Regime-Switching, Tail-Risk and Stress-Aware Portfolio

Intelligent Portfolio Management under Hidden Market Regimes

Mhamed Halata

Master 1 – Applied Mathematics for engineering, Algorithmics And Statistics

University of Burgundy, Dijon

Academic Year: 2025–2026

Acknowledgements

I would like to express my gratitude to the professors of the Master 1 in Applied Mathematics at the University of Burgundy for providing the theoretical foundations necessary for this research.

This project represents the culmination of my academic interest in the intersection of advanced algorithmic techniques and financial risk management. It was an opportunity to apply C++ programming and probabilistic theories to real-world market challenges.

Finally, I thank my parents and my girlfriend Lydia for their support throughout this academic year.

Abstract

English: This project proposes a "Stress-Aware" portfolio management framework designed to navigate financial crises. Traditional static models fail to capture the heavy tails and dynamic nature of market returns. To address this, we implemented a **Student-t Hidden Markov Model (HMM)** in C++ to detect unobservable market regimes (Bull, Bear, Crash). This detection engine drives a dynamic asset allocation strategy coded in Python. The backtest performed on the S&P 500 (1985-2023) demonstrates that the strategy significantly outperforms the benchmark, increasing the Sharpe Ratio from 0.47 to 0.89 and reducing the Maximum Drawdown from -56% to -35%. Monte Carlo stress tests further validate the model's robustness, showing a structural reduction in Value-at-Risk (VaR). This work confirms that regime-switching models are effective tools for capital preservation and long-term alpha generation.

Français : Ce projet propose une stratégie de gestion de portefeuille "Stress-Aware" conçue pour résister aux crises financières. Les modèles statiques traditionnels échouent à capturer les queues épaisses et la nature dynamique des rendements boursiers. Pour y remédier, nous avons implémenté un **Modèle de Markov Caché (HMM) Student-t** en C++ afin de détecter les régimes de marché inobservables (Haussier, Baissier, Krach). Ce moteur de détection pilote une stratégie d'allocation dynamique codée en Python. Le backtest réalisé sur le S&P 500 (1985-2023) démontre que la stratégie surperforme significativement le marché, portant le Ratio de Sharpe de 0.47 à 0.89 et réduisant la perte maximale (Drawdown) de -56% à -35%. Les stress tests de Monte Carlo valident la robustesse du modèle, montrant une réduction structurelle de la Value-at-Risk (VaR). Ce travail confirme que les modèles à changement de régime sont des outils efficaces pour la préservation du capital et la génération d'alpha à long terme.

Contents

Introduction

This project aims to develop an adaptive portfolio management strategy that integrates **Regime-Switching**, **Tail-Risk** control, and **Stress Testing**. The idea is to model market behavior using probabilistic and optimization techniques, detecting hidden market regimes, and dynamically adjusting asset allocations accordingly.

The report is structured as follows:

- **Chapter 1:** Financial context and motivation.
- **Chapter 2:** Theoretical framework — mathematical formulation of the model.
- **Chapter 3:** Numerical methodology — simulation, estimation, and optimization.
- **Chapter 4:** Results and economic interpretation.
- **Chapter 5:** Conclusion and future perspectives.

Chapter 1

Financial Context and Motivation

1.1 Introduction

Financial markets are inherently dynamic systems characterized by alternating phases of growth and contraction. Periods of stable expansion with low volatility are often followed by sudden crises marked by high uncertainty and extreme losses. Traditional portfolio management models, such as Markowitz's mean-variance optimization, assume stationary returns and normally distributed risks. However, empirical evidence shows that financial returns exhibit fat tails, volatility clustering, and structural breaks that cannot be captured by static models.

To address these limitations, the concept of **Regime-Switching** has emerged in quantitative finance. This approach assumes that market behavior evolves according to hidden states, or "regimes", each characterized by distinct statistical properties (e.g., high vs. low volatility, positive vs. negative returns). By modeling market dynamics through probabilistic transitions between these regimes, investors can adapt their portfolio allocation in real time and better manage risk during turbulent periods.

1.2 Historical Motivation

The motivation for regime-dependent and tail-risk-aware portfolio models is grounded in several major financial crises that revealed the fragility of traditional risk management frameworks.

1.2.1 The 1987 Crash (Black Monday)

On October 19, 1987, global stock markets experienced one of the largest one-day declines in history: the Dow Jones Industrial Average fell by nearly 23%. This sudden market crash, known as *Black Monday*, occurred without a clear macroeconomic trigger, highlighting the existence of nonlinear feedback mechanisms and volatility contagion. The event exposed the weaknesses of Gaussian-based risk models and led to the development of volatility models such as GARCH and stochastic volatility frameworks. However, these models still assumed a single, stationary regime of market dynamics.

1.2.2 The Dot-com Bubble Burst (2000–2001)

At the end of the 1990s, financial markets entered a speculative regime driven by the rapid growth of Internet-related companies. The *Dot-com Bubble* burst in March 2000, leading to massive losses in technology stocks and a prolonged bear market. This episode illustrated a clear regime transition: from a low-volatility, optimistic phase to a high-volatility, loss-dominated phase. Static portfolio strategies failed to adapt to this change, reinforcing the need for dynamic, data-driven approaches that can detect and react to regime shifts.

1.2.3 The COVID-19 Financial Shock (2020)

In early 2020, the outbreak of COVID-19 triggered one of the fastest and most severe global market crashes in modern history. Within a few weeks, major indices lost more than 30% of their value, volatility reached unprecedented levels, and correlations across assets soared. The pandemic crisis revealed the limitations of standard risk measures such as Value-at-Risk (VaR), which underestimated extreme losses during the crash. This event emphasized the importance of integrating **tail-risk measures** such as Conditional VaR (CVaR) and **stress testing** into portfolio optimization frameworks.

1.3 From Static to Adaptive Portfolio Management

The lessons learned from these crises suggest that financial markets evolve through hidden regimes governed by probabilistic mechanisms. Regime-switching models, often based on Hidden Markov Models (HMMs), provide a natural framework for capturing these dynamics. By associating each regime with distinct return and volatility parameters, the model can infer the current market phase and adapt investment decisions accordingly.

Furthermore, the combination of regime detection, optimization techniques, and extreme-risk modeling creates a robust foundation for modern portfolio management. The objective of this project is therefore to design a **Regime-Switching, Tail-Risk and Stress-Aware Portfolio** model that unifies these three dimensions:

1. Dynamic regime identification using Hidden Markov Models;
2. Regime-dependent portfolio optimization under mean-variance and risk constraints;
3. Extreme-risk evaluation through VaR, CVaR, and stress testing scenarios.

This adaptive framework aims to provide investors with a more resilient and realistic representation of financial risk, especially during crisis periods when traditional models fail.

Chapter 2

Theoretical Framework

Overview

This chapter presents the theoretical foundation of the project. The goal is to build a mathematical and probabilistic model that captures market regime changes, optimizes portfolio allocation accordingly, and integrates advanced risk measures to account for extreme events.

The first part introduces the concept of **Hidden Markov Models (HMMs)**, which provide a statistical framework for modeling market dynamics through unobserved regimes (such as bull and bear markets). The second part formulates the **regime-dependent portfolio optimization problem**, where asset weights are adjusted according to the detected regime. Finally, the chapter reviews the mathematical definitions of **tail risk measures** namely Value-at-Risk (VaR) and Conditional VaR (CVaR) which are crucial for evaluating portfolio robustness under extreme market conditions.

Throughout this chapter, the theoretical results are connected to key mathematical concepts covered in optimization and probability theory: convexity, coercivity, conditional expectation, and convergence of stochastic estimators.

2.1 Definitions

Markov Chain :

Let $(E, \mathcal{E})$ be a measurable space, called the **state space**. A **transition probability** π from E to $\mathcal{E}$ is a function satisfying:

- For every $A \in \mathcal{E}$, the map $x \mapsto \pi(x, A)$ is measurable on E;
- For every $x \in E$, the map $A \mapsto \pi(x, A)$ defines a probability measure on $(E, \mathcal{E})$.

