low]+10^{-6}}$
Compute recent volatility:
$df[recent\_volatility] \leftarrow df[close].pct\_change().abs().rolling(10).mean()$
$df[rv\_sma10] \leftarrow df[recent\_volatility].rolling(10).mean()$
$df[rv\_sma15] \leftarrow df[recent\_volatility].rolling(15).mean()$
$df[rv\_sma20] \leftarrow df[recent\_volatility].rolling(20).mean()$
Compute mean price and deviations:
$df[mean\_price] \leftarrow df[close].rolling(lookback\_period).mean()$
$df[price\_deviation] \leftarrow \frac{df[close]-df[mean\_price]}{df[mean\_price]+10^{-9}}$
$df[abs\_deviation] \leftarrow |df[price\_deviation]|$

3.3 Hyperparameters of the model

The following parameters are the hyperparameters of the model; we need to choose them carefully to adapt to our market. These parameters can be different depending on the market in which we are in.

Hyperparameters	**Description**
lookback period	Number of bars used to normalize the price, calculate volatility, and maintain a moving average of probabilities.
learning rate	The rate at which transition probabilities are adjusted to reflect changes in the market. It influence a fast learning but less precise than a slow learning more precise
outlier smoothing	Window for smoothing outlier detection (SMA on signal_strength).
outlier sensitivity	Multiplier to set the threshold for outlier detection (standard deviation multiplier).
sensitivity	Minimum probability threshold for a state (top or bottom) to be considered dominant.

3.4 States of the model

With these observations of the market, and after some steps we will be able to find the hidden states of the market. Before lauching our algorithm we need to define these states. I want to concentrate on three different states on the VIX index market that i find relevant :

State 0, **Normal Trading**: Market continuation patterns, balanced buying/selling
State 1, **Top Formation**: Exhaustion Patterns at Price Highs, the market is overbought
State 2, **Bottom Formation**: Capitulation patterns at price lows, the market is oversold

3.5 The algorithm

3.5.1 Emissions

Sometimes, in a HMM the emissions probabilities are modeled as a Gaussian Mixture Models (GMMs), but here, we will use heuristic emissions that i have defined manually using historical data and statistic of the VIX index.

For each bar, we calculate the probability of observing the current market values if the market state was top, bottom, or normal:

- Factors for top: high price, negative acceleration, positive cumulative price pressure, high volatility, gaps, bullish price streaks
$$e_T = \frac{p_T + a_T + cp_T + v_T + g_T + s_T}{6}$$
- Factors for bottom: low price, positive acceleration, negative cumulative price pressure, high volatility, gaps, bearish price streaks
$$e_B = \frac{p_B + a_B + cp_B + v_B + g_B + s_B}{6}$$
- Factors for normal: price in the middle range, moderate volatility, low price pressure, short price streaks
$$e_N = \frac{v_N + p_N + cp_N + s_N}{4}$$

with

- $p_{B/T/N}$ the position probability
- $a_{B/T/N}$ the acceleration probability
- $cp_{B/T/N}$ the pressure probability
- $v_{B/T/N}$ the volatility probability
- $g_{B/T/N}$ the gap probability
- $s_{B/T/N}$ the streak probability

These probabilites are calculated thanks to the observations of the market. they are then combined to yield the values top_emission, bottom_emission, and normal_emission.

3.5.2 States Transitions probabilities

State Transition Probabilities quantify the likelihood of the market moving from one hidden state to another between consecutive time steps. Mathematicaly :

$$T_{i \to j} = P(S_t = j \mid S_{t-1} = i)$$

Where:

- S_{t-1} is the market state at time t-1,
- S_t is the market state at time t,

- $T_{i \to j}$ is the probability that state i transitions to state j.

