SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis … in Informatics

Portfolio Optimization with Gaussian Process Regression

Xiyue ZHANG

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis … in Informatics

Portfolio Optimization with Gaussian Process Regression

Titel der Abschlussarbeit

Author:	Xiyue ZHANG
Supervisor:	Prof. Dr. Hans-Jochim Bungartz
Advisor:	Kislaya Ravi
Submission Date:	December 3rd

I confirm that this master's thesis ... is my own work and I have documented all sources and material used.

Munich, December 3rd Xiyue ZHANG

Acknowledgments

Abstract

Contents

1 Introduction

1.1 Background and Motivation

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` ⇒ Technical University of Munich (TUM), TUM

For more details, see the documentation of the `acronym` package[1].

1.1.1 Evolution of portfolio optimization techniques

The field of portfolio optimization has undergone significant transformations since its inception in the mid-20th century, evolving from simple diversification principles to sophisticated mathematical models incorporating machine learning and artificial intelligence. This evolution reflects both the advancing computational capabilities and our deepening understanding of financial markets' complexity.

Classical Foundations (1950s-1960s) The modern era of portfolio optimization began with [Mar52]'s seminal paper "Portfolio Selection," which laid the groundwork for Modern Portfolio Theory (MPT). Markowitz introduced several revolutionary concepts:

- The mathematical formalization of diversification
- The mean-variance optimization framework
- The efficient frontier of optimal portfolios
- The fundamental relationship between risk and return

This work established the first rigorous mathematical framework for portfolio selection, earning Markowitz the Nobel Prize in Economics and fundamentally changing how practitioners approached portfolio management.

[1] `https://ctan.org/pkg/acronym`

Early Developments (1960s-1980s) Building upon Markowitz's foundation, several crucial developments emerged:

1. **Capital Asset Pricing Model (CAPM)**
   - Developed by [Sha64], [Lin65], and [Mos66]
   - Introduced systematic and unsystematic risk concepts
   - Established the theoretical framework for asset pricing
   - Created the foundation for risk-adjusted performance measures
2. **Single-Index Models**
   - Simplified the estimation of covariance matrices
   - Reduced computational complexity
   - Introduced market beta as a risk measure
   - Enhanced practical applicability of portfolio optimization

Advanced Optimization Era (1980s-2000s) The advent of increased computational power led to more sophisticated approaches:

1. **Black-Litterman Model (1990s)**
   - Incorporated investor views into the optimization process
   - Addressed estimation error issues in mean-variance optimization
   - Introduced Bayesian methods to portfolio optimization
   - Provided more stable and intuitive portfolio allocations
2. **Risk-Based Portfolio Optimization**
   - Development of risk parity strategies [Qia05]
   - Introduction of alternative risk measures (VaR, CVaR)
   - Focus on downside risk management
   - Enhanced robustness to estimation errors

Modern Approaches (2000s-Present) Recent developments have focused on addressing classical methods' limitations:

1. **Robust Optimization**
   - Accounts for parameter uncertainty

- Provides protection against worst-case scenarios
- Incorporates estimation error in the optimization process
- Yields more stable portfolio allocations

2. **Dynamic Portfolio Optimization**
   - Considers time-varying investment opportunities
   - Incorporates transaction costs
   - Accounts for changing market conditions
   - Enables adaptive portfolio management
3. **Machine Learning Integration**
   - Neural networks for return prediction [HPW17]
   - Support vector machines for risk assessment
   - Reinforcement learning for portfolio management
   - Gaussian Processes (GP) for uncertainty quantification

Current Challenges and Future Directions Modern portfolio optimization faces several challenges:

1. **Data Quality and Quantity**
   - High-dimensional data processing
   - Non-stationary market conditions
   - Alternative data integration
   - Real-time data processing requirements
2. **Model Complexity**
   - Balance between model sophistication and robustness
   - Computational efficiency
   - Interpretability of results
   - Parameter stability
3. **Implementation Challenges**
   - Transaction costs
   - Market impact

- Regulatory constraints
- Operational considerations

The field continues to evolve with emerging technologies and methodologies:

1. **Artificial Intelligence Applications**
    - Deep learning for market prediction
    - Natural language processing for sentiment analysis
    - Alternative data processing
    - Automated portfolio rebalancing
2. **Advanced Risk Management**
    - Tail risk hedging
    - Dynamic risk allocation
    - Scenario analysis
    - Real-time risk monitoring
3. **Sustainability Integration**
    - ESG factors in optimization
    - Climate risk consideration
    - Impact investing metrics
    - Sustainable portfolio construction

This evolution of portfolio optimization techniques sets the stage for our research, which builds upon these foundations while incorporating modern machine learning approaches, specifically Gaussian Process Regression, to address current challenges in portfolio optimization.

1.1.2 Role of machine learning in financial forecasting

Machine Learning (ML) has gained significant traction in financial markets due to its ability to analyze vast amounts of data and identify complex patterns that traditional models may overlook. In particular, Gaussian Processes Regression (GPR) has emerged as a powerful tool for time-series forecasting, offering a flexible framework for capturing non-linear relationships and uncertainty in predictions.

1.1.3 Challenges in time-series forecasting and traditional portfolio optimization methods

Despite the advancements in ML techniques, predicting asset returns remains a challenging task due to the inherent volatility and non-stationarity of financial markets. Moreover, traditional portfolio optimization methods, while theoretically elegant, often face significant practical challenges in implementation. These challenges primarily stem from the difficulty in accurately estimating input parameters and the inherent uncertainty in financial time-series forecasting. This section examines these challenges and introduces how GPR provides a novel approach to addressing them.

Parameter Estimation Challenges in Modern Portfolio Theory Modern Portfolio Theory (MPT), despite its theoretical elegance, relies heavily on accurate estimation of key parameters: The practical implementation of MPT is fundamentally constrained by the difficulty in estimating volatility, arguably the most critical parameter in portfolio optimization. Traditional approaches to volatility estimation rely heavily on historical data, assuming that past patterns will persist into the future. However, financial markets are dynamic systems characterized by regime changes, varying volatility clusters, and complex interdependencies that make such assumptions problematic. Historical volatility estimates are inherently backward-looking and highly sensitive to the chosen estimation window, leading to potentially misleading inputs for portfolio optimization.

Moreover, the challenge extends beyond simple volatility estimation. The correlation structure between assets, another crucial input for MPT, exhibits time-varying properties that are difficult to capture using conventional methods. During periods of market stress, these correlations often shift dramatically, invalidating historical estimates precisely when accurate risk assessment is most critical. The dimensionality of this problem grows quadratically with the number of assets, making it particularly challenging for large, diversified portfolios.

