Hidden Markov Model for Trading

Alassane Diallo [1,✉]

[1] Quantitative Researcher , Paris France

The objective of this study is to evaluate the efficacy of a Hidden Markov Model (HMM) in the context of quantitative trading. To assess its performance, an HMM-based trading strategy was developed and compared against a simple buy-and-hold strategy, using gold spot data. The results demonstrate the significant potential of the HMM in predicting market dynamics and outperforming the traditional strategy.

Correspondence: *dialloalassann@yahoo.fr*

Motivation

A regime change refers to a significant shift in market conditions, such as a transition from a bull market to a bear market or from periods of high volatility to low volatility, among others. Understanding and modeling these shifts is crucial for making optimal trading decisions. In the literature, several models have been developed to detect regime changes among others [(1), (2), (3),(4), (5), (6)], etc.. One of the most notable is the Gaussian Hidden Markov Model, which classifies different regimes within a dataset into hidden states representing the different regimes.

Mathematical Framework

Here is presented a mathematical framework for classifying market regimes using a Gaussian Hidden Markov Model (HMM). The model consists of hidden states representing different market regimes (e.g., bull, bear, or sideways), with market observations being modeled as Gaussian distributions.

A. **Hidden States** Z_t**:**. Let Z_t denote the hidden state at time t, which represents the market regime at time t. We assume there are n possible hidden states corresponding to different market regimes, i.e., (2))

$$Z_t \in \{1, 2, \ldots, n\}$$

where each Z_t corresponds to a market regime (e.g., bull, bear, neutral).

B. **Observations** X_t . At each time t, we observe data X_t that is generated by the hidden state Z_t. Let X_t represent the market return, price, or other market variables such as volatility or momentum. The distribution of the observed data X_t depends on the hidden state Z_t. Given $Z_t = k$, the observation X_t is assumed to follow a Gaussian distribution with mean μ_k and covariance Σ_k :

$$P(X_t \mid Z_t = k) = \mathcal{N}(X_t \mid \mu_k, \Sigma_k)$$

where: - μ_k is the mean return for regime k, - Σ_k is the covariance (or variance if 1 D) of the observations in state k.

C. **Transition Probabilities** A . he transition probabilities describe the likelihood of moving between hidden states from one time step to the next. The transition probability matrix $A = [A_{ij}]$ is an $n \times n$ matrix, where A_{ij} represents the probability of transitioning from state i to state j :

$$A_{ij} = P(Z_{t+1} = j \mid Z_t = i)$$

The matrix A must satisfy the following condition:

$$\sum_{j=1}^{n} A_{ij} = 1 \quad \text{for all} \quad i \in \{1, 2, \ldots, n\}$$

D. **Initial State Probabilities** π . The initial state probability vector $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ represents the probability of starting in each state. It is a probability distribution, so:

$$\sum_{i=1}^{n} \pi_i = 1$$

where $\pi_i = P(Z_0 = i)$.

E. **Training the HMM: Baum-Welch Algorithm**. To estimate the parameters of the model (transition probabilities, emission parameters, and initial state probabilities), we use the Baum-Welch algorithm, which is a special case of the Expectation-Maximization (EM) algorithm. - E-Step: Compute the expected values of the hidden states Z_t given the observations $X_1, X_2, \ldots, X_T$, and the current estimates of the model parameters.
We compute the forward and backward probabilities: - Forward variable $\alpha_t(i) = P(X_1, X_2, \ldots, X_t, Z_t = i \mid \theta)$ - Backward variable $\beta_t(i) = P(X_{t+1}, X_{t+2}, \ldots, X_T \mid Z_t = i, \theta)$ - M-Step: Update the parameters μ_k, Σ_k, A_{ij}, and π_i using the expected values computed in the E-step: - Update the transition matrix A using the expected transitions between states:

$$A_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

where $\gamma_t(i,j)$ is the probability of transitioning from state i to state j at time t. - Update the emission parameters μ_k and Σ_k using the expected observations given the hidden states:

$$\mu_k = \frac{\sum_{t=1}^{T} \gamma_t(k) X_t}{\sum_{t=1}^{T} \gamma_t(k)}, \quad \Sigma_k = \frac{\sum_{t=1}^{T} \gamma_t(k)(X_t - \mu_k)(X_t - \mu_k)^T}{\sum_{t=1}^{T} \gamma_t(k)}$$

- Update the initial state probabilities π using:

$$\pi_i = \gamma_1(i)$$

where $\gamma_1(i)$ is the probability of starting in state i. These steps are iteratively applied until the model parameters converge.

