Modeling Financial Market Regimes with Hidden Markov Models

**Monica Whiteside**

Zephyr Analytics

# Abstract

This paper presents a comprehensive methodology for identifying and forecasting latent financial market regimes using Hidden Markov Models (HMMs). Our approach begins with the construction of a feature set that captures both momentum and volatility across multiple time horizons, two factors widely recognized in the literature for their predictive capacity in market behavior (Jegadeesh & Titman, 1993; Corsi, 2009). The HMM is configured with multivariate Gaussian emissions to model these features, allowing the system to infer unobserved market states over time. Following state inference, regime sequences are evaluated for stability using transition metrics to ensure robust classification. Forecasting is performed by propagating the inferred state through the model’s transition dynamics, yielding a probabilistic outlook over a forward horizon.

To enhance the practical utility of these regime classifications, we perform hierarchical clustering on recent regime sequences across assets, revealing groups of assets that behave similarly under market conditions. These clusters are used to structure the portfolio construction process, where we aggregate forecasted regime probabilities and apply category-based weight adjustments to form a diversified and regime-aware portfolio. This multi-layered approach is grounded in prior research on regime-switching models (Hamilton, 1989; Ang & Bekaert, 2002) and has been demonstrated to outperform static asset allocation strategies, particularly in non-stationary market environments (Maheu & McCurdy, 2000; Guidolin & Timmermann, 2007).

By integrating statistical modeling, stability diagnostics, forecast propagation, and cluster-based asset grouping, this framework provides a comprehensive tool for understanding and acting on market regimes. The result is a principled, interpretable system for allocation decisions that adapts to the evolving dynamics of financial markets.

# 2. Conceptual Overview of the HMM in Finance

Understanding the theoretical foundations of Hidden Markov Models (HMMs) is essential before applying them to financial regime detection. HMMs offer a principled approach to modeling time-series data where the system dynamics are governed by unobserved states. This capability aligns well with financial markets, where investors observe returns, volatility, and other indicators without direct access to the true market regime. By treating market regimes as latent states and observed financial variables as emissions from those states, the HMM provides a probabilistic structure that captures both temporal dependencies and structural shifts in financial behavior.

## 2.1. What Is an HMM?

A Hidden Markov Model is a statistical model composed of hidden states and observable outputs. Each state represents a latent condition that influences the observable financial data. The model consists of three main components: the initial state distribution, the state transition matrix, and the emission probabilities. Formally, at each time step t, the system resides in a state S\_t ∈ {1, ..., K}, and emits an observation x\_t, typically drawn from a multivariate normal distribution N(μ\_{S\_t}, Σ\_{S\_t}). The transition between hidden states is governed by a Markov chain, where the probability of moving from one state to another depends only on the current state, not the sequence of previous states. A Hidden Markov Model consists of a system that evolves through a set of hidden states, each of which generates observable data according to a probability distribution. In the context of financial modeling, the hidden states represent latent market regimes, and the observed data include indicators such as asset returns and volatility. The model estimates the likelihood of being in a particular state, the probabilities of transitioning from one state to another, and the distribution of observations given each state.

## 2.2. Application to Financial Markets

In the context of finance, the hidden states of the HMM are interpreted as unobservable market regimes such as bullish, bearish, or neutral. These regimes manifest through observable financial indicators like asset returns and volatility. Empirical studies have shown that regimes differ in their statistical properties, particularly in the mean and variance of returns. By engineering features such as momentum and volatility from price series, and modeling their joint distribution within the HMM framework, we can learn the structure and dynamics of these regimes. Once trained, the HMM infers the most likely regime for each time point and helps forecast future regime probabilities based on the learned transition dynamics. The HMM is especially well-suited for financial data due to its ability to model regime-switching behavior. In this application, hidden states can be interpreted as market conditions such as bullish (characterized by high momentum and low volatility), bearish (marked by declining returns and rising volatility), or neutral (indeterminate or transitioning phases). To capture these characteristics, we use engineered features that reflect momentum and volatility dynamics. These features are then used to train the HMM to identify and label time periods according to the underlying market regime.