Between each time period t we have a transition matrix. The initial transition matrix is defined manually with the probability to transit between the 3 states as described below :

$$T^{(0)} = \begin{bmatrix} T_{T\to T} & T_{T\to N} & T_{T\to B} \\ T_{N\to T} & T_{N\to N} & T_{N\to B} \\ T_{B\to T} & T_{B\to N} & T_{B\to B} \end{bmatrix}$$

The initial probabilities are set based on historical data, market intuition and prior calibration.

This system updates the transition probabilities over time to adapt to current market conditions. The update of the matrix per new bar is

$$T^{(t+1)} = \text{Normalize}\Big(\text{Clip}(T(t) + \alpha\,\Delta T(t), T_{\min}, T_{\max})\Big)$$

Where:

- $\Delta T(t)$ is the change suggested by recent market observations or confirmed signals,
- α is the learning rate controlling how quickly the model adapts,
- Clip: ensures probabilities stay within a reasonable range $\mathrm{T}_{i\to j}^{(t+1)} \in [T_{\min}^{i\to j}, T_{\max}^{i\to j}]$, preventing extreme jumps,
- the normalization ensures each row sums to 1 (valid probability distribution), $\sum_j (T_{i\to j}^{(t+1)}) = 1$

In matricial form we have the following formula :

$$\mathbf{T}^{(t+1)} = \begin{bmatrix} \text{clip}(T_{N\to N}^{(t)} + \alpha\Delta_{N\to N}) & \text{clip}(T_{N\to T}^{(t)} + \alpha\Delta_{N\to T}) & \text{clip}(T_{N\to B}^{(t)} + \alpha\Delta_{N\to B}) \\ \text{clip}(T_{T\to N}^{(t)} + \alpha\Delta_{T\to N}) & \text{clip}(T_{T\to T}^{(t)} + \alpha\Delta_{T\to T}) & \text{clip}(T_{T\to B}^{(t)} + \alpha\Delta_{T\to B}) \\ \text{clip}(T_{B\to N}^{(t)} + \alpha\Delta_{B\to N}) & \text{clip}(T_{B\to T}^{(t)} + \alpha\Delta_{B\to T}) & \text{clip}(T_{B\to B}^{(t)} + \alpha\Delta_{B\to B}) \end{bmatrix}$$

3.5.3 Calculation of priors and Bayesian updating

The market state at time t is hidden: we cannot observe it directly. What we do observe are market features (returns, volatility, positions, streaks, etc.).

Bayesian Updating provides the mathematical framework to combine prior beliefs about the market state with new information from observations.

The output is a set of posterior probabilities for each state (Normal, Top, Bottom), which guide trading signals.

- The prior probabilities are calculated on the basis of the previous probabilities and the transition matrices between states.
$$P(S_t) = \sum_{S_{t-1}} P(S_t \mid S_{t-1})\, P(S_{t-1})$$
- Bayesian updating combines the prior probabilities and the observed data to obtain the posterior probabilities:
$$P(S \mid O) \;=\; \frac{P(O \mid S)\, P(S)}{\sum_{S'} P(O \mid S')\, P(S')}$$

Algorithm 2 Bayes update and Sliding Memory Update

$unnorm_{normal} \leftarrow prior_{normal} \times emission_{normal}$
$unnorm_{top} \leftarrow prior_{top} \times emission_{top}$
$unnorm_{bottom} \leftarrow prior_{bottom} \times emission_{bottom}$
$total\_prob \leftarrow unnorm_{normal} + unnorm_{top} + unnorm_{bottom}$
if $total\_prob > 0$ **then**
  $prob_{normal} \leftarrow \dfrac{unnorm_{normal}}{total\_prob}$
  $prob_{top} \leftarrow \dfrac{unnorm_{top}}{total\_prob}$
  $prob_{bottom} \leftarrow \dfrac{unnorm_{bottom}}{total\_prob}$

$prev\_probs\_normal$.append$(prob_{normal})$
$prev\_probs\_top$.append$(prob_{top})$
$prev\_probs\_bottom$.append$(prob_{bottom})$

3.5.4 Calculation of the outlier line

Finally, we calculate the outlier line who is a **dynamic threshold** who represent an anormality on the price of the market, this will allow us to combine this line to the top and bottom probability so that determines when a probability is statistically significant enough to trigger a trading signal :

$$\text{outlier\_line}_t = \text{mean}_t + \lambda \times \text{std}_t$$

where:

- λ is the outlier sensitivity parameter,
- mean_t and std_t are computed over the rolling window.