- **Volatility Estimation**
  - Historical volatility may not reflect future risk
  - Sample estimates are sensitive to the chosen time window
  - Regime changes can invalidate historical estimates
  - Heteroskedasticity in financial time series complicates estimation
- **Expected Returns**
  - Notoriously difficult to estimate accurately

    - High sensitivity to estimation errors
    - Time-varying nature of expected returns
    - Impact of market regimes on return distributions

- **Correlation Structure**
    - Dynamic nature of asset correlations
    - Curse of dimensionality in large portfolios
    - Instability during market stress periods
    - Computational challenges in high dimensions

Limitations of Traditional Forecasting Methods Traditional forecasting approaches in finance have predominantly relied on methods that provide point estimates, failing to capture the inherent uncertainty in financial predictions. These methods often make strong assumptions about the underlying data distribution and struggle to adapt to the non-linear, non-stationary nature of financial time series. AutoRegressive Integrated Moving Average (ARIMA) models, exponential smoothing, and other classical approaches, while mathematically tractable, often fall short in capturing the complex dynamics of financial markets.

A fundamental limitation of these traditional approaches is their rigidity in handling uncertainty. Point forecasts, even when accompanied by confidence intervals based on historical volatility, fail to capture the dynamic nature of prediction uncertainty. This limitation becomes particularly problematic in portfolio optimization, where understanding the reliability of forecasts is as important as the forecasts themselves.

Conventional approaches to financial time-series forecasting exhibit several limitations:

1. **Point Estimates**
    - Traditional methods often provide single-point forecasts
    - Lack of uncertainty quantification
    - Limited ability to capture prediction confidence
    - Insufficient information for risk management

2. **Model Rigidity**
    - Assumption of specific probability distributions
    - Difficulty in capturing non-linear relationships
    - Limited adaptation to changing market conditions

- Oversimplification of complex market dynamics

3. **Data Requirements**
   - Need for large historical datasets
   - Sensitivity to outliers and noise
   - Challenge of incorporating multiple data sources
   - Difficulty in handling missing data

Advantages of Gaussian Process Regression Our research proposes GPR as a solution to these challenges, offering several key advantages:

1. **Probabilistic Framework**
   - Natural uncertainty quantification
   - Automatic volatility estimation through posterior variance
   - Capture of prediction confidence intervals
   - Robust handling of noise in financial data
2. **Flexible Modeling**
   - Non-parametric approach avoiding distributional assumptions
   - Ability to capture complex non-linear relationships
   - Automatic complexity adjustment through kernel selection
   - Incorporation of prior knowledge through kernel design
3. **Parameter Estimation**
   - Direct modeling of volatility through posterior variance
   - Joint estimation of returns and risk
   - Principled handling of uncertainty
   - Adaptive to changing market conditions

Addressing Traditional Limitations Our GPR-based approach specifically addresses the key limitations of MPT:

$$\sigma^2_{GPR}(x_*) = k(x_*, x_*) - k(x_*, X)[K(X, X) + \sigma_n^2 I]^{-1} k(X, x_*) \tag{1.1}$$

Where $\sigma^2_{GPR}(x_*)$ represents the posterior variance at prediction point x_*, providing a direct estimate of volatility that:

- Naturally accounts for uncertainty in predictions
- Adapts to local data density and quality
- Provides time-varying volatility estimates
- Incorporates both local and global market information

The GPR approach offers several fundamental advantages over traditional methods. Unlike historical volatility estimates that require arbitrary window selection, GPR's volatility estimates emerge naturally from the probabilistic learning process. The method adapts automatically to different market regimes through its kernel function, which can capture both long-term trends and short-term fluctuations in market behavior.

Furthermore, GPR's non-parametric nature frees it from restrictive assumptions about return distributions. The method can capture complex, non-linear patterns in the data while maintaining the ability to quantify uncertainty in its predictions. This combination of flexibility and uncertainty awareness makes it particularly well-suited for financial applications where both accuracy and risk assessment are crucial.

Implications for Portfolio Optimization The GPR framework transforms the traditional portfolio optimization problem by: The integration of GPR into portfolio optimization transforms the traditional MPT framework by providing more reliable and dynamic parameter estimates. By directly modeling the uncertainty in our predictions, we can make more informed portfolio allocation decisions that account for both expected returns and our confidence in those expectations. This approach naturally leads to more robust portfolios that adapt to changing market conditions while maintaining a principled approach to risk management.

The significance of this advancement cannot be overstated. By addressing one of the fundamental criticisms of MPT – the difficulty of accurately estimating volatility – our GPR-based approach bridges the gap between theoretical elegance and practical applicability. This enhancement makes MPT more reliable and useful for real-world portfolio management, where accurate risk assessment is crucial for maintaining stable, long-term investment performance.

1.2 Research Objectives

The primary objective of this study is to develop a predictive portfolio optimization framework that leverages Gaussian Process Regression (GPR) for time-series forecasting in financial markets. By integrating advanced predictive modeling with strategic asset allocation, the study aims to enhance investment performance through informed decision-making. The specific objectives are as follows:

1. **Develop and Validate GPR Models for Asset Return Prediction**
   - *Model Construction:* Build individual GPR models to forecast future returns of selected assets, including forex, gold, Bitcoin, and various stocks.
   - *Feature Engineering:* Utilize historical one-month returns and time as input features to capture both market dynamics and temporal patterns.
   - *Model Updating:* Implement an iterative training process where models are updated daily with new market data to ensure predictions remain current and adaptive.
2. **Integrate GPR Predictions into Portfolio Optimization Strategies**
   - *Expected Returns and Volatilities:* Extract predicted returns and associated volatilities from the GPR models for use in portfolio construction.
   - *Strategy Formulation:* Design multiple optimization strategies—Maximum Return, Minimum Volatility, Maximum Sharpe Ratio, and a Dynamic Strategy—based on the GPR outputs.
3. **Develop a Dynamic Portfolio Optimization Strategy**
   - *Probabilistic Assessment:* Calculate the probability distribution of next-day cumulative portfolio returns using the predicted normal distribution of asset returns.
   - *Threshold-Based Decision Making:* Establish a threshold probability to decide when to reallocate the portfolio for maximizing returns versus holding the current positions.
   - *Adaptive Allocation:* Enable the portfolio to adapt dynamically to changing market conditions by selectively applying the Maximum Return Strategy based on probabilistic forecasts.
4. **Evaluate and Compare the Performance of Optimization Strategies**
   - *Backtesting Framework:* Conduct backtesting over a historical period using real market data to assess the strategies' performance.
   - *Incorporation of Transaction Costs:* Include realistic transaction fees in the evaluation to account for the costs associated with portfolio rebalancing.
   - *Performance Metrics:* Measure total returns, portfolio volatility, Sharpe ratios, and transaction costs to provide a comprehensive performance analysis.