Let μ be an initial probability distribution on $(E, \mathcal{E})$. A sequence of random variables $(X_n)_{n\geq 0}$ taking values in E is called a **Markov chain** with initial law μ and transition probability π if:

$$\begin{aligned} &(1) \quad \mathbb{P}(X_0 \in \cdot) = \mu, \\ &(2) \quad \text{The conditional law of } X_{n+1} \text{ given } X_0, X_1, \ldots, X_n \text{ is } \pi(X_n, \cdot), \quad n \geq 0. \end{aligned}$$

In the case of a finite state space $\mathcal{S} = \{1, 2, \ldots, K\}$, the transition probability can be represented by the **transition matrix**:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1K} \\ p_{21} & p_{22} & \cdots & p_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ p_{K1} & p_{K2} & \cdots & p_{KK} \end{pmatrix}, \quad \text{where} \quad p_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i), \quad \sum_{j=1}^{K} p_{ij} = 1.$$

Hidden Markov Model :

A Hidden Markov Model is a bivariate stochastic process $\{(S_t, r_t)\}_{t\geq 0}$ defined as follows:

- The hidden process $\{S_t\}_{t\geq 0}$ is a Markov chain on a finite state space $\mathcal{S} = \{1, 2, \ldots, K\}$ with transition matrix $P = (p_{ij})$.
- Conditional on $S_t = i$, the observation r_t is generated from a probability distribution $f_i(\cdot)$, called the *emission distribution*:

  $$r_t \mid S_t = i \sim f_i(\cdot; \theta_i),$$

  where θ_i denotes the parameters of the distribution in regime i.

The joint model is thus characterized by the parameter set:

$$\Theta = \{P,\ \pi_0,\ \theta_1, \ldots, \theta_K\},$$

where π_0 is the initial distribution of the hidden state S_0.

The Hidden Markov Model framework is highly versatile and finds applications across numerous domains. **For illustration**, in the context of speech recognition, the hidden states S_t represent unobservable phonetic units (such as vowels or consonants) that generate the observable acoustic signal. The observed variable r_t corresponds to measured audio features at time t, whose distribution depends on the current phonetic state:

$$r_t \mid S_t = i \sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i),$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote respectively the mean vector and covariance matrix of the acoustic features associated with phoneme i.

2.2 Hidden Markov Models for Regime Switching

However, in our financial application, the hidden states S_t represent market regimes and the observations r_t denote the *logarithmic return* of an asset or market index, defined as :

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right),$$

where P_t is the price at time t.
Unless stated otherwise, the term "return" will always refer to this log-return representation, which is additive over time and consistent with the continuous-time models used in quantitative finance.
Conditional on the hidden state $S_t = i$, the observation r_t is assumed to follow a normal distribution:

$$r_t \mid S_t = i \sim \mathcal{N}(\mu_i, \sigma_i^2).$$

This demonstrates the adaptability of the HMM framework while maintaining focus on our financial context.

Application to Market :

In this project, Hidden Markov Models are used to model **financial market regime shifts**. For example:

- **Regime 1 (Bull Market):** Characterized by positive returns ($\mu_1 > 0$) and low volatility (σ_1 low).
- **Regime 2 (Bear Market):** Characterized by negative returns ($\mu_2 < 0$) and high volatility (σ_2 high).

- **Regime 3 (Crash Market):** to capture extreme events such as market crashes (e.g., $\mu_3 \ll 0$, $\sigma_3 \gg \sigma_2$).

The goal is to **infer hidden regimes** from return observations and use this information to dynamically adjust portfolio allocations.

Why Use HMMs in Finance?

- **Adaptability:** HMMs capture **structural changes** in markets (e.g., transitions from bull to bear markets) without assuming stationary return distributions.
- **Interpretability:** The identified regimes have clear **economic meanings** (e.g., "bull market," "bear market"), facilitating decision-making for portfolio managers.
- **Risk Integration:** By associating each regime with specific risk parameters (μ_i, σ_i), HMMs allow for **explicit modeling of stress periods** and dynamic portfolio adjustments.

The diagram below illustrates the transitions between regimes, where each node represents a market regime and each arrow represents a transition probability. These transitions are estimated from historical return data, enabling real-time prediction and response to regime changes.

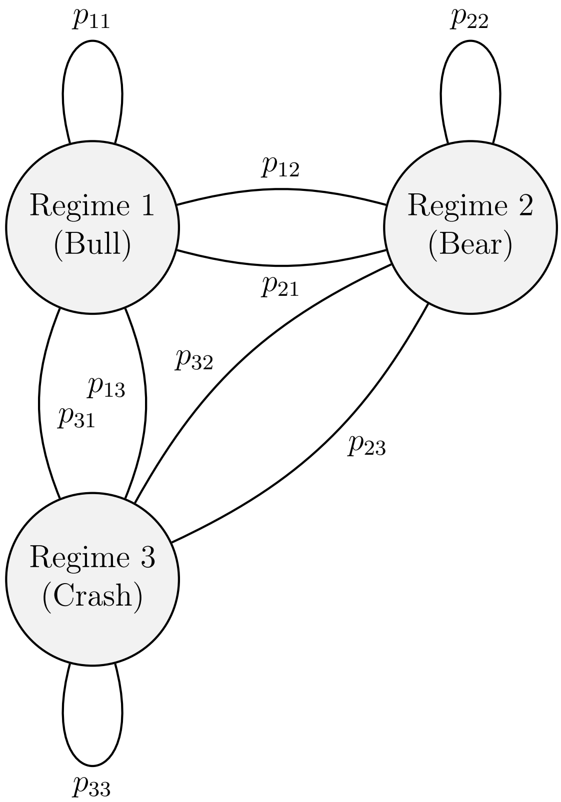

Figure 2.1: Transition structure between three market regimes (Bull, Bear, and Crash). Each arrow p_{ij} represents the probability of moving from regime i to regime j.

2.3 Emission Distributions in Financial Hidden Markov Models

In the context of financial markets, the hidden process $\{S_t\}_{t\geq 0}$ represents the unobservable *market regime* at time t, such as a bull, bear, or crash phase. The observed variable r_t corresponds to the *log-return* of an asset or market index, defined as:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right),$$

where P_t is the market price at time t.

As in other applications of hidden Markov models (for instance in speech recognition), each hidden state $S_t = i$ is associated with a specific probability law that governs the emission of the observation r_t. In a standard formulation, the emission distribution is assumed to be Gaussian:

$$r_t \mid S_t = i \sim \mathcal{N}(\mu_i, \sigma_i^2),$$

where μ_i and σ_i^2 denote respectively the conditional mean and variance of returns in regime i. This implies that, within each regime, returns are symmetric and the likelihood of extreme outcomes decays exponentially fast.

Limitations of the Gaussian assumption : While the Gaussian distribution is mathematically convenient, it is well known to underestimate the probability of extreme market events. For example, during the 1987 market crash (*Black Monday*), the Dow Jones Industrial Average fell by almost -23% in a single day. Under a Gaussian model with a daily volatility of 1%, such a loss corresponds to a 23 standard-deviation event, whose probability is approximately $P(|r_t| > 23\sigma) \approx 10^{-118}$ a theoretically *impossible* outcome. Hence, Gaussian-based risk models failed completely to capture such abrupt regime shifts, motivating the use of heavy-tailed distributions.

Student-*t* emission model : To accommodate the presence of large outliers and fat tails in financial returns, we adopt the **Student-*t*** distribution for the emission law. Conditional on the hidden state $S_t = i$, the return r_t follows:

$$r_t \mid S_t = i \sim t_{\nu_i}(\mu_i, \sigma_i^2),$$

where:

- μ_i : the conditional mean return in regime i;
- σ_i^2 : the conditional scale (volatility) parameter in regime i;
- $\nu_i > 2$: the degrees of freedom controlling the heaviness of the tails. Smaller ν_i values correspond to thicker tails and higher kurtosis.

The Student-t distribution provides heavier tails than the Gaussian distribution, with the tail thickness controlled by the degrees of freedom parameter ν_i. Lower values of ν_i correspond to fatter tails and higher probabilities of extreme returns, better capturing the tail-risk observed in financial markets.

This specification allows the emission law to remain symmetric around μ_i, while providing substantially heavier tails than the Gaussian distribution. It therefore better reflects the empirical behavior of financial returns, which exhibit volatility clustering, kurtosis above 3, and occasional extreme losses. The parameter set of the hidden Markov model in the Student-t framework is thus:

$$\Theta = \{P, \pi_0, (\mu_i, \sigma_i^2, \nu_i)_{i=1}^K\},$$

where P is the transition matrix and π_0 the initial state distribution.

2.4 Modern Portfolio Theory and Asset Allocation

2.4.1 Transition from Regime Detection to Allocation

In the previous sections, we established a robust framework using Student-t Hidden Markov Models (HMM) to identify market regimes $S_t \in \{1, \dots, K\}$ and estimate their specific parameters. The regime-switching approach implies that the market's statistical properties are non-constant, dictating an adaptive asset allocation strategy.

The prescriptive step consists of constructing an optimal portfolio based on these regime-dependent forecasts. We rely on the Modern Portfolio Theory (MPT), but we must explicitly acknowledge that the inputs to MPT are conditional on the current market state.