F. **Classifying Market Regimes: Viterbi Algorithm**. Once the model parameters are trained, we use the Viterbi algorithm to classify the most likely sequence of hidden states $Z_1, Z_2, \ldots, Z_T$ given the observed data $X_1, X_2, \ldots, X_T$. The Viterbi algorithm finds the most probable sequence of hidden states by maximizing the joint probability:

$$\hat{Z}_1, \hat{Z}_2, \ldots, \hat{Z}_T = \arg \max_{Z_1, Z_2, \ldots, Z_T} P(Z_1, Z_2, \ldots, Z_T \mid X_1, X_2, \ldots, X_T$$

This can be done recursively using dynamic programming. The probability of the most likely state sequence up to time t for each state i is:

$$\delta_t(i) = \max_{Z_1, Z_2, \ldots, Z_{t-1}} P(Z_1, Z_2, \ldots, Z_t = i \mid X_1, X_2, \ldots, X_t)$$

The Viterbi algorithm efficiently computes this using:

$$\delta_t(i) = \max_j (\delta_{t-1}(j) A_{ji}) P(X_t \mid Z_t = i)$$

and the corresponding backtracking to recover the sequence of most likely hidden states $Z_1, Z_2, \ldots, Z_T$.

Application in Gold data

This dataset provides historical values of gold spot prices (XAUUSD).The dataset spans over 1,000 rows of data, tracking price movement patterns from January 2021 to December 2024.

Fig. 1. Gold spot price (XAUUSD) from January 2021 to December 2024. The prices fluctuate between 1600 and 2800 $

Predictive Analysis. In order to evaluate the performance of the HMM model, it was first trained on a training dataset and evaluated on a test data.

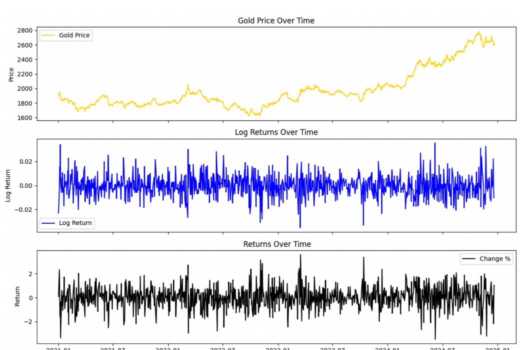


Fig. 2. The blue image depicts the distribution of log-returns, while the black image represents the distribution of simple returns. Both visuals highlight clear patterns of volatility clustering, suggesting the potential existence of multiple market regimes.

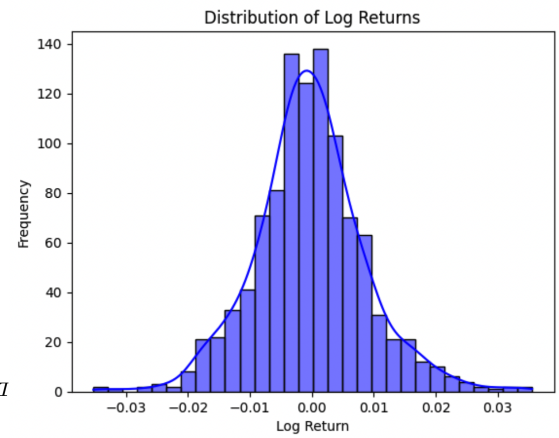


Fig. 3. This represents the distribution of the log-returns covered by a continuous kernel

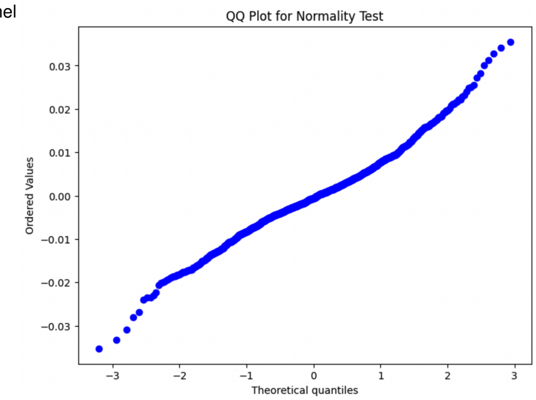


Fig. 4. The QQ plot below compares the quantiles of the log-return distribution with the theoretical quantiles of a Gaussian distribution.

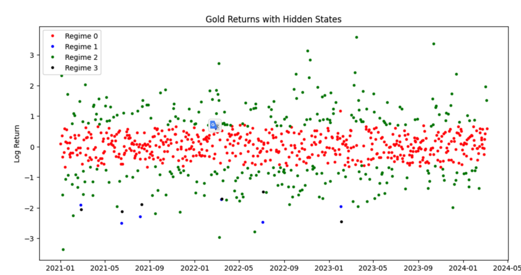


Fig. 5. The Hidden Markov Model applied to the training set identifies an optimal number of four regimes. The first regime (Red) corresponds to high positive returns with moderate variability. The second regime (Green) reflects stable returns near zero with minimal variability. The third regime (Blue) is associated with low, mostly negative returns, tightly clustered. Finally, the fourth regime (Black) is characterized by extreme negative returns with high variabi