5. **Demonstrate the Effectiveness of the Dynamic Strategy**
    - *Performance Analysis:* Analyze the results to determine if the Dynamic Strategy achieves higher returns and lower transaction costs compared to traditional strategies.
    - *Risk Management:* Assess how the Dynamic Strategy balances the trade-off between pursuing higher returns and minimizing risks and costs.
    - *Statistical Significance:* Use statistical methods to verify the significance of the observed performance differences among the strategies.
6. **Contribute to the Field of Predictive Portfolio Optimization**
    - *Innovative Approach:* Present a novel integration of GPR-based forecasting with adaptive portfolio optimization strategies.
    - *Practical Implications:* Provide insights and recommendations for practitioners on implementing dynamic, data-driven approaches in portfolio management.
    - *Foundation for Future Research:* Establish a basis for further exploration into advanced predictive models and adaptive strategies in financial optimization.

By achieving these objectives, the study seeks to demonstrate that incorporating sophisticated predictive models like GPR into portfolio optimization can significantly enhance investment outcomes. The findings aim to contribute valuable knowledge to the field of quantitative finance, particularly in the areas of time-series forecasting and dynamic asset allocation.

1.2.1 Research Contributions

This study makes several significant contributions to the field of predictive portfolio optimization and quantitative finance. The key research contributions are outlined below:

1. **Novel Integration of Gaussian Process Regression with Dynamic Portfolio Optimization**

    This research presents a unique integration of GPR models with dynamic portfolio optimization strategies. By employing GPR for time-series forecasting, we capture complex, non-linear relationships in financial data, enhancing the accuracy of return predictions. The integration facilitates a more responsive and informed portfolio allocation process, adapting to market changes in real-time.

2. **Probabilistic Approach to Strategy Selection**

   We introduce a probabilistic framework for strategy selection within the portfolio optimization process. By calculating the probability distribution of future cumulative returns, the Dynamic Strategy makes informed decisions on whether to reallocate the portfolio based on a predefined threshold. This approach incorporates uncertainty and risk directly into the decision-making process, allowing for a more nuanced and adaptive investment strategy.

3. **Practical Implementation Considering Transaction Costs**

   The study emphasizes practical applicability by incorporating realistic transaction costs into the optimization and backtesting processes. By accounting for these costs, we provide a more accurate assessment of the strategies' net performance. This consideration is crucial for real-world portfolio management, where transaction fees can significantly impact returns, especially in high-frequency trading environments.

4. **Comparative Analysis of Different Optimization Strategies**

   We conduct a comprehensive comparative analysis of multiple portfolio optimization strategies, including Maximum Return, Minimum Volatility, Maximum Sharpe Ratio, and the proposed Dynamic Strategy. By evaluating these strategies under the same conditions and performance metrics, we provide valuable insights into their relative effectiveness. This analysis helps identify the strengths and limitations of each approach, guiding practitioners in selecting appropriate strategies based on their investment goals and risk tolerance.

These contributions collectively advance the understanding of how advanced predictive models and adaptive strategies can be effectively combined to enhance portfolio performance. The novel methodologies and findings offer practical benefits for portfolio managers and lay the groundwork for future research in predictive asset allocation.

Table 1.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

!TeX root = ../main.tex

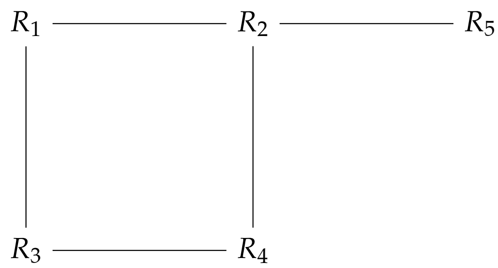

Figure 1.1: An example for a simple drawing.

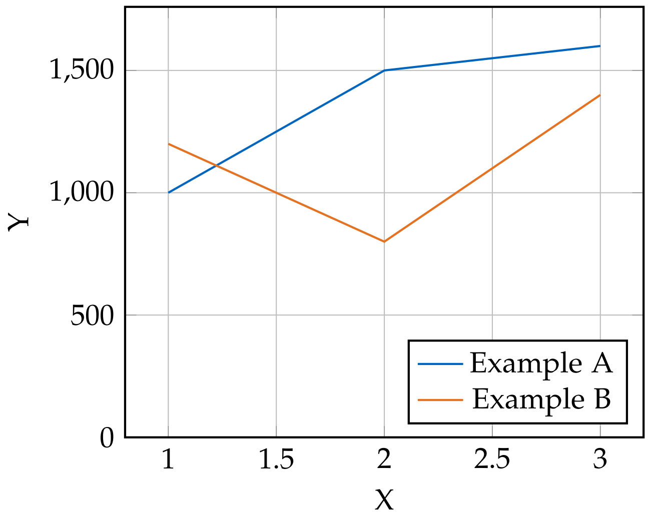

Figure 1.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 1.3: An example for a source code listing.

2 Previous Literature

2.1 Theoretical Framework

This chapter discusses the fundamental theories underpinning the study, focusing on portfolio optimization and GPR. Understanding these theories is essential for developing the predictive portfolio optimization framework proposed in this research.

2.1.1 Portfolio Optimization Theory

Portfolio optimization is a cornerstone of modern finance, aiming to allocate assets in a way that balances expected returns against risk. The foundational theory in this domain is the MPT, introduced by Harry Markowitz in 1952 [Mar52].

Modern Portfolio Theory (MPT)

MPT posits that investors can construct an optimal portfolio that offers the maximum expected return for a given level of risk or, equivalently, the minimum risk for a given level of expected return. The key assumptions of MPT are:

- Investors are rational and risk-averse, preferring higher returns and lower risk.
- Markets are efficient, and all investors have access to the same information.
- Asset returns are normally distributed and can be described by their mean (expected return) and variance (risk).

Expected Return and Risk The expected return of a portfolio, $E[R_p]$, is the weighted sum of the expected returns of the individual assets:

$$E[R_p] = \sum_{i=1}^{n} w_i E[R_i], \tag{2.1}$$

where w_i is the weight of asset i in the portfolio, $E[R_i]$ is the expected return of asset i, and n is the total number of assets.

The portfolio variance, σ_p^2, representing risk, is given by:

$$\sigma_p^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} w_i w_j \sigma_{ij}, \tag{2.2}$$

where σ_{ij} is the covariance between asset i and asset j. The standard deviation σ_p is the square root of the variance.

Efficient Frontier The set of optimal portfolios that offer the highest expected return for a given level of risk forms the *Efficient Frontier*. Portfolios on the efficient frontier are considered optimal, as no other portfolios offer higher returns for the same risk level.