For bottom signals, the sign can be reversed if a downward deviation detection is used:

$$\text{outlier\_line}_t = \text{mean}_t - \lambda \times \text{std}_t$$

This creates a statistical boundary above (or below) which a probability is considered exceptional.

Algorithm 3 Compute Signal Strength and Outlier Line

Convert probabilities to percentages:
$df[top\_prob\_value] \leftarrow df[prob\_top] \times 100$
$df[bottom\_prob\_value] \leftarrow df[prob\_bottom] \times 100$
Compute signal strength:
$df[signal\_strength] \leftarrow \max(df[top\_prob\_value], df[bottom\_prob\_value])$ along axis 1
Compute rolling baseline and standard deviation:
$baseline \leftarrow df[signal\_strength].rolling(outlier\_smoothing).mean()$
$signal\_std \leftarrow df[signal\_strength].rolling(outlier\_smoothing).std()$
Compute outlier line:
$df[outlier\_line] \leftarrow (baseline + signal\_std \times outlier\_sensitivity)\ .rolling(10).mean()$

3.5.5 Determining the dominant state and signals

- If prob_top > prob_bottom and prob_top > sensitivity threshold then there is a possible top, so a possible sell signal
- If prob_bottom > prob_top and prob_bottom > sensitivity threshold then there is a possible bottom, so a possible buy signal

To validate these signals, we calculate the outlier line on the most strong probability (top or bottom). The signals are triggered only if the probability of top and bottom are above this line, like showed in the Figure 9

After these steps, the transition matrices are adjusted based on confirmed signals and recent volatility, making the HMM more responsive to market movements.

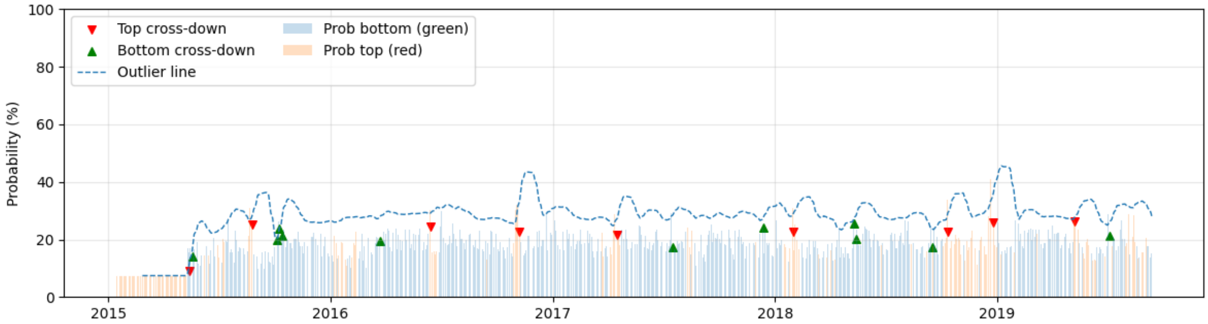

Figure 4: Trading strategy on daily timeframe

4 Backtest and results

Here are some results of the backtest. We defined the hyperparameters manually, with optuna on a train period. Then we have tested our model on a period of 5 years. From 2021 to october 2025. I didn't take in account the fees form the broker or to enter a position. I put a slippage of 0.2% on every trade, this way, our backtest is more realistic. We want to do daily trading, because we don't want to pay the rollover cost and VIX futures contracts have a maturity of one month, so our trades must be made over a shorter period than this.