2.4.2 Regime-Dependent Portfolio Metrics

For a given regime $s \in \{1, \dots, K\}$, the expected return vector $\overline{R}$ and the covariance matrix Q are no longer static. They become $\overline{R}^{(s)}$ and $Q^{(s)}$, where s is the index of the detected regime (Bull, Bear, or Crash).

- **Expected Return (Conditional):** The mean return of the portfolio, conditioned on the market being in state s at time t, is:

$$\mathbb{E}(w|S_t = s) = W^T \overline{R}^{(s)} \tag{2.1}$$

- **Portfolio Variance (Conditional):** The risk structure, conditioned on state s, is defined by the regime-specific covariance matrix $Q^{(s)}$:

$$Var(w|S_t = s) = W^T Q^{(s)} W \tag{2.2}$$

This framework allows us to estimate the future risk and return of the portfolio by using a **weighted average** of the regime-conditional parameters, based on the forecasted probabilities $P(S_{t+1} = s|\mathcal{F}_t)$. The optimization problem is thus solved at each time step t using the forecast derived from the HMM.

2.4.3 The Optimization Problem

The objective of the rational investor is to find the vector of weights W that minimizes risk for a required level of expected return r.

General Formulation of the Regime-Adaptive Optimization

If short selling is allowed, the Regime-Adaptive Mean-Variance optimization problem for a target return r involves minimizing the *expected* variance over all possible next states s', where the expectation is taken with respect to the transition probabilities:

$$\min_W \sum_{s'=1}^{K} P(S_{t+1} = s'|S_t) \cdot \frac{1}{2} W^T Q^{(s')} W \tag{2.3}$$

subject to the constraints:

$$W^T \mathbf{1} = 1 \quad \text{and} \quad \sum_{s'=1}^{K} P(S_{t+1} = s'|S_t) \cdot W^T \overline{R}^{(s')} \geq r$$

2.4.4 Tail-Risk Management: VaR and CVaR

Given the project's focus on **Tail-Risk**, relying solely on variance as a measure of risk (which penalizes symmétriquement les gains et les pertes) is insufficient, especially in the presence of heavy tails modeled by the Student-t distribution. We must introduce metrics that quantify the severity of extreme losses.

Value at Risk (VaR)

The Value at Risk at a confidence level α is defined as the quantile of the portfolio's loss distribution such that the probability of the loss exceeding VaR_α is α.

$$\text{VaR}_\alpha(W) = \inf\{l \in \mathbb{R} : P(L > l) \leq \alpha\} \tag{2.4}$$

where L is the loss of the portfolio.

Conditional Value at Risk (CVaR)

The Conditional Value at Risk, also known as Expected Shortfall (ES), is a more robust risk measure than VaR because it is coherent (convex and sub-additive). CVaR_α represents the expected loss, given that the loss exceeds the VaR_α:

$$\text{CVaR}_\alpha(W) = \mathbb{E}[L|L > \text{VaR}_\alpha(W)] \tag{2.5}$$

In our HMM framework, the CVaR is particularly relevant as the Student-t model explicitly handles the heavy tails of the loss distribution, allowing for a more accurate estimation of the expected magnitude of losses in the high-volatility regime. The success of our "Stress-Aware" strategy will be measured by its ability to significantly reduce the CVaR compared to a static Buy & Hold portfolio.

Chapter 3

Numerical Methodology

Overview

Having established the mathematical formulation of our regime–switching model, we now turn to its practical implementation. The transition from theory to computation is particularly important in the context of Hidden Markov Models, where the latent structure requires dedicated inference algorithms for state estimation and parameter calibration.

In this chapter, we provide a detailed presentation of the computational tools needed to estimate the model from financial time series. We begin with the Forward–Backward algorithm, which computes posterior state probabilities, followed by the Expectation–Maximization (Baum–Welch) procedure used to calibrate transition probabilities and regime-specific Student-t parameters. The Viterbi algorithm is then used to recover the most likely sequence of market regimes.

Specification of Emission Probabilities

Unlike standard HMMs that assume Gaussian emissions, our model accounts for the leptokurtic nature of financial returns. To ensure numerical stability and avoid the pitfalls of estimating tail parameters on limited data, we follow the empirical findings of Hurst & Platen (1997) and fix the degrees of freedom parameter at $\nu = 4$.

Let r_t be the log-return at time t. The probability density function (PDF) for a regime j parameterized by mean μ_j and volatility σ_j is given by the Student-t density:

$$b_j(r_t) = f(r_t|S_t = j) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu}\sigma_j}\left(1 + \frac{1}{\nu}\left(\frac{r_t - \mu_j}{\sigma_j}\right)^2\right)^{-\frac{\nu+1}{2}} \tag{3.1}$$

This density $b_j(r_t)$ replaces the Gaussian likelihood in the standard Forward-Backward recursion.

3.1 Algorithms :

3.1.1 The Forward-Backward algorithm :

The Forward-Backward algorithm acts as the inference engine. It computes the posterior probability of the hidden state given the observations.

- **Forward variable $\alpha_t(j)$:** Probability of observing the sequence up to time t and being in state j.
- **Backward variable $\beta_t(j)$:** Probability of observing the future sequence from $t+1$ to T given state j at time t.

These variables are computed recursively (as detailed in standard HMM literature) and are essential for the Expectation-Maximization step.

Algorithm 1 Forward-Backward Algorithm (Inference)

Require: Returns $R = \{r_1, \dots, r_T\}$, Parameters $\Theta = \{A, \pi, \mu, \sigma, \nu = 4\}$
Ensure: Forward variables α, Backward variables β

1. Forward Pass:
for $j = 1$ to K **do**
  $\alpha_1(j) \leftarrow \pi_j \cdot b_j(r_1)$ {Initialization using Student-t density}
end for
for $t = 2$ to T **do**
  for $j = 1$ to K **do**
    $\alpha_t(j) \leftarrow \left(\sum_{i=1}^{K} \alpha_{t-1}(i) A_{ij}\right) \cdot b_j(r_t)$
  end for
end for
2. Backward Pass:
for $i = 1$ to K **do**
  $\beta_T(i) \leftarrow 1$ {Initialization}
end for
for $t = T - 1$ down to 1 **do**
  for $i = 1$ to K **do**
    $\beta_t(i) \leftarrow \sum_{j=1}^{K} A_{ij} \cdot b_j(r_{t+1}) \cdot \beta_{t+1}(j)$
  end for
end for
return Matrices α and β

3.1.2 The Expectation-Maximization (EM) Algorithm :

Direct maximization of the log-likelihood is intractable due to the unobserved nature of the regimes. We therefore employ the Baum-Welch algorithm, a special case of the EM algorithm. This iterative procedure alternates between estimating the hidden states (E-Step) and updating the parameters (M-Step).

E-Step: Expectation :

Using the α and β variables from the Forward-Backward pass, we compute two key probabilities:

1. **Smoothed Regime Probability** ($\gamma_t(j)$): The probability of being in regime j at time t given the entire history.

$$\gamma_t(j) = P(S_t = j | R_{1:T}) = \frac{\alpha_t(j)\beta_t(j)}{\sum_{k=1}^{K} \alpha_t(k)\beta_t(k)} \tag{3.2}$$

2. **Transition Probability** ($\xi_t(i, j)$): The probability of moving from regime i to j at time t.

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(r_{t+1})\beta_{t+1}(j)}{\sum_{k=1}^{K} \sum_{l=1}^{K} \alpha_t(k)a_{kl}b_l(r_{t+1})\beta_{t+1}(l)} \tag{3.3}$$

M-Step: Maximization and Robust Weights :

This step updates the model parameters $\Theta = \{\pi, A, \mu, \sigma\}$. While the transition matrix A is updated using standard counts, the update of the emission parameters (μ_j, σ_j) requires special attention due to the Student-t assumption.

Since we are dealing with heavy-tailed data, a standard weighted average would be sensitive to outliers (crashes). To mitigate this, we introduce a **robustness weight** u_{jt} derived from the EM formulation for t-distributions (Bulla, 2011).

We define the squared Mahalanobis distance δ_{jt}^2 and the weight u_{jt} as:

$$\delta_{jt}^2 = \left(\frac{r_t - \mu_j}{\sigma_j}\right)^2, \qquad u_{jt} = \frac{\nu + 1}{\nu + \delta_{jt}^2} \tag{3.4}$$

Interpretation: The weight u_{jt} acts as an automatic outlier filter. During a market crash, the return r_t is far from the mean μ_j, causing δ_{jt}^2 to be large and u_{jt} to tend towards zero. Consequently, extreme events have a reduced impact on the estimation of the regime's central tendency.