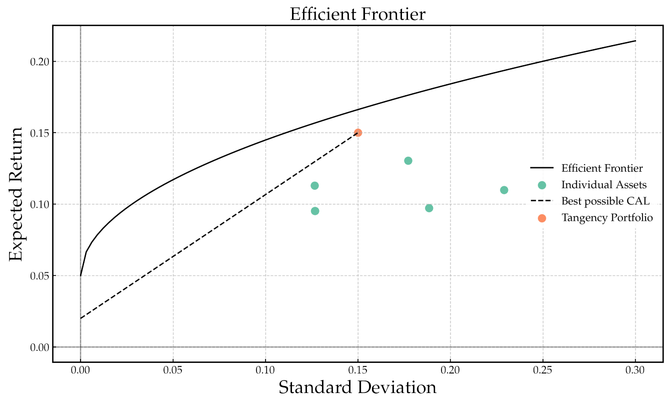

Figure 2.1: Efficient Frontier in Mean-Variance Space

Mean-Variance Optimization Mean-variance optimization involves solving for the portfolio weights that minimize the portfolio variance for a given expected return. The optimization problem can be formulated as:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \mathbf{w}^T \mathbf{\Sigma} \mathbf{w} \\ \text{subject to} \quad & \mathbf{w}^T \mathbf{E} = E_p, \\ & \sum_{i=1}^{n} w_i = 1, \\ & w_i \geq 0, \quad i = 1, \ldots, n, \end{aligned} \tag{2.3}$$

where:

- $\mathbf{w} \in \mathbb{R}^n$ is the vector of asset weights

- $\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is the covariance matrix of asset returns
- $\mathbf{E} \in \mathbb{R}^n$ is the vector of expected asset returns
- $E_p \in \mathbb{R}$ is the desired expected portfolio return

Risk-Return Trade-off and the Sharpe Ratio

Investors seek to maximize returns while minimizing risk. The *Sharpe Ratio*, introduced by William F. Sharpe [**sharpe1966mutual**], measures the risk-adjusted return of a portfolio:

$$S = \frac{E[R_p] - R_f}{\sigma_p}, \tag{2.4}$$

where R_f is the risk-free rate. A higher Sharpe Ratio indicates a more favorable risk-return trade-off.

Portfolio Optimization Strategies

Various strategies exist for portfolio optimization, each with different objectives and constraints:

- **Maximum Return Strategy**: Focuses on maximizing expected returns, often leading to higher risk.
- **Minimum Volatility Strategy**: Aims to minimize risk while achieving a minimum acceptable return.
- **Maximum Sharpe Ratio Strategy**: Seeks the optimal balance between return and risk by maximizing the Sharpe Ratio.
- **Dynamic Strategies**: Adjust portfolio allocations based on changing market conditions and predictive insights.

2.1.2 Gaussian Process Regression Theory and Applications

Gaussian Process Regression (GPR) is a non-parametric, Bayesian approach to regression that is particularly powerful for modeling complex, non-linear relationships. GPR provides not only predictions but also uncertainty estimates, which are valuable in risk-sensitive applications like finance.

Gaussian Processes

A *Gaussian Process* (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution **[rasmussen2006gaussian]**. A GP is fully specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right). \tag{2.5}$$

Mean Function The mean function $m(\mathbf{x})$ represents the expected value of the function at point $\mathbf{x}$:

$$m(\mathbf{x}) = E[f(\mathbf{x})]. \tag{2.6}$$

Covariance Function The covariance function $k(\mathbf{x}, \mathbf{x}')$ defines the covariance between function values at points $\mathbf{x}$ and $\mathbf{x}'$:

$$k(\mathbf{x}, \mathbf{x}') = E\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))\right]. \tag{2.7}$$

Common choices for the covariance function include the squared exponential kernel and the Matérn kernel.

Gaussian Process Regression for Time-Series Forecasting

GPR models the underlying function mapping inputs to outputs, capturing uncertainty in predictions. For time-series forecasting:

- **Inputs**: Historical data points, such as lagged returns and time indices.
- **Outputs**: Future values of the time series, such as asset returns.

Given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i$ are inputs and y_i are observations, the goal is to predict the output f_* at a new input $\mathbf{x}_*$.

Predictive Distribution The predictive distribution of f_* given the training data is Gaussian:

$$p(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}\left(f_*|\mu_*, \sigma_*^2\right), \tag{2.8}$$

where

$$\mu_* = k_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - k_*^\top(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}k_*. \tag{2.9}$$

Here, $k_* = [k(\mathbf{x}_*, \mathbf{x}_1), \ldots, k(\mathbf{x}_*, \mathbf{x}_N)]^\top$, $\mathbf{K}$ is the covariance matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, and σ_n^2 is the noise variance.

Covariance Functions (Kernels)

The choice of kernel function $k(\mathbf{x}, \mathbf{x}')$ is critical in GPR as it encodes assumptions about the function being learned.

- *Squared Exponential Kernel*:

$$k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2}||\mathbf{x} - \mathbf{x}'||^2\right), \tag{2.10}$$

  where σ_f^2 is the signal variance and ℓ is the length-scale parameter.

- *Matérn Kernel*:

$$k_{\text{Matérn}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||}{\ell}\right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}||\mathbf{x} - \mathbf{x}'||}{\ell}\right), \tag{2.11}$$

  where ν controls the smoothness, Γ is the gamma function, and K_ν is the modified Bessel function.

Gaussian Process Regression for Time-Series Forecasting

In GPR, given training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal is to predict the output y_* for a new input $\mathbf{x}_*$. The predictive distribution is Gaussian with mean and variance:

$$E[y_*] = \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y}, \tag{2.12}$$

$$\mathrm{Var}[y_*] = k_{**} - \mathbf{k}_*^T(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{k}_*, \tag{2.13}$$

where:

- $\mathbf{K}$ is the $n \times n$ covariance matrix evaluated at the training inputs.
- $\mathbf{k}_*$ is the covariance vector between the test point and the training inputs.
- k_{**} is the covariance between the test point and itself.
- σ_n^2 is the variance of the observation noise.
- $\mathbf{y}$ is the vector of training targets.

Advantages of GPR in Financial Modeling GPR offers several benefits for financial time-series forecasting:

- *Non-parametric Flexibility*: Does not assume a specific functional form, allowing for modeling of complex, non-linear relationships.
- *Uncertainty Quantification*: Provides probabilistic predictions with associated confidence intervals, which are crucial for risk management.
- *Bayesian Framework*: Naturally incorporates prior knowledge and updates beliefs with new data.

Challenges in Applying GPR to Finance

Despite its advantages, GPR faces challenges in financial applications:

- *Computational Complexity*: Involves inverting an $n \times n$ matrix, which can be computationally intensive for large datasets.
- *Non-Stationarity*: Financial time-series often exhibit non-stationary behavior, violating assumptions of stationarity in standard GPR.
- *Noise Characteristics*: Financial data can be noisy and volatile, affecting model performance.
- *Hyperparameter Tuning*: Selection of kernel functions and hyperparameters significantly impacts model performance and requires careful tuning.

2.1.3 Integrating Portfolio Optimization and GPR

Combining portfolio optimization with GPR-based forecasting aims to leverage accurate predictions of asset returns and associated uncertainties to make informed allocation decisions. The integration involves:

- Using GPR to predict expected returns and volatilities for assets.
- Incorporating these predictions into the optimization models to determine optimal portfolio weights.
- Adjusting for uncertainties by considering the confidence intervals provided by GPR in the optimization process.