# 3. Supporting Literature

A strong foundation in previous research validates the approach taken in this study. Hidden Markov Models have a well-established presence in both theoretical and applied finance, particularly for modeling non-stationary, regime-switching dynamics that traditional linear models struggle to capture. In this section, we review foundational work that has informed the structure and application of the model presented here.

## 3.1. HMMs in Financial Regime Analysis

The seminal work by Hamilton (1989) introduced Markov-switching models in the context of macroeconomic cycles, demonstrating that observed fluctuations in output could be better understood through latent regimes. This concept has since been extended to asset markets, where returns and volatilities also exhibit regime-like behaviors. Guidolin and Timmermann (2007) provided empirical evidence that portfolio strategies adjusted to regime changes improve performance. Ang and Bekaert (2002) further advanced this by incorporating regime switches into international asset pricing models, improving the explanatory power for global equity returns. The use of regime-switching models in economics began with Hamilton (1989), who introduced Markov-switching autoregressive models for business cycle analysis. Building on this, Guidolin and Timmermann (2007) showed that asset allocation can be optimized by incorporating regime-dependent dynamics. Similarly, Ang and Bekaert (2002) demonstrated that regime-switching models provide improved explanations for time-varying risk premia and return predictability in global equity markets.

## 3.2. Volatility and Momentum as Regime Signals

The selection of momentum and volatility as primary features is supported by extensive empirical work. Jegadeesh and Titman (1993) showed that momentum is a persistent anomaly that can be exploited across markets and time periods, indicating it reflects more than transient noise. Corsi (2009) introduced long-memory models that account for the persistent and clustered nature of volatility, which makes it an excellent candidate for differentiating between regimes. Ding, Granger, and Engle (1993) showed that volatility exhibits autocorrelation at multiple horizons, making its historical patterns a predictive signal for future market behavior. Momentum and volatility are widely recognized as important indicators of market state. Jegadeesh and Titman (1993) found momentum strategies to be robust anomalies in return predictability, while Corsi (2009) developed long-memory models to better capture volatility clustering. Ding, Granger, and Engle (1993) provided additional support for volatility modeling by showing that volatility exhibits persistence across time horizons, reinforcing the case for dynamic regime modeling.

## 3.3. Practical Implementations

Beyond theory, numerous practical applications confirm the viability of HMMs in real-world financial modeling. Lo, Mamaysky, and Wang (2000) applied HMMs to identify technical patterns in asset prices, showing that such probabilistic models outperform heuristic-based rules. Maheu and McCurdy (2000) used HMMs to segment financial time series into bull and bear markets, offering superior forecasting accuracy compared to GARCH models. Mitra and Mitra (2011) demonstrated how HMM-based state predictions can be used as inputs to dynamic asset allocation, enhancing portfolio performance during transitional market conditions. Applications of HMMs have extended to pattern recognition in asset prices, as explored by Lo, Mamaysky, and Wang (2000). In forecasting applications, Maheu and McCurdy (2000) showed that HMMs outperform traditional GARCH models in identifying bull and bear markets. Mitra and Mitra (2011) integrated HMMs into portfolio optimization, demonstrating their utility in constructing regime-aware investment strategies.

# 4. Mathematical Formulation and Data Pipeline

This section outlines the technical implementation of the model, including the transformation of raw price data into engineered features, the mathematical formulation of the HMM, and the procedures for inference, forecasting, and evaluating model stability. This pipeline enables both classification of market regimes and prediction of future conditions using a structured statistical approach.

## 4.1. Input Data

The raw input to the model consists of historical adjusted closing prices, denoted as , where . These prices serve as the basis for calculating derived features that are more informative for regime detection.

## 4.2. Feature Engineering

To model latent states, we construct a feature vector for each time step. This vector includes two key features: momentum and volatility. Momentum is calculated as the average of compounded returns over 3, 6, 9, and 12-month windows. Volatility is measured as the standard deviation of daily returns over 1-month and 3-month rolling windows. To make volatility comparable across time and assets, it is normalized between zero and one.