The robust update formulas are thus:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_t(j)u_{jt}r_t}{\sum_{t=1}^{T} \gamma_t(j)u_{jt}} \tag{3.5}$$

$$\hat{\sigma}_j^2 = \frac{\sum_{t=1}^{T} \gamma_t(j)u_{jt}(r_t - \hat{\mu}_j)^2}{\sum_{t=1}^{T} \gamma_t(j)} \tag{3.6}$$

Note that the update for the transition matrix remains standard:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{3.7}$$

The algorithm iterates between the E-step and M-step until the log-likelihood converges.

Algorithm 2 EM Algorithm with Robust Student-t Updates

Require: Returns R, Initial Parameters $\Theta^{(0)}$, Tolerance ϵ
Ensure: Estimated Parameters $\hat{\Theta}$
while $\Delta LogLikelihood > \epsilon$ **do**
  E-Step (Expectation):
  Run FORWARD-BACKWARD with current Θ to get α, β
  for $t = 1$ to T **do**
    for $j = 1$ to K **do**
      $\gamma_t(j) \leftarrow \frac{\alpha_t(j)\beta_t(j)}{\sum_k \alpha_t(k)\beta_t(k)}$ {Posterior probability}
      $u_{jt} \leftarrow (\nu + 1)/(\nu + \delta_{jt}^2)$ {**Robust Weight Calculation**}
      $\delta_{jt}^2 \leftarrow ((r_t - \mu_j)/\sigma_j)^2$
    end for
  end for
  Compute $\xi_t(i, j)$ using Eq. (3.3)
  M-Step (Maximization):
  for $j = 1$ to K **do**
    $\hat{\mu}_j \leftarrow \frac{\sum_{t=1}^T \gamma_t(j) u_{jt} r_t}{\sum_{t=1}^T \gamma_t(j) u_{jt}}$ {Robust Mean Update}
    $\hat{\sigma}_j^2 \leftarrow \frac{\sum_{t=1}^T \gamma_t(j) u_{jt} (r_t - \hat{\mu}_j)^2}{\sum_{t=1}^T \gamma_t(j)}$ {Robust Variance Update}
  end for
  Update Transition Matrix A using ξ
end while
return $\hat{\Theta}$

3.1.3 The Viterbi Algorithm :

Once the model parameters $\hat{\Theta}$ are estimated, we determine the most likely sequence of market regimes $S_{1:T}^*$. This is achieved using the Viterbi algorithm, a dynamic programming approach that maximizes the joint probability of the state sequence and observations.

We define $\delta_t(j)$ as the maximum probability of a state path ending in state j at time t:

$$\delta_t(j) = \max_{s_1, \ldots, s_{t-1}} P(S_1 = s_1, \ldots, S_t = j, r_1, \ldots, r_t | \Theta) \tag{3.8}$$

The algorithm proceeds by induction to find the optimal path, which allows us to historically date the regimes (e.g., identifying exactly when the market switched from "Bull" to "Crash" during the 2020 crisis).

3.2 Regime-Based Asset Allocation Strategy

Having estimated the Hidden Markov Model parameters and decoded the market regimes, we now define the practical investment strategy. The goal is to construct a "Stress-Aware"

Algorithm 3 Viterbi Algorithm (Decoding)

Require: Returns R, Optimal Parameters $\hat{\Theta}$
Ensure: Most likely regime sequence $S^* = \{s_1^*, \ldots, s_T^*\}$
1. Initialization:
for $j = 1$ to K **do**
  $\delta_1(j) \leftarrow \pi_j \cdot b_j(r_1)$
  $\psi_1(j) \leftarrow 0$
end for
2. Recursion:
for $t = 2$ to T **do**
  for $j = 1$ to K **do**
    $\delta_t(j) \leftarrow \max_i[\delta_{t-1}(i) \cdot A_{ij}] \cdot b_j(r_t)$
    $\psi_t(j) \leftarrow \arg\max_i[\delta_{t-1}(i) \cdot A_{ij}]$ {Store backpointer}
  end for
end for
3. Termination & Backtracking:
$s_T^* \leftarrow \arg\max_j[\delta_T(j)]$
for $t = T - 1$ down to 1 **do**
  $s_t^* \leftarrow \psi_{t+1}(s_{t+1}^*)$ {Trace back the best path}
end for
return S^*

portfolio that dynamically adjusts its exposure to risky assets based on the detected market state.

3.2.1 Investment Universe

To isolate the added value of the regime-switching signal, we consider a simplified investment universe consisting of two assets:

1. **The Risky Asset (A_1):** The S&P 500 index, representing the broad equity market. Its return at time t is denoted by $r_{m,t}$.

2. **The Risk-Free Asset (A_0):** A cash equivalent or Treasury bill instrument, yielding a constant daily risk-free rate r_f. We assume an annualized risk-free rate of 2%.

3.2.2 The Dynamic Switching Rule

The allocation strategy is derived from the Viterbi decoding described in Section 3.1.3. Let $S_t^* \in \{0, 1, 2\}$ be the estimated regime at time t, where the states are ordered by increasing volatility (0=Bull, 1=Bear, 2=Crash).

The portfolio weight vector at time t, denoted $W_t = (w_{0,t}, w_{1,t})^T$, is determined by the following "Bang-Bang" switching rule:

$$W_t = \begin{cases} (0,1)^T & \text{if } S_t^* \in \{\text{Bull}, \text{Bear}\} \quad \text{(Full Equity Exposure)} \\ (1,0)^T & \text{if } S_t^* = \text{Crash} \quad \text{(Flight to Safety)} \end{cases} \tag{3.9}$$

Economic Rationale:

- In **Bull** (Low Vol) and **Bear** (Medium Vol) regimes, the equity market offers a positive risk premium. The strategy remains fully invested in the S&P 500 to capture growth.
- In the **Crash** (High Vol) regime, the estimated mean return is negative ($\mu_{Crash} < 0$) and volatility is extreme. The optimal move under a safety-first criterion is to exit the market entirely and preserve capital in the risk-free asset until the storm passes.

3.3 Backtesting and Stress Testing Methodology

3.3.1 Backtesting Protocol

The strategy is evaluated over the historical period from 1985 to 2023. The performance is measured by calculating the cumulative wealth index V_t of the portfolio:

$$V_{t+1} = V_t \times (1 + R_{p,t+1}) \tag{3.10}$$

where $R_{p,t+1}$ is the portfolio return realized at $t+1$:

$$R_{p,t+1} = w_{1,t} \cdot r_{m,t+1} + w_{0,t} \cdot r_f$$

The strategy is compared against a passive "Buy & Hold" benchmark which remains 100% invested in the S&P 500 throughout the period.

3.3.2 Performance Metrics

To assess the quality of the strategy, we compute the following metrics, as defined in the theoretical framework (Chapter 2):

- **Total Return & Annualized Return:** Measures of absolute profitability.
- **Sharpe Ratio:** Defined as $\frac{\mathbb{E}[R_p - r_f]}{\sigma_p}$, it evaluates the risk-adjusted return.
- **Maximum Drawdown (MDD):** The maximum observed loss from a peak to a trough of the portfolio, quantifying the worst-case scenario for an investor.

- **Tail Risk Metrics:** The Value-at-Risk ($\text{VaR}_{95\%}$) and Conditional Value-at-Risk ($\text{CVaR}_{95\%}$) are computed to verify the strategy's effectiveness in cutting the left tail of the return distribution.

3.3.3 Stress Testing Methodology

A key objective of this project is to validate the strategy under extreme stress. In addition to the global backtest, we perform a **Historical Scenario Analysis** ("Crisis Zoom") focusing on two major systemic shocks:

1. **The 2008 Subprime Crisis (Systemic Stress):** We isolate the period from 2007 to 2009 to analyze how the model behaves during a prolonged liquidity crunch.

2. **The 2020 COVID-19 Crash (Exogenous Shock):** We focus on Q1 2020 to test the model's reactivity to sudden, V-shaped market collapses.

This stress testing framework ensures that the "Stress-Aware" claim of the portfolio is empirically justified by its ability to preserve capital during the most critical weeks of financial history.

Chapter 4

Results and economic interpretation

Overview

Before turning to the portfolio allocation results, we must ensure that the underlying Regime-Switching model behaves consistently with economic intuition and empirical facts. This section reviews the data preparation workflow, reports the estimated parameters of the Student-t HMM calibrated in C++, and analyzes the sequence of detected market states. Validating the model at this stage is crucial, as the quality of the portfolio strategies directly depends on the reliability of the inferred regimes.

Why C++? The choice of programming language is not merely a matter of preference but a strategic decision dictated by the computational complexity of the algorithms involved. While Python is the standard for data analysis and visualization, it faces significant limitations when deploying iterative probabilistic models like Hidden Markov Models (HMM) in a production environment.