Dynamic Portfolio Optimization Dynamic optimization involves updating the portfolio allocation as new information becomes available. By retraining the GPR models with new data and adjusting the portfolio accordingly, the strategy adapts to changing market conditions, potentially enhancing performance.

2.1.4 Conclusion

The theoretical foundation provided by portfolio optimization theory and Gaussian Process Regression is critical for developing the predictive portfolio optimization framework. Understanding the principles, advantages, and limitations of these theories allows for effective integration and application in financial modeling and asset allocation.

2.2 Risk measures and portfolio optimization

2.3 Machine Learning in Financial Markets

2.3.1 Overview of Machine Learning Applications in Finance

The financial markets generate vast amounts of data daily, encompassing stock prices, trading volumes, economic indicators, and news articles. Machine learning algorithms are uniquely positioned to process and analyze this data, uncovering patterns and insights that traditional statistical methods may overlook. Key applications of machine learning in finance include:

Time-Series Forecasting

Predicting future asset prices and market trends is a fundamental objective in finance. Machine learning models, such as neural networks, support vector machines, and ensemble methods, are employed to forecast time-series data by capturing non-linear relationships and complex temporal dependencies [**sezer2020financial**].

Algorithmic Trading

Machine learning algorithms facilitate the development of automated trading systems that execute trades based on predefined strategies and real-time data analysis. Techniques like reinforcement learning enable the optimization of trading strategies through continuous learning from market interactions [**nevmyvaka2006reinforcement**].

Risk Management

Accurate risk assessment is crucial for financial institutions. Machine learning models help in predicting credit risk, market risk, and operational risk by analyzing historical data and identifying factors that contribute to potential losses [**lessmann2015benchmarking**].

Portfolio Optimization

Machine learning enhances portfolio optimization by providing more accurate estimates of expected returns and covariances between assets. Advanced models can adapt to changing market conditions and incorporate a broader set of predictive features [HPW17].

Fraud Detection and Anomaly Detection

Detecting fraudulent activities and anomalies is vital for maintaining the integrity of financial systems. Machine learning algorithms, particularly unsupervised learning techniques, are used to identify unusual patterns in transaction data that may indicate fraud [**phua2010comprehensive**].

Sentiment Analysis and Natural Language Processing

Analyzing news articles, social media, and financial reports using natural language processing (NLP) helps investors gauge market sentiment and its potential impact on asset prices [**hagenau2013automated**].

2.3.2 Conclusion

Machine learning plays a pivotal role in modern financial analysis, offering sophisticated tools for modeling and decision-making. Gaussian Process Regression, with its probabilistic nature and flexibility, is particularly well-suited for financial applications that require modeling uncertainty and non-linear relationships. Understanding the theoretical foundations and practical considerations of GPR is essential for its effective integration into financial modeling and portfolio optimization.

2.4 Dynamic Portfolio Management

Dynamic portfolio management involves continuously adjusting asset allocations in response to changing market conditions, forecasts, and investment objectives. Unlike static strategies, dynamic approaches aim to optimize the portfolio over time by incorporating new information and adapting to market dynamics. This section reviews dynamic optimization strategies, discusses the impact of transaction costs on portfolio rebalancing, and examines existing approaches to strategy switching.

2.4.1 Review of Dynamic Optimization Strategies

Dynamic optimization strategies are designed to adapt portfolio allocations over time, taking into account the stochastic nature of asset returns and changing investment opportunities. Key concepts and methods in dynamic portfolio optimization include:

Dynamic Programming

Dynamic programming is a mathematical optimization approach that solves complex problems by breaking them down into simpler subproblems. In the context of portfolio optimization, dynamic programming can be used to determine the optimal investment policy over multiple periods [**bellman1957dynamic**].

Bellman's Principle of Optimality Bellman's principle states that an optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. This principle underpins dynamic programming methods in portfolio optimization.

Stochastic Control Theory

Stochastic control theory deals with decision-making in systems that evolve over time under uncertainty. In portfolio optimization, stochastic control models can determine optimal asset allocations by considering the stochastic processes governing asset returns and investor wealth [**merton1969lifetime**].

Merton's Portfolio Problem Robert C. Merton extended the continuous-time portfolio optimization framework by incorporating stochastic control techniques. Merton's model determines the optimal consumption and portfolio allocation strategies for an investor with a finite or infinite investment horizon, maximizing expected utility [**merton1971optimum**].

Reinforcement Learning

Reinforcement learning (RL) is a machine learning paradigm where agents learn optimal policies through interactions with the environment by maximizing cumulative rewards. In finance, RL can be applied to portfolio management by treating asset allocation as a sequential decision-making problem [**moody1998performance**].

Applications in Portfolio Management RL algorithms, such as Q-learning and policy gradients, have been used to develop adaptive trading strategies that learn from market data and adjust allocations dynamically [**almahdi2019adaptive**].

Scenario Analysis and Model Predictive Control

Scenario analysis involves evaluating portfolio performance under different hypothetical future states of the world. Model Predictive Control (MPC) uses forecasts to optimize current decisions while considering future trajectories, adjusting strategies as new information becomes available [**primbs2009dynamic**].

Advantages of MPC MPC allows for the incorporation of predictive models (e.g., GPR forecasts) into the optimization process, enabling the portfolio to adapt dynamically to anticipated market changes.

2.4.2 Transaction Costs and Portfolio Rebalancing

Transaction costs play a significant role in dynamic portfolio management, as frequent rebalancing can erode returns. Understanding and modeling transaction costs are essential for effective portfolio optimization.

Types of Transaction Costs

Transaction costs can be broadly categorized into:

- *Fixed Costs*: Costs that are constant per transaction, such as brokerage fees.
- *Variable Costs*: Costs that depend on the transaction size, including bid-ask spreads and market impact.
- *Slippage*: The difference between the expected transaction price and the actual execution price.

Impact on Portfolio Performance

High transaction costs can negate the benefits of frequent rebalancing. Incorporating transaction costs into the optimization model encourages more conservative adjustments, potentially leading to better net performance [**garleanu2009dynamic**].

Optimal Rebalancing Frequency

Determining the optimal frequency of portfolio rebalancing involves a trade-off between maintaining the desired asset allocation and minimizing transaction costs. Strategies to address this include:

- *Threshold-Based Rebalancing*: Rebalancing only when asset weights deviate beyond predefined thresholds.
- *Periodic Rebalancing*: Adjusting the portfolio at regular intervals (e.g., monthly, quarterly).
- *Cost-Aware Optimization*: Incorporating transaction costs directly into the optimization problem to balance expected returns against costs.

2.4.3 Existing Approaches to Strategy Switching

Strategy switching involves changing between different portfolio optimization strategies based on market conditions, forecasts, or performance metrics. This adaptive approach seeks to capitalize on the strengths of various strategies under different scenarios.