**Daily returns:**

**Momentum:**

Where is the adjusted closing price at time , and is the lookback window in trading days (3, 6, 9, and 12 months).

**Volatility:**

Rolling standard deviations:

Normalized volatilities:

Combined volatility:

## 4.3. HMM Specification

The HMM consists of hidden states , a transition matrix defining probabilities of moving between states, and an initial state distribution . Each state emits a vector drawn from a multivariate normal distribution with state-specific mean and covariance . This generative structure allows the model to probabilistically infer which latent regime most likely generated the observed feature at each time point.

**State transition probability:**

**Emission probability (Gaussian):**

## 4.4. Inference and Forecasting

Inference involves estimating the most likely sequence of hidden states that generated the observed data. This is typically done using the Viterbi algorithm, which maximizes the joint posterior over the hidden state sequence. Once the model is trained and the states are inferred, forecasting involves computing the future distribution of states by raising the transition matrix to the power of the number of forecast steps and applying it to the current state vector.

**Most likely sequence of states (Viterbi):**

**Forecasting future state distribution after steps:**

Where is the state distribution at time , and is the transition matrix.

## 4.5. Stability Evaluation

To assess the quality of the model, we evaluate the stability of the inferred state sequences. This involves calculating the transition rate, defined as the ratio of state changes to total time steps. Models with high instability (frequent switching) are penalized and may trigger retraining. The stability check helps ensure the regime sequences are meaningful and not driven by noise.

**Transition rate:**

Where is the number of regime changes and is the total number of time steps.

# 5. Portfolio Construction via Clustering and Forecast Aggregation

The portfolio construction process builds on the output of the HMM-based regime classification and forecast. It leverages hierarchical clustering to group assets that share similar regime behavior and constructs a regime-aware portfolio with weights influenced by cluster structure and forecast distributions.

## 5.1. Hierarchical Clustering of Regime Sequences

Each asset's recent regime sequence is treated as a time series of categorical states. These state labels are converted into numerical sequences using label encoding to facilitate quantitative analysis. All sequences are aligned to a common length (typically 252 trading days) to ensure comparability. Pairwise Euclidean distances are calculated to measure similarity between sequences, and Ward’s linkage method is used to generate a hierarchical clustering tree. Cutting the tree at a predefined threshold partitions the assets into distinct clusters that exhibit similar regime behaviors.

## 5.2. Forecast Distribution Aggregation by Cluster

After clustering, the next step is to aggregate the regime forecast distributions from all assets within a cluster. Each asset's HMM provides a probability distribution over future regimes, which are mapped to common categories (Bullish, Neutral, Bearish) based on the model's state labels. Within each cluster, these probabilities are summed and normalized to produce a representative outlook for the entire cluster. This outlook determines the weight each cluster receives in the final portfolio.

## 5.3. Weighting and Asset Selection

Within each cluster-category pair, a two-step allocation procedure is followed. First, clusters receive weights in proportion to their contribution to each regime category (e.g., how much of the total Bullish probability is explained by Cluster 1). Second, assets within a cluster are filtered to remove those with excessive Bearish probability (greater than 15%). The remaining assets are assigned weights based on their relative Bullish strength. If no eligible assets remain in a cluster, its weight is redistributed to the rest of the portfolio in proportion to existing allocations.

## 5.4. Final Portfolio Construction and Visualization

The portfolio weights across all selected assets are normalized to ensure the total adds to one. These weights are then visualized using a pie chart, offering an intuitive view of the allocation. The final portfolio reflects a structured integration of forecasted regime strength, cluster similarity, and individual asset outlook, resulting in a forward-looking, risk-aware investment allocation.