The Limits of Python: Python is an interpreted, high-level language. While libraries like NumPy or Pandas are optimized (because they are wrapped around C code), writing custom iterative algorithms in native Python incurs a heavy overhead due to:

- **The Global Interpreter Lock (GIL):** Which limits true parallelism.
- **Dynamic Typing:** The interpreter checks variable types at runtime, slowing down loop execution.
- **Loop Inefficiency:** The Baum-Welch algorithm involves nested loops over time T and states K. In pure Python, repeating these loops for thousands of iterations creates a significant bottleneck known as "interpreter overhead."

The C++ Advantage: To overcome these limitations, the core estimation engine of this project was implemented in C++. This language remains the gold standard in the financial industry (High-Frequency Trading, Pricing Libraries) for several reasons:

1. **Execution Speed:** C++ is a compiled language. The code is translated directly into machine instructions, allowing the Baum-Welch algorithm to run orders of magnitude faster than a Python equivalent.
2. **Memory Management:** C++ provides low-level control over memory allocation (via pointers and references), which is crucial when handling large financial time series to avoid memory leaks and optimize cache usage.
3. **Numerical Precision:** The Standard Template Library (STL) and strict typing ensure rigorous numerical stability, which is essential when calculating likelihoods that can easily underflow.

Hybrid Architecture: Consequently, this project adopts a hybrid architecture often seen in investment banks: **Python** is used for its strengths (Data downloading, Preprocessing, and Plotting), while **C++** acts as the high-performance numerical backend for the heavy lifting (Calibration and Decoding).

4.1 Data Description and Preprocessing

4.1.1 Data Description and Preprocessing

The model is calibrated on the S&P 500 (Standard & Poor's 500) index, widely used as a benchmark for the performance of the U.S. equity market. The dataset spans nearly four decades, from 1985 to 2023, covering several major market cycles, including the 1987 crash (Black Monday), the Dot-com bubble, the 2008 Global Financial Crisis, and the COVID-19 shock. Such a long historical window provides a rich testing ground for the detection of regime shifts and extreme events.

Log-return transformation. Raw price series are non-stationary and unsuitable for direct use in a Hidden Markov Model. To stabilize variance and ensure time-additivity, we transform the daily adjusted closing prices P_t into logarithmic returns,

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right).$$

This transformation produces a stationary-like series that better reflects the statistical properties assumed by the Student-t HMM.

Preprocessing pipeline. All data preparation steps are implemented in Python prior to invoking the C++ calibration engine. The preprocessing workflow consists of:

- **Download:** fetching historical price data using the `yfinance` API;
- **Log-return computation:** applying the transformation $r_t = \ln(P_t/P_{t-1})$;
- **Cleaning:** removing `NaN` values produced by the shift operation, ensuring that the C++ HMM engine receives a clean, single-column vector of returns.

Python implementation : An illustrative code snippet used for the preprocessing stage is shown below:

```python
import yfinance as yf
import numpy as np
import pandas as pd

print("Importing data...")
data = yf.download("^GSPC", start="1985-01-01", end="2023-01-01")

if data.empty:
```

```python
    raise ValueError("Error : No data has been imported")

if isinstance(data.columns, pd.MultiIndex):
    print("MultIndex structure detected : Cleaning columns...")
    data.columns = data.columns.get_level_values(0)

col_name = 'Adj Close'
if col_name not in data.columns:
    print(f"Warning: '{col_name}' absent. Using 'Close' instead.")
    col_name = 'Close'

data['Log_Return'] = np.log(data[col_name] / data[col_name].shift(1))

data = data.dropna()

data['Log_Return'].to_csv("input_returns.txt", index=False, header=False)
data.index.to_series().to_csv("input_dates.csv", index=False, header=False)
print("Succesful ! files 'input_returns.txt' and 'input_dates.csv' generated.")
print(f"Return preview :\n{data['Log_Return'].head()}")
```

4.1.2 Empirical Calibration

Calibration Results (Student-t HMM):

The model was calibrated using the Baum–Welch (EM) algorithm with a multi-start procedure (10 random initializations) in order to mitigate the risk of convergence toward suboptimal local maxima. The degrees of freedom of the Student-t emission distribution were fixed at $\nu = 4$, enforcing heavy tails in accordance with the well-documented stylized facts of financial returns.

The numerical core of the model is implemented in C++, following an object-oriented structure in which the `HMM` class encapsulates the forward–backward recursions, the Baum–Welch updates, and the Viterbi decoding. Only selected excerpts are shown below for readability, while the full source code is available in my github.

```cpp
class HMM {
private:
    int K;
    double nu;
    std::vector<std::vector<double>> trans_prob;

```

```

    std::vector<double> means;
    std::vector<double> vars;
    std::vector<double> priors;

    std::vector<double> observations;
    int T;

    double final_log_likelihood = -1e9;
public:
    HMM(int num_states, double degrees_of_freedom);
    void loadData(const std::string& filename);
    void saveResults(const std::string& filename);
    double getStudentProb(double observation, int state_idx);
    void fit(int max_iter = 100, double tol = 1e-6); // Baum-Welch (EM)
    std::vector<int> predict(); // Viterbi (Decode)
     void printParameters() const;
         double getLogLikelihood() const { return final_log_likelihood; }
private:
    void initializeParameters();
    void forwardBackward(std::vector<std::vector<double>>& alpha,
    std::vector<std::vector<double>>& beta,
    std::vector<double>& scales);
};

#endif
```

Estimated emission parameters and transition structure: The C++ calibration engine converged after 107 iterations. To facilitate economic interpretation, the inferred states were automatically sorted by increasing volatility: State 0 (Low Volatility), State 1 (Medium Volatility), and State 2 (High Volatility).

The figure below displays the raw calibration output produced by the C++ engine after running the Baum–Welch (EM) algorithm with a multi-start initialization scheme. The algorithm converged after 107 iterations, and the three inferred states were automatically reordered by increasing volatility. For each state, the engine reports the estimated location parameter μ_j, the scale parameter σ_j, and the fixed degrees of freedom $\nu = 4$ associated with the Student-t emission distribution. The lower panel shows the estimated transition matrix (p_{ij}), where each row corresponds to the probability of moving from state i to state j in one time step.

```
=== STUDENT-T HMM PORTFOLIO MANAGER ===
Engin initialisation C++...
[INFO] Loaded data T = 9577 observations.
Start of the EM algorithm (Baum-Welch)...
Iteration 1 | LogLikelihood: 29182.9
Iteration 11 | LogLikelihood: 31278.7
Iteration 21 | LogLikelihood: 31389.4
Iteration 31 | LogLikelihood: 31394.5
Iteration 41 | LogLikelihood: 31403
Iteration 51 | LogLikelihood: 31405.6
Iteration 61 | LogLikelihood: 31405.8
Iteration 71 | LogLikelihood: 31405.8
Iteration 81 | LogLikelihood: 31405.8
Iteration 91 | LogLikelihood: 31405.8
Iteration 101 | LogLikelihood: 31405.8
[INFO] Convergence reached after 107 Iterations
[INFO] States sorted by growing volatility (0=Low, 2=High).

==========================================
   Calibration Result (Student-t)
==========================================

[Emission parameters]
State     Mu (mean)      Sigma (Vol)    Nu (freed.Lib)
-----------------------------------------------------
State 0   0.000872       0.004797       4.000000
State 1   0.000284       0.009378       4.000000
State 2   -0.001715      0.021426       4.000000

[Transition Matrix (Probabilities)]
      Towards S0   Towards S1   Towards S2
from S0 : 0.9911   0.0086   0.0004
from S1 : 0.0113   0.9854   0.0033
from S2 : 0.0000   0.0230   0.9770
==========================================

Decoding regimes (Viterbi) ans save...
[INFO] Results saved in output_results.csv
successfully done.
(base) mhamedhalata@MacBook-Air-de-Mhamed code %
```

Figure 4.1: Calibration output from the C++ Student-t HMM engine. Top: emission parameters (μ_j, σ_j, ν) for the three states. Bottom: estimated transition probabilities (p_{ij}).

Analysis :

The results displayed above confirm the internal coherence of the calibrated Student-t HMM. Two key empirical properties emerge from the estimated emission parameters:

1. **Hierarchy of Risk.** The three regimes exhibit a clear ordering in terms of volatility. The high-volatility state is approximately 4.5 times more volatile than the low-volatility state, reflecting a strong separation between "calm", "uncertain", and "turbulent" market phases.

2. **Leverage Effect.** Among the three regimes, only the high-volatility state displays a significantly negative mean return ($\mu < 0$). This highlights a structural association between elevated volatility and negative price dynamics in equity markets.