I tested 2 different models, one who is better in a stable with low volatility market and the other who take advantage of high peak of volatility and robust in extreme market conditions, but weak in low volatility market.

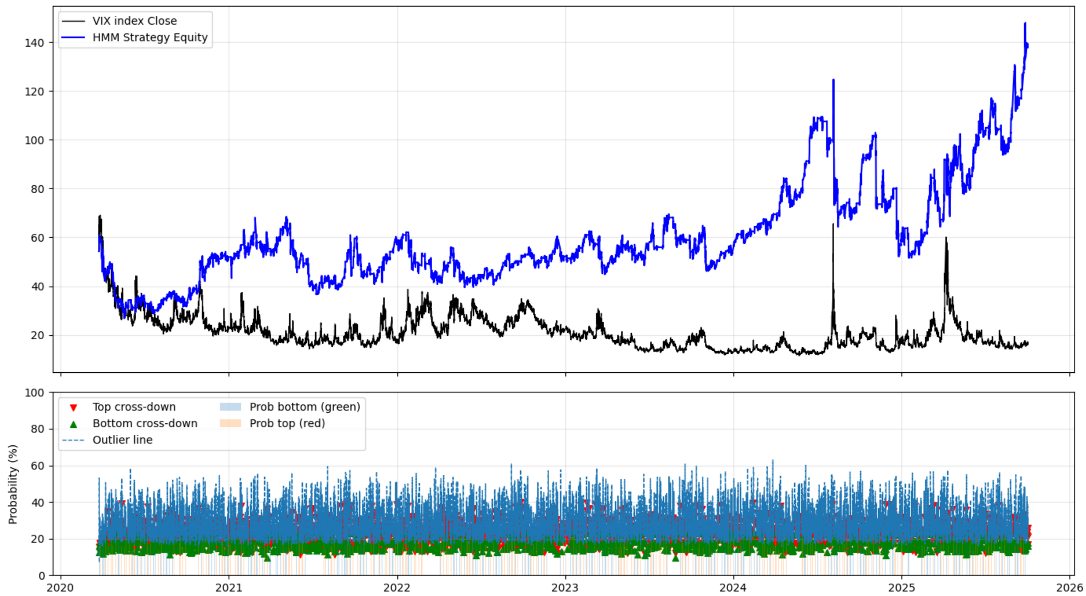