# 6. Implementation

## Model Training:

At the center of this module is a **Gaussian Hidden Markov Model (HMM)**, a generative probabilistic model designed to represent systems that evolve over time through unobserved discrete states, each of which governs the distribution of observed data. In financial time series analysis, this framework is highly effective for modeling regime-switching behaviors, where markets may transition between latent states like expansion, contraction, or high-volatility phases. The HMM assumes that at each time step, an observation (e.g., asset return characteristics) is drawn from a multivariate Gaussian distribution whose parameters depend on the current hidden state. Transitions between hidden states follow a first-order Markov process, meaning the next state depends only on the current state, not the entire history.

Before the model can be trained, the raw financial data is transformed through a **feature engineering** process tailored to capture structural behaviors of asset prices. Momentum is calculated as a compounded return over multiple lag intervals (e.g., 1-month, 3-month, 6-month, 9-month). This gives a forward-looking signal reflecting cumulative price appreciation or depreciation, acting as a proxy for trend strength. Volatility is measured as the standard deviation of log returns over a rolling window, capturing local uncertainty or risk. These two features—momentum and volatility—are selected because they tend to separate regimes well: for instance, high momentum and low volatility might correspond to bull markets, while the opposite often signifies bearish regimes.

Once these features are extracted, they are normalized using **standard score normalization (z-score scaling)** to ensure each has zero mean and unit variance. This preprocessing step is essential because the Expectation-Maximization (EM) algorithm used by HMMs is sensitive to the scale of the input features; unnormalized data can cause instability or poor convergence during training.

To initialize the Gaussian HMM parameters, the module employs **K-Means clustering**, an unsupervised learning method that assigns each observation to one of k clusters by minimizing within-cluster Euclidean distances. The rationale here is to use cluster centroids as proxies for the means of the Gaussian emissions in each latent state. Moreover, within-cluster variance is computed and used to initialize the diagonal covariance matrices. This provides the HMM with informed priors that dramatically reduce the likelihood of poor convergence compared to random initialization. The initial state probabilities and transition matrix are set uniformly, reflecting a non-informative prior where each state and transition is equally likely at the outset.

The model fitting process uses the **EM algorithm**, a two-step iterative process. In the E-step (Expectation), the algorithm computes the posterior probabilities over the hidden states given the current parameters—this is done using the Forward-Backward algorithm, which leverages dynamic programming to efficiently calculate the likelihoods across all time steps. In the M-step (Maximization), the parameters of the model (state transition probabilities, means, and covariances of each Gaussian) are updated to maximize the expected log-likelihood of the complete data. This continues iteratively until convergence, typically assessed via the change in log-likelihood between iterations falling below a threshold (e.g., 1e-5).

After the model converges, the trained HMM is used to infer the most likely sequence of hidden states for the training data through the **Viterbi algorithm**, a dynamic programming method that identifies the single most probable path through the state space. This sequence of states represents the model’s classification of the market regime at each time step in the training set.

To assess the quality of this state sequence, the module applies a **state stability evaluation** function. This likely involves statistical heuristics such as the minimum dwell time (i.e., how long the model remains in a state before switching), transition entropy, or intra-state variance. Models that switch states too frequently or produce unstable regimes (e.g., splitting what appears to be a coherent trend across multiple states) are flagged as unstable. In such cases, the training process is retried with the same data but a fresh initialization, leveraging the non-determinism of EM to potentially reach a better local optimum.

If the model produces a stable and interpretable set of hidden states, each state is then **labeled**—a process that may involve sorting states by average momentum or volatility, or applying domain-specific logic to identify, for instance, which state corresponds to bull, bear, or sideways markets. These labels enhance the interpretability of the model, turning abstract statistical regimes into actionable insights.

Finally, the trained model, now fully parameterized and labeled, is serialized and saved for future use. This model can then be used to infer market regimes on unseen data or serve as a foundational layer in more complex decision-making systems, such as algorithmic trading strategies or portfolio optimization frameworks.

In totality, this module orchestrates a tightly integrated application of unsupervised machine learning, Bayesian inference, and time series analysis to create a sophisticated system for detecting latent structure in financial markets. It emphasizes robustness (through retries and evaluation), statistical rigor (via EM and likelihood convergence), and interpretability (through feature design and regime labeling)—all critical features for effective deployment in real-world financial applications.