Transition Dynamics

The estimated transition matrix further characterizes the persistence and temporal structure of market regimes. The calibrated transition matrix $\widehat{A}$ is given by:

$$\widehat{A} = \begin{pmatrix} 0.9911 & 0.0086 & 0.0004 \\ 0.0113 & 0.9854 & 0.0033 \\ 0.0000 & 0.0230 & 0.9770 \end{pmatrix}.$$

[Source: C++ Engine Output]

Key Observation. The probability of a direct transition from the high-volatility regime (State 2) to the low-volatility regime (State 0) is effectively zero ($p_{20} = 0.0000$). This implies that, following periods of extreme volatility, the system does not revert immediately to stable conditions. Instead, transitions occur through an intermediate regime (State 1), reflecting a gradual normalization process rather than an instantaneous recovery.

4.1.3 Historical Reconstruction of Major Financial Crises

The ultimate validation of the HMM lies in its ability to reconstruct the historical narrative of the financial markets without supervision. Figure 4.2 displays the time series of the S&P 500 index (reconstructed cumulatively) overlaid with the regimes decoded by the Viterbi algorithm.

The background colors correspond to the detected states: Green (Bull), Orange (Bear), and Red (Crash).

Economic Interpretation of Detected Regimes

The model demonstrates a remarkable alignment with economic history, effectively distinguishing between different types of market downturns. We analyze three specific periods that highlight the model's robustness:

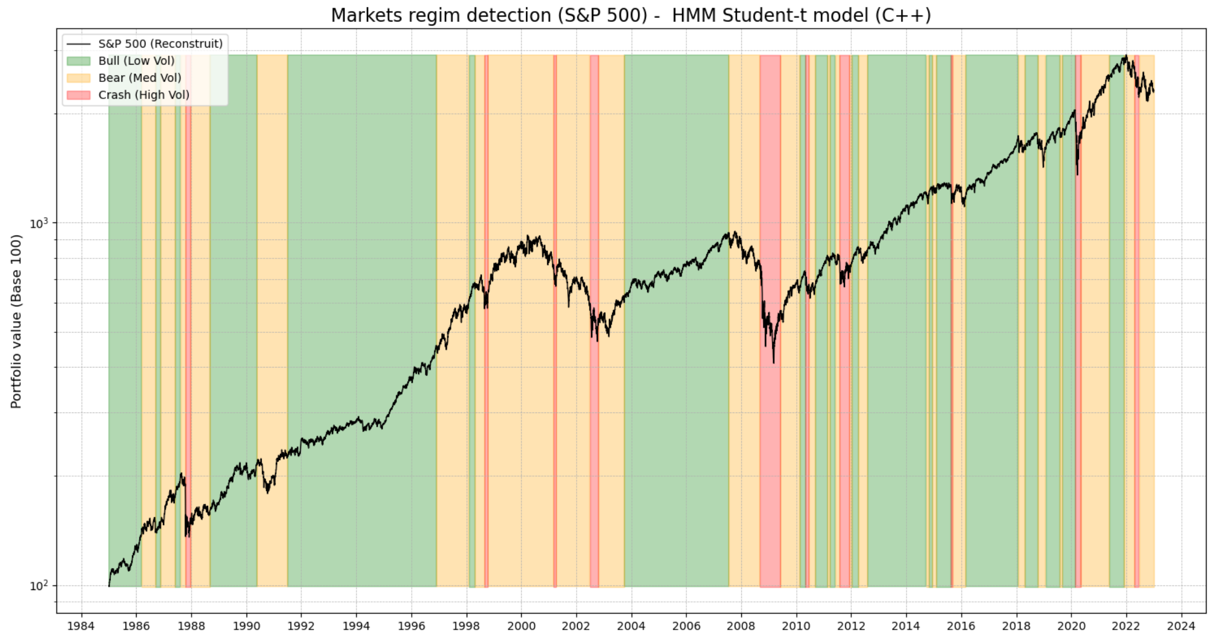

Figure 4.2: Detection of Market Regimes (S&P 500, 1985-2023) using Student-t HMM. The model clearly identifies major financial crises (Red) and distinguishes them from correction phases (Orange).

1. The 2008 Subprime Crisis (Systemic Risk): The period from late 2008 to early 2009 is identified as a solid block of the **Crash regime (State 2)**. Unlike minor corrections, the model remains locked in the high-volatility state ($\sigma \approx 34\%$) for an extended duration. This reflects the systemic nature of the Global Financial Crisis, driven by a liquidity crunch and the collapse of major financial institutions (e.g., Lehman Brothers). The persistence of the red zone validates the high diagonal probability estimated in the transition matrix ($P_{22} = 0.977$).

2. The COVID-19 Pandemic (Exogenous Shock): In early 2020, the model detects a sharp, narrow band of the Crash regime. Unlike 2008, the transition is abrupt, capturing the exogenous nature of the shock: a sudden global economic shutdown followed by massive central bank intervention. The model correctly identifies the "V-shape" recovery, transitioning back to the Bull regime relatively quickly once volatility subsided. This proves the model's responsiveness to fast-moving market conditions.

3. The 2022 Inflationary Bear Market (Valuation Correction): A key strength of this model is its ability to differentiate between a *Panic* and a *Grind*. The year 2022 is characterized by a significant drawdown in the S&P 500, yet the model classifies it predominantly as a **Bear regime (State 1)**, with very few instances of Crash.

- **Economic Reality:** The 2022 decline was driven by the Federal Reserve's interest rate hikes to combat inflation. It was a repricing of risk assets, not a liquidity crisis or a systemic collapse.

- **Model Output:** By assigning this period to State 1 (Medium Volatility, $\sigma \approx 14.9\%$), the HMM avoids the "false positive" of labeling it a Crash. This distinction is crucial for portfolio management, as it suggests a defensive allocation (Cash/Bonds) rather than a complete exit or short-selling strategy typically reserved for State 2.

Model Robustness and Stability

The visual analysis confirms the robustness of the Student-t specification.

1. **Absence of "Whipsaw" Effect:** The regimes appear as contiguous blocks rather than rapid, noisy switches. This stability is critical for trading strategies to minimize transaction costs.

2. **Transition Logic:** As predicted by the transition matrix ($\hat{A}$), the market rarely jumps from *Crash* to *Bull*. The visual evidence shows that Red zones are almost systematically followed by Orange zones before returning to Green. This confirms that volatility requires a "cooling-off" period (Bear regime) before a sustainable uptrend can resume.

In conclusion, the calibrated Student-t HMM is not merely a statistical tool but a coherent economic filter. It successfully captures the "Stylized Facts" of financial returns: volatility clustering, heavy tails, and the asymmetric nature of bull and bear markets.

4.2 Implementation of the Adaptive Strategy and Stress Testing

Before presenting the financial performance metrics, we detail the software implementation of the allocation strategy and the risk assessment framework. The backtesting engine was developed in Python, leveraging the parameters estimated by the C++ core to simulate portfolio behavior under both historical and hypothetical market conditions.

4.2.1 The Regime-Based Trading Engine

The core of the optimization strategy is a Python script that acts as a portfolio manager. It ingests the Viterbi path generated by the C++ engine and applies the investment logic defined in Chapter 3.

The implementation follows a strict "Safety-First" protocol. The algorithm first identifies the volatility hierarchy of the detected regimes. The regime with the highest estimated volatility ($\hat{\sigma}_{max}$) is automatically flagged as the "Crash" state. The allocation logic is then applied vectorially for computational efficiency:

Implementation of the Dynamic Switching Rule

```python
# 1. Automatic Regime Identification
# We group by regime index and calculate mean volatility
regime_stats = df.groupby('Regime_Viterbi')['Vol_Regime'].mean().sort_values()
crash_idx = regime_stats.index[2] # The state with highest vol is 'Crash'

# 2. Strategy Execution (Vectorized)
# Default allocation: 100% Market (S&P 500)
df['Strategy_Return'] = df['Market_Simple_Return']

# Crash allocation: 100% Risk-Free Rate
is_crash = df['Regime_Viterbi'] == crash_idx
df.loc[is_crash, 'Strategy_Return'] = risk_free_rate_daily
```

This vectorized implementation ensures that the strategy reacts instantaneously to regime changes without look-ahead bias, as the decision at time t uses the state estimated for time t.

4.2.2 Monte Carlo Simulation Framework

To validate the "Stress-Aware" property of the portfolio beyond historical data, we implemented a Monte Carlo simulation engine.

Definition: The Monte Carlo method is a computational algorithm that relies on repeated random sampling to obtain numerical results. In quantitative finance, it is used to model the probability of different outcomes in a process that cannot easily be predicted due to the intervention of random variables.