Regime-Switching Models

Regime-switching models assume that financial markets alternate between different states or regimes (e.g., bull and bear markets). By identifying the current regime, investors can switch to strategies that are better suited to prevailing conditions [**ang2002regime**].

Markov Switching Models These models use Markov processes to model transitions between regimes, allowing for probabilistic predictions of future states and informing strategy selection [**hamilton1989new**].

Meta-Learning and Ensemble Methods

Meta-learning approaches involve training a higher-level model to select the best-performing strategy from a set of candidates. Ensemble methods combine multiple strategies to create a more robust overall strategy [**huang2019building**].

Performance-Based Selection Strategies can be evaluated based on historical or real-time performance metrics, such as Sharpe ratios or drawdowns, with the best-performing strategy selected for implementation [**poterba2000portfolio**].

Rule-Based Switching

Rule-based systems use predefined criteria or indicators to trigger strategy changes. Examples include:

- *Technical Indicators*: Switching strategies based on signals from moving averages, momentum indicators, or relative strength indexes.
- *Economic Indicators*: Adjusting strategies in response to macroeconomic data releases or changes in interest rates.
- *Forecast Confidence*: Modifying the strategy based on the confidence level of predictive models, such as forecast variances from GPR.

Machine Learning-Based Switching

Machine learning models can be trained to predict the optimal strategy based on market data and indicators. Classification algorithms, such as support vector machines or neural networks, can identify patterns associated with the outperformance of specific strategies [**fernandez2018machine**].

2.4.4 Conclusion

Dynamic portfolio management seeks to enhance investment performance by adapting to market conditions and new information. Incorporating transaction costs into the optimization process is crucial for realistic strategy implementation. Existing approaches to strategy switching provide valuable frameworks for developing adaptive strategies, leveraging techniques such as regime-switching models, meta-learning, rule-based systems, and machine learning. This study builds upon these concepts by introducing a probabilistic, threshold-based strategy switching mechanism informed by Gaussian Process Regression forecasts.

3 Methodology

3.1 Data Collection and Preprocessing

3.1.1 Asset Selection and Justification

In this study, we selected a diverse set of 10 assets to capture a wide range of market dynamics and enhance the robustness of our predictive models. The assets include foreign exchange (forex) pairs, commodities such as gold, cryptocurrencies like Bitcoin (BTC), and various stocks from different sectors. The inclusion of these assets allows us to model a comprehensive financial market and test the generalizability of our Gaussian Process Regression models across different asset classes.

The justification for selecting these assets is based on their liquidity, volatility, and significance in global financial markets. Forex pairs and commodities like gold are known for their high liquidity and serve as benchmarks for economic stability. Bitcoin represents the rapidly evolving cryptocurrency market, offering unique volatility characteristics. The selected stocks provide exposure to equity markets and contribute to the diversification of the portfolio.

To assess the complexity of the time-series data for these assets, we utilized entropy-based methods. Specifically, we employed the `OrdianlEntropy` package in Python, which provides time-efficient, ordinal pattern-based entropy algorithms for computing the complexity of one-dimensional time-series. This analysis informed our feature engineering process by highlighting the inherent unpredictability and dynamic behavior of the asset prices.

3.1.2 Data Sources and Time Period

To acquire the historical market data necessary for this study, we employed the EOD Historical Data API, a reputable and comprehensive source for end-of-day and historical financial data across various asset classes. The data retrieval process was automated using a Python script that constructs API requests based on the asset ticker, desired data period (e.g., daily, weekly), and specified date range.

For U.S.-listed assets, the script utilizes the endpoint format:

```
https://eodhd.com/api/eod/{ticker}.US?period={period}&api_token={api_token}&fmt=json&fron
```

where `{ticker}` represents the stock symbol, `{period}` denotes the data frequency, `{api_token}` is the authentication token, and `{start_date}` and `{end_date}` define the data range. For Bitcoin (BTC), which is categorized differently in the API, the script accesses data using the endpoint:

```
https://eodhd.com/api/eod/BTC-USD.CC?period={period}&api_token={api_token}&fmt=json&from=
```

An API token, securely stored using environment variables to maintain confidentiality, is included in the requests for authentication. The script handles HTTP responses by checking for successful status codes and raising exceptions in case of errors, ensuring robust data retrieval.

Upon receiving a valid response, the JSON data is parsed into a pandas DataFrame for efficient data manipulation and analysis. The DataFrame includes essential financial indicators such as open, high, low, close prices, and trading volumes. The data is then saved as a CSV file in a structured directory hierarchy corresponding to each asset, following the path:

```
../Stocks/{ticker}/{ticker}_us_{period}.csv
```

By automating the data fetching and saving process, we ensured consistency and repeatability in data collection of the whole pipeline. This method allowed us to systematically gather historical market data for all selected assets over the specified time periods, providing a reliable dataset for training the Gaussian Process Regression models and conducting backtesting for the portfolio optimization strategies. The use of the EOD Historical Data API ensured that the data was up-to-date and accurate, reflecting real market conditions essential for the validity of our analysis.

Typically, stocks indexes data like S&P500 and Nasdaq100 is fetched from https://www.nasdaq.com/, are used to represent the overall market performance. In this study, we included the S&P 500 index as a benchmark for the U.S. equity market. The S&P 500 index is widely regarded as a barometer for the U.S. stock market and is composed of 500 large-cap companies representing various sectors.

3.1.3 Data Preprocessing and Log Returns

Data preprocessing and feature engineering are critical steps in preparing the dataset for modeling. They ensure that the data fed into the Gaussian Process Regression models are clean, consistent, and informative.

Use of Log Returns In this project, log returns are utilized for modeling asset price movements due to several compelling reasons that align with both theoretical and practical considerations in financial analysis.

Theoretical Foundation in Finance Log returns are integral to many foundational financial models, such as the Black-Scholes option pricing model, which assume that asset prices follow a log-normal distribution. By using log returns, we align our modeling approach with these theoretical frameworks, facilitating more accurate and consistent analyses. GPR models assume normally distributed outputs, and since log returns of log-normally distributed prices are normally distributed, this compatibility enhances the effectiveness of our predictive modeling.

Stability Over Time Log returns exhibit greater stability compared to simple arithmetic returns, particularly in the presence of extreme outliers or during periods of high market volatility. They tend to smooth out spikes and reduce the impact of short-term noise, making the models less sensitive to sudden market anomalies. This stability is crucial for developing robust predictive models that can perform reliably under various market conditions.