## Model Inferencing:

Once a Gaussian Hidden Markov Model (HMM) has been trained to identify latent regimes in financial time series data, the next critical phase is inference—the process of applying the trained model to unseen data to uncover hidden state sequences and make forward-looking probabilistic forecasts. This module implements that process, emphasizing not only the accurate mapping of historical patterns but also the projection of future market dynamics through state transition probabilities.

The inference pipeline begins by reinitializing the same structural parameters used during training, including the asset ticker and the date boundaries that define the time horizon for analysis. At the heart of the process is the retrieval of two essential components: the trained HMM and the historical feature data (momentum and volatility) on which the model was originally fitted. These components are deserialized from disk and loaded into memory, ensuring that the inferencing engine has access to the same data distribution and statistical assumptions that governed model training.

The model is then applied to the unseen test dataset, which contains normalized momentum and volatility features for more recent time steps. The **prediction of hidden states** is accomplished via the model’s internal sequence decoder, which computes the most probable state at each time step using the **Viterbi algorithm**. This algorithm finds the single most likely sequence of hidden states given the model parameters and observed data, leveraging dynamic programming to maximize the joint probability of state transitions and emissions. The result is a discrete sequence of states that corresponds to the regime classification of the test period.

After decoding the hidden state sequence, a **state labeling process** is performed to map numeric states to semantically meaningful regime names. This labeling is based on the statistical characteristics of each state—typically derived from prior analysis of the training data—such as average momentum or volatility. For example, a state with high momentum and low volatility might be labeled “Bull,” while the inverse might be labeled “Bear.” This mapping enhances interpretability, allowing users to connect abstract statistical regimes to real-world financial narratives.

A particularly sophisticated aspect of this module is its **forward state forecasting capability**. Instead of merely identifying the current state, the model simulates how the system might evolve over a defined number of time steps (e.g., 21 days, equivalent to one trading month). This is achieved by applying the **state transition matrix**, a core component of the HMM that encodes the probabilities of transitioning from one state to another. Starting from the most recently inferred state, the algorithm iteratively multiplies the current state distribution vector by the transition matrix. Each multiplication represents a single step into the future, and after n steps, the resulting probability vector represents the distribution over all possible states at the forecast horizon.

This probabilistic forecast is then translated back into the labeled regime space, producing a **forecast distribution** that quantifies the likelihood of the market being in each regime in the near future. For instance, the model might conclude there is a 70% chance of remaining in a “Neutral” regime, with a 20% chance of transitioning into “Bear” and a 10% chance of entering “Bull.” Such outputs are invaluable for risk assessment, tactical asset allocation, and decision-making under uncertainty.

Overall, this inferencing system encapsulates the full probabilistic logic of Hidden Markov Models. It combines retrospective analysis (state inference) with forward simulation (forecasting), offering a principled statistical foundation for understanding and predicting financial market behavior. The approach aligns with key tenets of modern time series analysis and regime-switching models, delivering not only interpretability but also actionable foresight based on well-founded probabilistic dynamics.

## Portfolio Construction:

The portfolio construction module represents the final analytical stage in the regime-based modeling framework, integrating probabilistic state inference with unsupervised learning and risk-filtered weighting techniques to construct a dynamically adaptive investment portfolio. This system leverages both statistical state forecasts and structural similarity among assets to allocate capital in a way that aligns with forecasted market conditions and observed historical behavior.

The process begins by aggregating and parsing previously inferred regime data for a set of financial instruments. Each instrument has already undergone regime classification using a Hidden Markov Model (HMM), producing time-indexed sequences of hidden state labels and associated probabilistic forecasts of future states. These sequences, composed of labeled market regimes such as "Bullish," "Bearish," or "Neutral," are extracted and encoded numerically using a label encoding scheme that prepares them for clustering.