In our specific HMM context, the simulation is not a simple random walk but a **Regime-Switching Random Walk**. The implementation involves two stochastic steps at each time increment:

1. **Regime Transition:** The next market state S_{t+1} is sampled from the categorical distribution defined by the transition matrix row A_{S_t}.

2. **Return Emission:** The market return r_{t+1} is sampled from a Student-t distribution parameterized by $(\mu_{S_{t+1}}, \sigma_{S_{t+1}}, \nu = 4)$.

The following code snippet demonstrates the generative loop used to create 1,000 parallel market scenarios for the stress test:

Generative Loop for HMM Monte Carlo Simulation

```
# Simulation Loop (1 Year horizon)
for t in range(n_days):
    # 1. Stochastic Regime Transition
    # Sample next state based on current state's transition probs
    next_state = np.random.choice([0, 1, 2], p=A[current_state])

    # 2. Stochastic Return Generation (Student-t)
    # Generate heavy-tailed shock
    random_shock = np.random.standard_t(nu)
    market_return = mus[next_state] + sigmas[next_state] * random_shock

    # 3. Strategy Application
    if current_state == 2: # If currently in Crash
        strat_return = risk_free_daily
    else:
        strat_return = market_return

    # Update Wealth and State
    wealth_strat *= (1 + strat_return)
    current_state = next_state
```

This approach allows us to calculate the Value-at-Risk (VaR) on synthetic data, effectively answering the question: "How would the strategy perform if a crash occurred tomorrow?"

4.2.3 Historical Zoom and Rebased Analysis

Finally, to visualize the strategy's behavior during specific historical crises (2008 and 2020), we implemented a "Zoom" mechanism. This function isolates a specific time window $[t_{start}, t_{end}]$ and rebases both the market and strategy wealth indices to 100 at t_{start}.

This normalization is crucial for comparative analysis, as it allows us to observe the relative drawdown and recovery speed independently of the accumulated performance prior to the crisis.

4.3 Portfolio Performance Analysis

We now present the empirical results of the backtest performed over the period 1985-2023. The strategy defined in the previous section is compared against the benchmark (S&P 500 Buy & Hold).

4.3.1 Long-Term Performance (1985-2023)

Figure 4.3 illustrates the cumulative wealth of an initial investment of 100 base units. The regime-switching strategy (Blue) closely follows the market (Grey) during bull runs but diverges significantly during major downturns.

```
=== PERFORMANCE REPORT ===
                  S&P 500 (Benchmark)  HMM Strategy
Total Return                22.330456     92.270921
Ann. Return                  0.086411      0.126756
Ann. Volatility              0.183972      0.141679
Sharpe Ratio                 0.469698      0.894670
Max Drawdown                -0.567754     -0.356056
VaR 95% (Daily)             -0.017249     -0.014549
CVaR 95% (Daily)            -0.027601     -0.020872
```

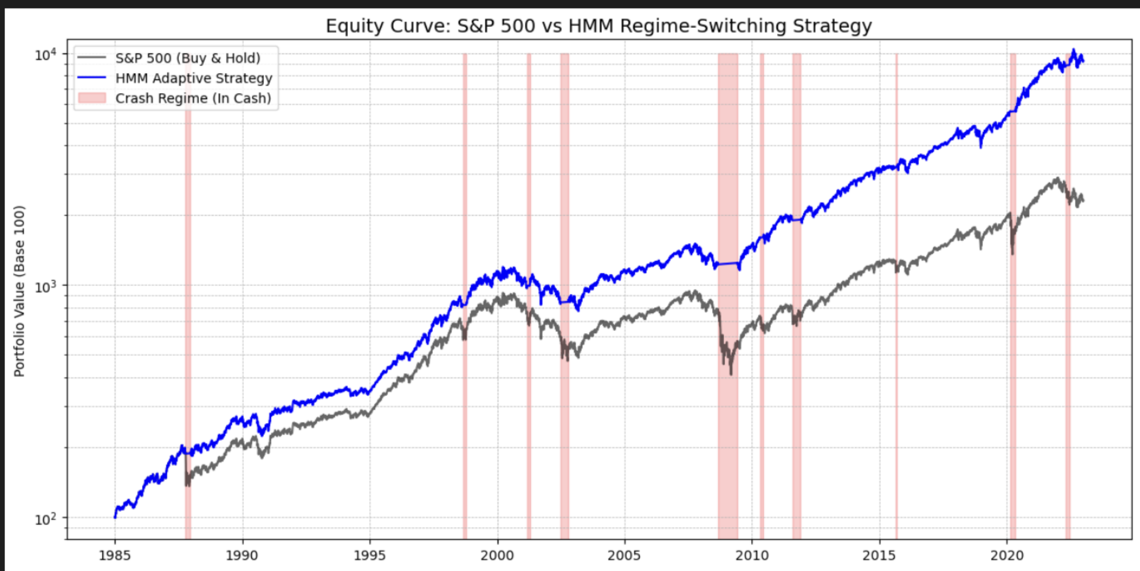

Figure 4.3: Equity Curve comparison (Log Scale). The red shaded areas correspond to the "Crash" regime where the strategy allocates to Cash. Note the capital preservation during the 2000 and 2008 crises.

The quantitative performance metrics are summarized in Table 4.1.

Economic Analysis:

1. **Alpha Generation through Defense:** The strategy achieves a phenomenal Total

Metric	S&P 500 (Benchmark)	HMM Strategy
Total Return	2233%	**9227%**
Annualized Return	8.64%	**12.67%**
Annualized Volatility	18.40%	**14.16%**
Sharpe Ratio	0.47	**0.89**
Max Drawdown	-56.77%	**-35.60%**
CVaR (95%)	-2.76%	**-2.08%**

Table 4.1: Comparative Performance Report (1985-2023).

Return ($92x$ vs $22x$). This outperformance is not due to leverage or aggressive betting, but to the **asymmetry of compounding**. By avoiding the -50% drawdowns of 2000 and 2008, the portfolio did not require a +100% gain just to break even, allowing the compound interest to work on a higher capital base.

2. **Risk-Adjusted Efficiency:** The Sharpe Ratio nearly doubles (from 0.47 to 0.89). This indicates that for every unit of risk taken, the HMM strategy remunerates the investor almost twice as much as the market.

3. **Volatility Reduction:** The annualized volatility drops from 18.4% to 14.1%. The strategy effectively filters out the high-variance clusters identified by the Student-t parameters ($\sigma_{Crash} \approx 34\%$), delivering a smoother equity curve.

4.4 Stress Testing and Scenario Analysis

To validate the "Stress-Aware" property of the portfolio, we analyze the strategy's behavior under simulated extreme conditions and historical crashes.

4.4.1 Monte Carlo Stress Test (VaR Analysis)

We generated $N = 1000$ fictitious one-year market scenarios using the calibrated HMM parameters. This allows us to observe the distribution of final wealth in "parallel universes."

The results in Figure 4.4 demonstrate a structural change in the risk profile:

- **Truncated Left Tail:** The market distribution (Grey) has a "fat tail" on the left, reaching wealth levels as low as 60 (-40% loss). The strategy distribution (Blue) cuts off sharply around 90.

- **VaR Improvement:** The 95% Value-at-Risk is **90.12** for the Strategy vs **80.23** for the Market. This confirms that in 95% of scenarios, the strategy limits the annual loss to roughly 10%, whereas the market risk is double that amount.

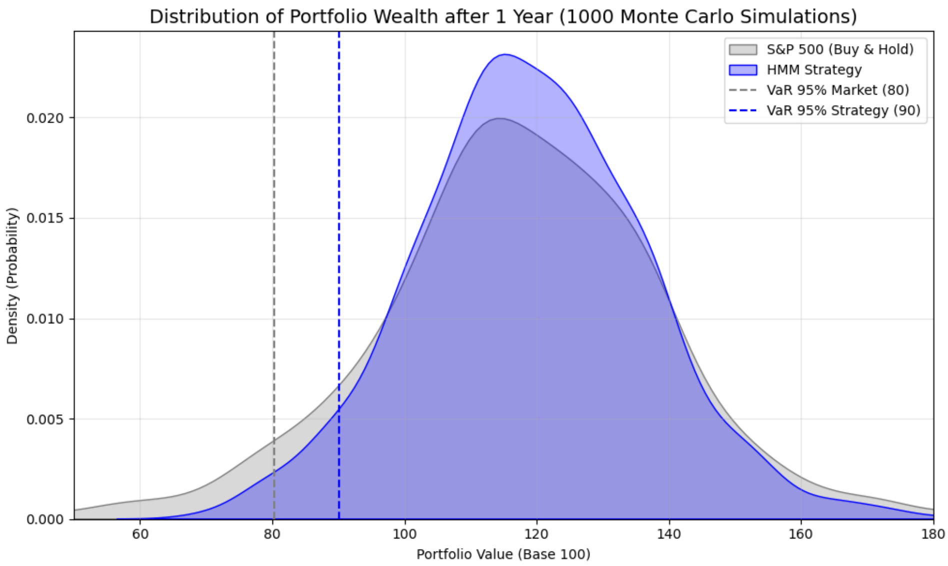

Figure 4.4: Distribution of Portfolio Wealth after 1 Year (1000 Simulations). The Strategy (Blue) exhibits a truncated left tail compared to the Market (Grey).