Time Consistency (Additivity) One of the key mathematical properties of log returns is their additive nature over time. The total log return over a period is the sum of the log returns over sub-periods:

$$\log\left(\frac{S_t}{S_0}\right) = \log\left(\frac{S_t}{S_{t-1}}\right) + \log\left(\frac{S_{t-1}}{S_{t-2}}\right) + \cdots + \log\left(\frac{S_1}{S_0}\right), \tag{3.1}$$

where S_t is the asset price at time t. This additive property simplifies the computation of returns over arbitrary time horizons, such as weekly or monthly periods, by allowing us to sum daily log returns. It is particularly beneficial for forecasting and portfolio optimization over multi-day horizons, as it facilitates the aggregation of returns without the need for complex compounding calculations.

Normalization of Price Scale Log returns are scale-invariant, meaning they standardize returns across assets regardless of their price levels. Whether an asset is priced at \$1 or \$1,000, the log return brings their percentage changes onto a consistent scale. This normalization simplifies comparisons across assets with vastly different price levels and reduces the need for additional data scaling or normalization procedures. It ensures that no single asset disproportionately influences the model due to its absolute price, allowing for a more balanced and equitable analysis within the portfolio.

Conclusion By incorporating log returns into our modeling framework, we leverage their theoretical compatibility with financial models, enhance stability against market volatility, benefit from their time-additive properties, and achieve scale normalization across diverse assets. These advantages contribute to the robustness and accuracy of our Gaussian Process Regression models and improve the effectiveness of our dynamic portfolio optimization strategies.

Data Normalization and Scaling To bring all features onto a similar scale and improve the numerical stability of the models, we applied data normalization techniques. Specifically, we used min-max scaling to normalize the historical return features and the time index:

$$X_{\text{normalized}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{3.2}$$

where X represents the original feature values, and X_{min} and X_{max} are the minimum and maximum values of the feature, respectively. This scaling transforms the data to a [0, 1] range, facilitating efficient model training.

Treatment of Missing Data and Outliers Financial time-series data often contain missing values and outliers due to market closures, data recording errors, or extreme market events. To address missing data, we employed interpolation methods appropriate for time-series, such as linear interpolation and forward/backward filling, ensuring temporal continuity in the data.

Outliers were identified using the Interquartile Range (IQR) method:

$$\text{IQR} = Q_3 - Q_1, \tag{3.3}$$

where Q_1 and Q_3 are the first and third quartiles, respectively. Data points lying outside 1.5 times the IQR from the quartiles were considered outliers. We assessed these outliers to determine whether they were due to data errors or genuine market anomalies. Genuine outliers representing significant market movements were retained to preserve the dataset's integrity, while erroneous data points were corrected or removed.

Data Splitting and Cross-Validation The dataset was divided into training and testing sets to evaluate the model's predictive performance. The training set consisted of the first 80% of the time period, while the remaining 20% was reserved for testing. This chronological split respects the temporal order of the data, avoiding look-ahead bias.

To further validate the models, we used time-series cross-validation with a rolling window approach. In each iteration, the model was trained on a window of consecutive data points and tested on the subsequent period. This method provides a more robust assessment of the model's performance over time and simulates real-world forecasting conditions.

Sliding Window Approach To denoise the time-series data and reduce short-term fluctuations, we employed a sliding window approach using a centered rolling window mechanism. For each data point in the series, a window of a specified size w was centered around it, and a statistical function was applied to the data within this window to compute a denoised value. The primary function used was the mean, though other functions could be applied as needed.

Formally, let $\{x_t\}_{t=1}^{T}$ represent the original time-series data, and $\{\tilde{x}_t\}_{t=1}^{T}$ denote the denoised series. The denoised value at time t, $\tilde{x}_t$, is calculated as:

$$\tilde{x}_t = \frac{1}{n_t} \sum_{i=t-k}^{t+k} x_i, \tag{3.4}$$

where $k = \left\lfloor \frac{w}{2} \right\rfloor$, and n_t is the number of data points within the window centered at time t. The window size w is an odd integer to ensure symmetry around the central point. At the edges of the time-series (when $t - k < 1$ or $t + k > T$), the window is adjusted by including available data points, and the minimum number of periods is set to 1 to allow computation even with incomplete windows.

To handle any missing values that may arise at the edges due to insufficient data points, we applied forward and backward filling methods. Forward filling propagates the last observed non-missing value forward to fill subsequent missing positions, while backward filling fills missing values by propagating the next observed non-missing value backward. These steps ensure that the denoised series is complete and free from missing values.

This sliding window denoising process effectively smooths the data by averaging over the local neighborhood of each data point, reducing random noise while preserving significant trends and patterns. By enhancing the signal-to-noise ratio in the time-series, this approach improves the quality of the input data for the Gaussian Process Regression models, leading to better predictive performance and more reliable portfolio optimization decisions.

Gaussian Filter Denoising Method To further enhance the quality of the time-series data and reduce high-frequency noise, we employed the Gaussian filter denoising method. The Gaussian filter is a convolutional filter that applies a Gaussian kernel to smooth data by averaging neighboring points with weights determined by the Gaussian function. This technique preserves significant trends and patterns while effectively attenuating random fluctuations and noise.

The Gaussian kernel is defined by the Gaussian (normal) distribution function:

$$G(i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{i^2}{2\sigma^2}\right), \tag{3.5}$$

where i is the index distance from the central data point, and σ is the standard deviation of the Gaussian distribution, controlling the degree of smoothing. A larger σ results in a wider kernel and more extensive smoothing.

The denoised value at time t, $\tilde{x}_t$, is computed by convolving the original time-series data x_t with the Gaussian kernel:

$$\tilde{x}_t = \sum_{i=-k}^{k} G(i) \cdot x_{t+i}, \tag{3.6}$$

where k is the window size parameter determining the range of data points considered around time t.

In our implementation, we applied the Gaussian filter to the closing prices of the assets using a standard deviation $\sigma = 1$, which provides a balanced smoothing effect without overly distorting the data. The following code snippet illustrates the application of the Gaussian filter using the `gaussian_filter` function from the SciPy library in Python:

```
if isFiltered:
    df['filtered_close'] = gaussian_filter(df['close'], sigma=1)
```

In this code, `df['close']` represents the original closing price data, and `df['filtered_close']` stores the denoised data after applying the Gaussian filter. The `isFiltered` flag allows for conditional application of the filtering process.

By using the Gaussian filter denoising method, we effectively reduced the impact of noise and short-term fluctuations in the financial time-series data. This preprocessing step enhances the signal-to-noise ratio, allowing the Gaussian Process Regression models to focus on underlying market trends and improving the accuracy of the return predictions. The combination of the sliding window approach and Gaussian filtering provides a robust methodology for data smoothing, contributing significantly to the reliability and performance of the predictive modeling and subsequent portfolio optimization.

The filtered data retained essential market movements while reducing random fluctuations, allowing the Gaussian Process Regression models to focus on meaningful signals.

3.1.4 Summary

By carefully selecting assets, sourcing reliable data, and meticulously preprocessing the dataset, we established a solid foundation for our predictive modeling. The combination of normalization, outlier treatment, strategic data splitting, and denoising ensured that the inputs to our models were of high quality. These steps are crucial for enhancing the performance of the Gaussian Process Regression models and, ultimately, for developing effective portfolio optimization strategies.