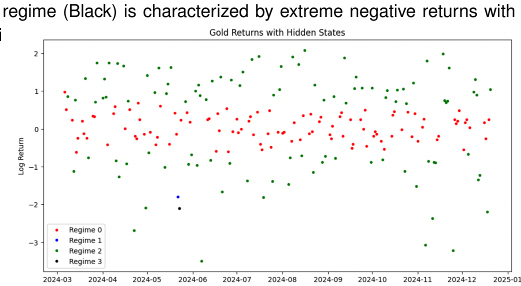


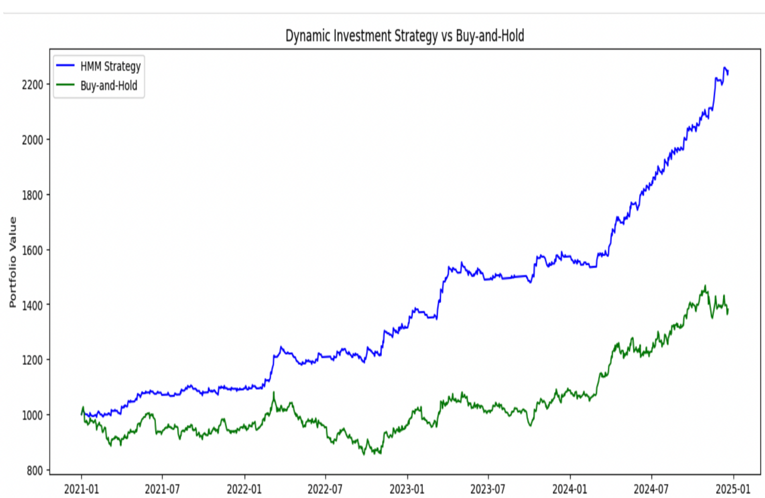
Fig. 6. Hidden Markov Model used to predict the regime in the test data.

Performance Indicator. The performance of the Hidden Markov Model (HMM) was tested using the following trading strategy.

1. Set Initial Conditions:
   - Start with a portfolio value of $1000.
   - Set the annual risk-free return to 3%.
   - Set a 0.4% transaction fee for switching between risky and stable market states.
2. For each day in the data:
   - Get the current market state (e.g., risky or stable) and the percentage change in market price.
3. If the market is risky (states 0, 1, or 3):
   - The portfolio earns the equivalent of the risk-free return.
4. If the market is stable (state 2):
   - If the market goes up (positive return):
     – Keep the portfolio invested in the same assets.
   - If the market goes down (negative return):
     – Switch the portfolio to a risk-free investment.
5. If the market state changes (from risky to stable or vice versa):
   - Deduct 0.4% from the portfolio value to account for the transaction cost of switching.
6. Update the portfolio value at the end of the day.
7. Repeat for all days in the data.

Results. This trading strategy was then compared with a buy-and-hold trading strategy. We get the following table summarizing the results .:

Metric	HMM Strategy	Buy-and-Hold
Sharpe Ratio	2.26	0.56
Max Drawdown	-0.05	-0.21
Annualized Return	0.22	0.08
Annualized Volatility	0.09	0.14
Calmar Ratio	4.12	0.38



Conclusions

This investigation aimed to evaluate the performance of a Hidden Markov Model (HMM) trading strategy using gold spot data. The strategy, developed with the HMM, was compared to a traditional buy-and-hold approach. The results showed that the HMM strategy outperformed the buy-and-hold strategy by a significant margin, demonstrating the potential of this model. However, for practical deployment, the model needs to undergo further validation through extensive testing across various asset classes and datasets.

Bibliography

1. Rogemar S Mamon and Robert James Elliott. *Hidden Markov models in finance*, volume 4. Springer, 2007.
2. Nguyet Nguyen. Hidden markov model for stock trading. *International Journal of Financial Studies*, 6(2):36, 2018.
3. Iman Abu Hashish, Fabio Forni, Gianluca Andreotti, Tullio Facchinetti, and Shiva Darjani. A hybrid model for bitcoin prices prediction using hidden markov models and optimized lstm networks. 2019.
4. Luigi Catello, Ludovica Ruggiero, Lucia Schiavone, and Mario Valentino. Hidden markov models for stock market prediction. 2023.
5. Mengqi Zhang, Xin Jiang, Zehua Fang, Yue Zeng, and Ke Xu. High-order hidden markov model for trend prediction in financial time series. *Physica A: Statistical Mechanics and its Applications*, 517:1–12, 2019.
6. Peng Chen, Dongyun Yi, and Chengli Zhao. Trading strategy for market situation estimation based on hidden markov model. *Mathematics*, 8(7):1126, 2020.