To capture structural similarities across assets based on regime behavior, the module performs **hierarchical agglomerative clustering** using cosine distance on the encoded regime sequences. This technique is especially well-suited for identifying similarity in directional patterns rather than magnitude. A linkage matrix is computed using average linkage, and a series of clustering solutions are evaluated across a range of cluster counts. For each candidate solution, clustering quality is assessed using a composite of **internal validation metrics**, including the Silhouette Score (which measures separation and cohesion), the Calinski-Harabasz Index (which evaluates cluster dispersion), and the Davies-Bouldin Index (which penalizes overlapping clusters). These scores are scaled and combined to select the optimal number of clusters in a data-driven fashion.

Each identified cluster represents a group of assets with similar regime sequences over the lookback period. These clusters are then used as categories within the portfolio framework. The next step involves retrieving the **probabilistic forecast distributions** for each asset. These distributions quantify the likelihood of each asset being in a particular market regime (e.g., a 70% chance of being in a Bullish state 21 days into the future). Using this forecast, each asset is evaluated on its **adjusted bullish exposure**, typically computed as the difference between Bullish and Bearish probabilities. A reference asset (e.g., SHV) is used as a conservative benchmark: assets that do not outperform this benchmark in bullish probability are discarded from further allocation.

The cluster-specific weights are determined by aggregating adjusted bullish exposure scores across assets within the same cluster. This results in a **category weight distribution** per cluster and regime, reflecting how strongly a given cluster tilts toward bullish, bearish, or neutral outlooks. These weights are then used to derive a proportional allocation scheme: assets with stronger bullish signals receive a larger share of their cluster’s weight. Assets with insufficient bullish advantage or excessive bearish probability are excluded, and their weights are reallocated among the remaining assets.

An important risk-control layer is introduced via a **Simple Moving Average (SMA) filter**. Each candidate asset must satisfy a momentum confirmation criterion: its current and prior closing prices must lie above its trailing average over a configurable window (e.g., 50 or 200 days). This serves as a safeguard against allocating to assets in a technical downtrend, further reinforcing the robustness of the portfolio.

The final result is a set of normalized weights across selected tickers. If no asset passes the filtering criteria, the portfolio defaults to a conservative allocation (e.g., 100% in SHV) to avoid exposure during uncertain or bearish market conditions. This approach blends unsupervised learning, probabilistic forecasting, and technical validation into a systematic portfolio construction process.

Finally, visual summaries of the portfolio—such as allocation pie charts and cluster dendrograms—are generated and saved for documentation and reporting. These artifacts help investors or analysts verify the construction logic and interpret the resulting asset mix. By integrating regime recognition with predictive distributions and structural clustering, this module enables a machine learning–driven investment process that dynamically adjusts exposure based on evolving market conditions and behavioral similarity among assets

# References

Ang, A., & Bekaert, G. (2002). International asset allocation with regime shifts. *Review of Financial Studies*, *15*(4), 1137–1187.

Bulla, J., & Bulla, I. (2006). Stylized facts of financial time series and hidden semi-Markov models. *Computational Statistics & Data Analysis*, *51*(4), 2192–2209.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, *7*(2), 174–196.

Ding, Z., Granger, C. W. J., & Engle, R. F. (1993). A long memory property of stock market returns and a new model. *Journal of Empirical Finance*, *1*(1), 83–106.

Guidolin, M., & Timmermann, A. (2007). Asset allocation under multivariate regime switching. *Journal of Economic Dynamics and Control*, *31*(11), 3503–3544.

Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica*, *57*(2), 357–384.

Jegadeesh, N., & Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, *48*(1), 65–91.

Lo, A. W., Mamaysky, H., & Wang, J. (2000). Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. *Journal of Finance*, *55*(4), 1705–1765.

Maheu, J. M., & McCurdy, T. H. (2000). Identifying bull and bear markets in stock returns. *Journal of Business & Economic Statistics*, *18*(1), 100–112.

Mitra, G., & Mitra, L. (Eds.). (2011). *Handbook of news analytics in finance*. Wiley.