Figure 5: daily strategy 1

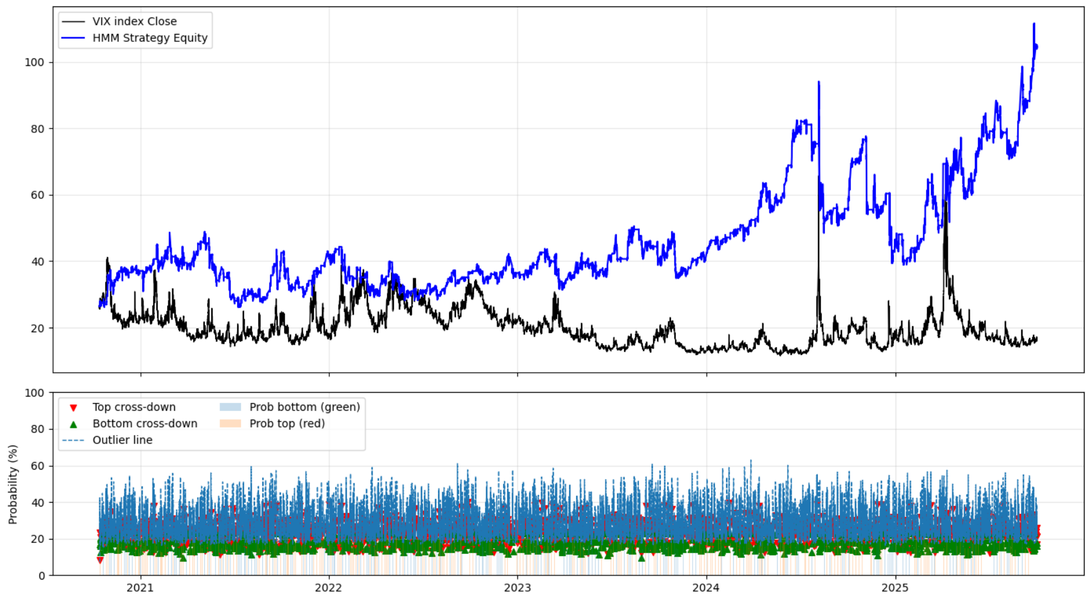

Figure 6: Daily strategy 2

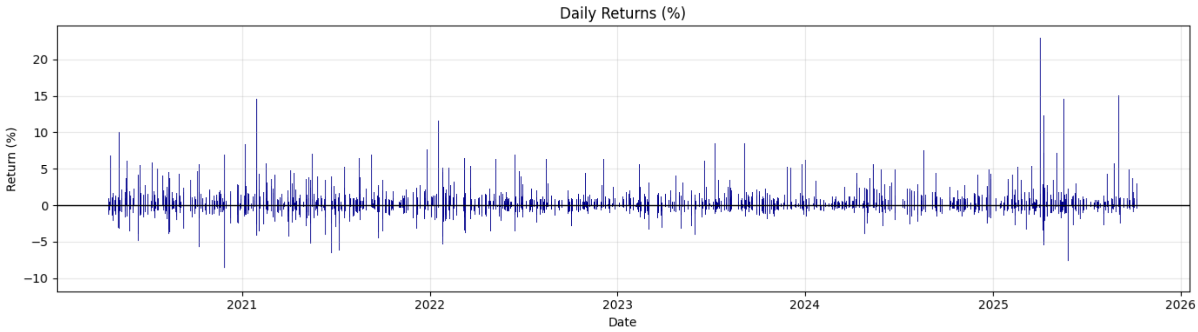

Figure 7: daily returns strategy 3

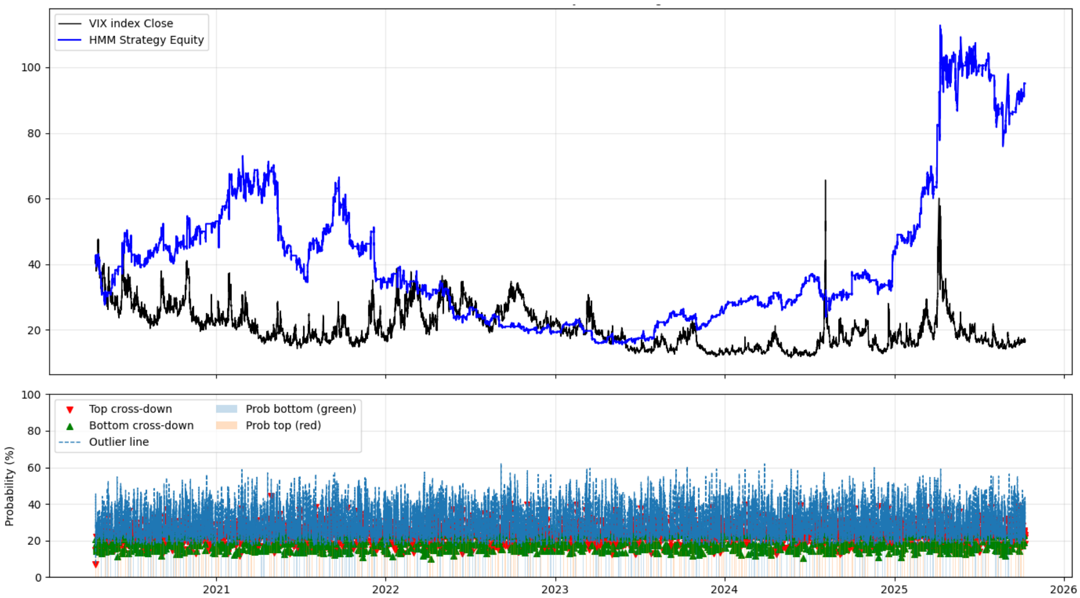

Figure 8: Trading strategy 3(2020-2025)