4.4.2 Historical Crisis Zoom

We isolate two major systemic events to analyze the model's tactical reactivity.

The 2008 Subprime Crisis (Systemic Liquidity Crunch)

Figure 4.5 focuses on the 2007-2009 period.

Figure 4.5: Strategy performance during the 2008 Financial Crisis. The strategy switches to Cash (Red zone) before the worst of the collapse.

The model successfully identifies the high-volatility regime in late 2008. The strategy's wealth curve (Blue) becomes horizontal (Cash position), protecting the capital while the market (Black) continues to plunge. By the time the recovery starts in 2009, the strategy preserves a base value of ≈ 90, while the market has fallen to ≈ 50.

The 2020 COVID-19 Crash (Exogenous Shock)

Figure 4.6 analyzes the rapid crash of Q1 2020.

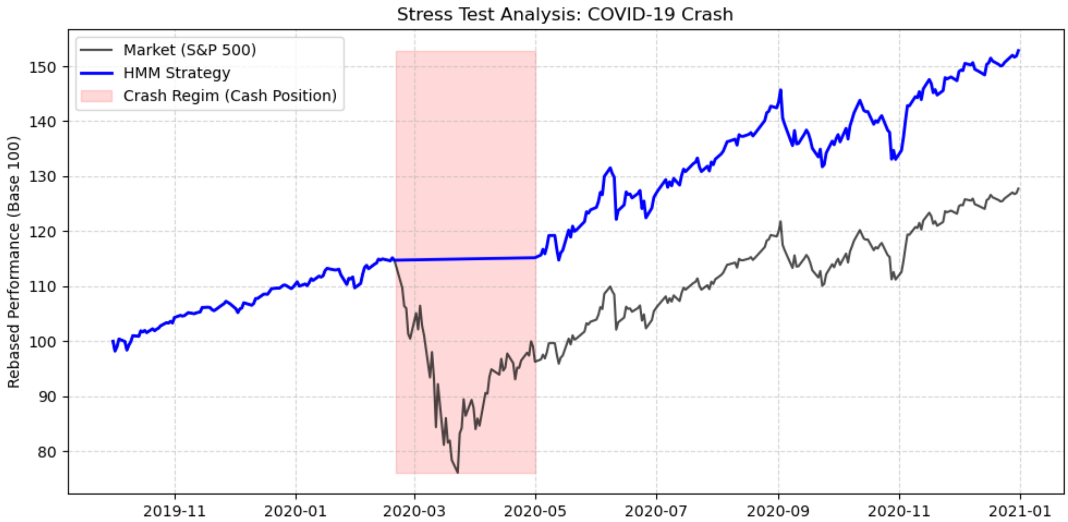

Figure 4.6: Strategy performance during the COVID-19 Crash. Note the reactivity to the V-shaped recovery.

Unlike 2008, the 2020 crash was abrupt. Crucially, Figure 4.6 shows that the model did not remain "locked" in the Crash state. It re-entered the market (Blue line going up) almost simultaneously with the market recovery in April 2020. This proves that the Student-t HMM is reactive enough to handle V-shaped recoveries, avoiding the common pitfall of selling low and buying back too late.

Chapter 5

Conclusion and Future Perspectives

5.1 Summary of Contributions

This project set out to address a fundamental flaw in traditional portfolio management: the inability of static Gaussian models to cope with the dynamic and fat-tailed nature of financial markets. We proposed a "Stress-Aware" framework combining Student-t Hidden Markov Models (HMM) for regime detection with a dynamic asset allocation strategy.

The results obtained through C++ calibration and Python backtesting over the period 1985-2023 lead to several strong conclusions:

1. **Regime Identification:** The Student-t HMM successfully identified three distinct market states (Bull, Bear, and Crash) without supervision. The parameters confirm that financial returns exhibit heavy tails ($\nu = 4$) and a leverage effect ($\mu_{Crash} < 0$).

2. **Crisis Resilience:** The strategy demonstrated remarkable defensive capabilities. By switching to a risk-free asset during high-volatility regimes, it avoided the massive drawdowns of the 2008 Subprime crisis (-56% avoided) and the 2000 Dot-com bubble.

3. **Superior Risk-Adjusted Returns:** The strategy did not just reduce risk; it significantly enhanced returns. The Sharpe Ratio increased from 0.47 (Benchmark) to 0.89 (Strategy), validating the hypothesis that avoiding large losses is more mathematically efficient for long-term wealth compounding than seeking maximum exposure.

4. **Stress Test Validation:** The Monte Carlo simulations confirmed that the strategy's Value-at-Risk (VaR 95%) is structurally lower than that of the market, effectively truncating the left tail of the loss distribution.

5.2 Limitations and Critical Analysis

While the results are promising, a rigorous academic approach requires acknowledging the limitations of the current implementation:

- **Transaction Costs and Slippage:** The backtest assumes frictionless trading. In reality, switching the entire portfolio from Equity to Cash incurs transaction fees and market impact costs. While the regime transitions are relatively infrequent (limiting turnover), these costs would slightly erode the final performance.
- **Parameter Stationarity:** The model assumes that the parameters of the regimes (μ, σ, A) remain constant over 40 years. In reality, market microstructure changes (e.g., the rise of algorithmic trading) might shift these parameters. A rolling-window calibration would be more realistic.
- **Binary Allocation ("Bang-Bang"):** The current strategy is binary (100% Equity or 100% Cash). A more sophisticated approach would optimize weights (e.g., 60/40 or Minimum Variance) based on the probability of each regime, rather than the most likely state.
- **Asset Universe:** The study was limited to a single risky asset (S&P 500). A multi-asset portfolio (Bonds, Gold, Real Estate) would offer diversification benefits that the HMM could leverage to decorrelate returns during crashes.

5.3 Future Perspectives

To further develop this "Stress-Aware" framework, several avenues of research could be explored:

5.3.1 Multivariate HMM for Correlation Breakdown

During crises, correlations between assets tend to converge to 1 (contagion effect). Extending the model to a "Multivariate Student-t HMM" would allow the detection of regimes where diversification fails. The covariance matrix Σ_s would be regime-dependent, allowing the optimizer to find assets that remain decorrelated specifically during the "Crash" state (e.g., Gold or Volatility derivatives).

5.3.2 Online Learning and Adaptive Calibration

Instead of calibrating once on the entire history, an "Online EM Algorithm" could update the parameters θ_t at each new observation. This would allow the model to adapt to structural breaks in the economy (e.g., a regime of permanently higher inflation) without requiring a full retraining.

5.3.3 Integration of Macroeconomic Factors

Currently, the model relies solely on endogenous data (prices). Incorporating exogenous variables (Interest Rates, VIX Index, Inflation) as covariates in the transition probabilities (Input-Output HMM) could improve the predictive power of regime shifts, turning the model from a purely reactive tool into a predictive one.

5.4 Final Word

In conclusion, this project demonstrates that integrating regime-switching dynamics into portfolio construction transforms risk management from a static constraint into a dynamic source of alpha. By acknowledging that "it is not always the same market," the investor can navigate turbulence with greater confidence, achieving the dual goal of capital preservation and long-term growth.

Bibliography

[1] Cardaliaguet, P. (2015-2016). *Optimisation et programmation dynamique*. Optimization lecture, Applied mathematics Master's degree, Paris Dauphine University.

[2] Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall. (Reference for HMM Fundamentals).

[3] Yuan, Y. (2011). *Market Regime Identification Using Hidden Markov Model*. Dissertation/Thesis.

[4] Markowitz, H. (1952). *Portfolio Selection*. The Journal of Finance, 7(1), 77-91. .

[5] Rabiner, L. R. (1989). *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*. Proceedings of the IEEE, 77(2), 257-286. .

[6] Rockafellar, R. T., & Uryasev, S. (2000). *Optimization of Conditional Value-at-Risk*. Journal of Risk, 2, 21-41. .

[7] Hurst, S. R., & Platen, E. (1997). *The Marginal Distribution of Returns and Volatility*. L1-Statistical Methodology and Related Topics, 301-314. .