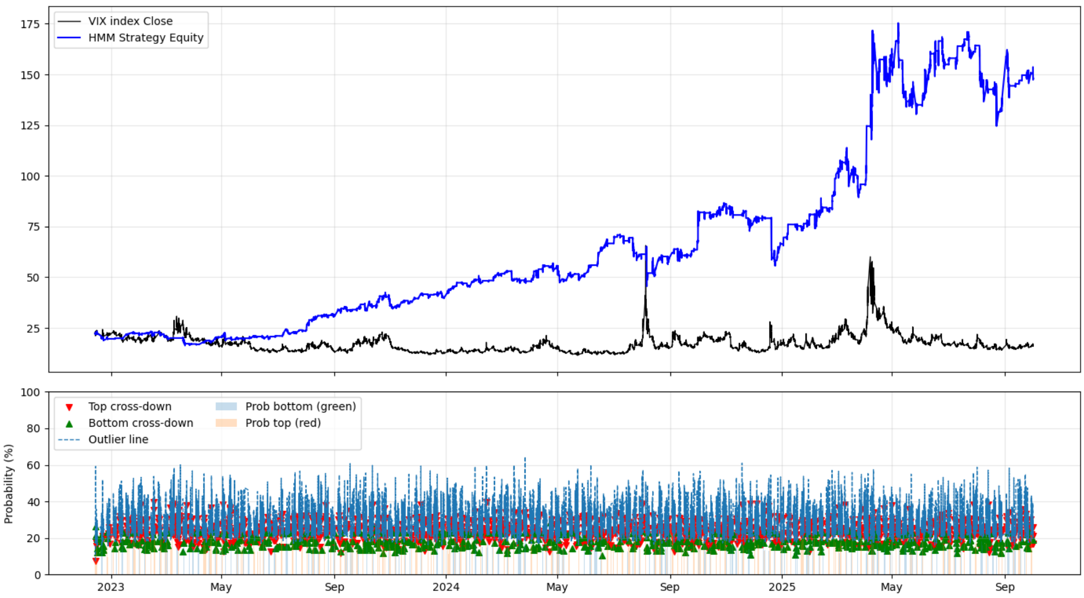

Figure 9: strategy 3 (zoom on the period 2023-2025)

With the third trading strategy, we see that when the market is calm with just 2 high peak of volatility (2023-2025) our model stay still and don't have large drawdowns. But when there is constantly high volume (2020-2023) our model see difficulties and lose money.

Performance backtest 5 years	**HMM strategy 1**	**HMM strategy 2**
Total P&L	294 %	155 %
annual P&L	32 %	18 %
Annualized volatility	5 %	5 %
Sharpe ratio	5.58	2.85
Average drawdown	23 %	25 %
Max drawdown	58 %	59 %
numbers of Buy signals	1452	1658
numbers of sell signals	4151	4628
Average duration of a trade	25h	24h
Average profit of a trade	0.34%	0.26%
numbers of trades	284	312

Thanks for reading !

5 References

[1] Luigi Catello, Ludovica Ruggiero, Hidden Markov Models for Stock Market Prediction, July 2023

[2] Matthew Wang, Yi-Hong Lin, Ilya Mikhelson, Regime-Switching Factor Investing with Hidden Markov Models, december 2020

[3] R. Pardo, The Evaluation and Optimization of Trading Strategies, John Wiley Sons, 2008

[4] E.P. Chan, Quantitative Trading, John Wiley Sons, 2009