3.2 Model Development

3.2.1 Performance metrics

MSE, MAE, RMSE, MAPE Sharpe ratio

3.2.2 Baseline models

ARIMA model

3.2.3 Hyperparameter tuning

3.2.4 Model evaluation and comparison

3.2.5 Portfolio optimization strategies

3.2.6 Transaction costs and rebalancing

3.2.7 Strategy selection mechanism

3.2.8 Implementation details

3.3 Forecasting Approach

Describe the iterative forecasting method and how predictions are updated daily.

3.3.1 Multi-input Gaussian Process Regression model

3.3.2 Kernel functions selection and hyperparameter optimization

3.3.3 Implementation of rolling window predictions

3.3.4 Model updating mechanism

3.4 Portfolio Optimization Strategies

Explain each portfolio optimization strategy in depth, including mathematical formulations and constraints.

3.4.1 Traditional Strategies

Maximum return strategy formulation Minimum volatility approach Maximum Sharpe ratio optimization Constraint specifications and justifications as Baseline models

3.4.2 Dynamic Strategy

Probability distribution modeling

3.5 Probability Estimation: $P(S_1 > S_2)$

When estimating the probability $P(S_1 > S_2)$ for two random variables S_1 and S_2, the methodology depends on the nature of their distributions and their dependence structure. This section outlines three approaches: numerical integration, Monte Carlo simulation, and the use of copulas for dependent variables.

3.5.1 Numerical Integration

If the probability density functions (PDFs) of S_1 and S_2 are known, the probability $P(S_1 > S_2)$ can be expressed as:

$$P(S_1 > S_2) = \int_{-\infty}^{\infty} \int_{y}^{\infty} f_{S_1}(x) f_{S_2}(y)\, dx\, dy, \tag{3.7}$$

where:

- $f_{S_1}(x)$ is the PDF of S_1,
- $f_{S_2}(y)$ is the PDF of S_2.

This double integral represents the joint probability over the region where $S_1 > S_2$, and it requires numerical methods for evaluation when closed-form solutions are unavailable.

3.5.2 Copulas for Dependence

Especially, When S_1 and S_2 are dependent, the joint distribution can be modeled using a copula. A copula is a function that describes the dependence structure between random variables, linking their marginal distributions. Let $F_{S_1}(x)$ and $F_{S_2}(y)$ represent the cumulative distribution functions (CDFs) of S_1 and S_2, respectively. The joint CDF can be expressed as:

$$F_{S_1,S_2}(x, y) = C(F_{S_1}(x), F_{S_2}(y)), \tag{3.8}$$

where $C(u, v)$ is the copula function.

The probability $P(S_1 > S_2)$ can then be computed as:

$$P(S_1 > S_2) = \int_{-\infty}^{\infty} \int_{y}^{\infty} \frac{\partial^2 C(F_{S_1}(x), F_{S_2}(y))}{\partial u \partial v} \, dx \, dy. \tag{3.9}$$

The steps to compute this are:

1. Determine the marginal distributions $F_{S_1}(x)$ and $F_{S_2}(y)$ for S_1 and S_2.
2. Select an appropriate copula function $C(u, v)$ based on the dependence structure (e.g., Gaussian, Clayton, or Gumbel copulas).
3. Use numerical methods to evaluate the double integral above.

Copulas are particularly effective when the marginal distributions are non-normal or when the dependence structure is non-linear and cannot be captured by simple correlation measures.

3.5.3 Monte Carlo Methods

If the distributions of S_1 and S_2 are not explicitly known but sampling from these distributions is possible, a Monte Carlo simulation can be used to estimate $P(S_1 > S_2)$. The steps are as follows:

1. Generate N independent samples $S_1^{(i)}$ and $S_2^{(i)}$ from the respective distributions of S_1 and S_2.
2. Count the number of instances where $S_1^{(i)} > S_2^{(i)}$. Denote this count by n.
3. Estimate the probability as:

   $$P(S_1 > S_2) \approx \frac{n}{N}, \tag{3.10}$$

   where

   $$n = \sum_{i=1}^{N} \mathbb{I}(S_1^{(i)} > S_2^{(i)}), \tag{3.11}$$

   and $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition is true and 0 otherwise.

Monte Carlo simulation is particularly useful for complex distributions or dependent variables, where analytical integration is impractical. In our case, we have multiple assets with potentially non-normal distributions and complex dependencies, making Monte Carlo methods a valuable tool for estimating $P(S_1 > S_2)$. Specifically, we will use Monte Carlo simulation to estimate the probability of one asset outperforming another in our portfolio optimization strategies. And we chose a sample size of $N = 10,000$ to ensure accurate probability estimates.

3.5.4 Comparison of Methods

- **Numerical Integration:** Provides an exact solution given the PDFs of S_1 and S_2, but computationally intensive for high-dimensional problems or non-standard distributions.
- **Copulas:** Allows modeling of complex dependence structures, particularly useful for non-normal distributions or asymmetric dependencies.
- **Monte Carlo Simulation:** Flexible and practical alternative when sampling is straightforward, though its accuracy depends on the number of samples N.

Each method has its strengths and limitations, given the availability of distributional information and computational resources of our case, we chose to use Monte Carlo simulation for estimating $P(S_1 > S_2)$.

3.5.5 Strategy switching criteria

Describe the criteria for switching between portfolio optimization strategies based on the estimated probability $P(S_1 > S_2)$. We set a threshold probability τ such that if $P(S_1 > S_2) > \tau$, the strategy with the higher expected return is selected, and if $P(S_1 > S_2) \leq \tau$, the strategy with the lower volatility is chosen. This threshold ensures that the strategy selection is based on a balance between return and risk, incorporating the estimated probability of one asset outperforming the other.

3.5.6 Position holding logic

3.5.7 Transaction cost considerations

3.6 Backtesting Framework

Describe the backtesting process and how the strategies are evaluated.

4 Results and Analysis

4.1 Model Performance Analysis

4.1.1 Comparison with benchmark models

- ARIMA model
- Markets are efficient, and all investors have access to the same information.
- Asset returns are normally distributed and can be described by their mean (expected return) and variance (risk).

Comparison with ARIMA MSE, Present the prediction accuracy of the GPR models for the target assets.

4.1.2 Analysis of prediction intervals

4.1.3 Model robustness and generalization

4.2 Portfolio Optimization Outcomes

4.2.1 Strategy Performance Comparison

Return analysis

Risk metrics Transaction costs impact

Strategy switching frequency analysis

Comparative Analysis: Compare the performance of all strategies, highlighting the strengths and weaknesses of each.

4.2.2 Dynamic Strategy Evaluation

Probability threshold sensitivity Strategy switching effectiveness Portfolio turnover analysis Risk-adjusted performance metrics

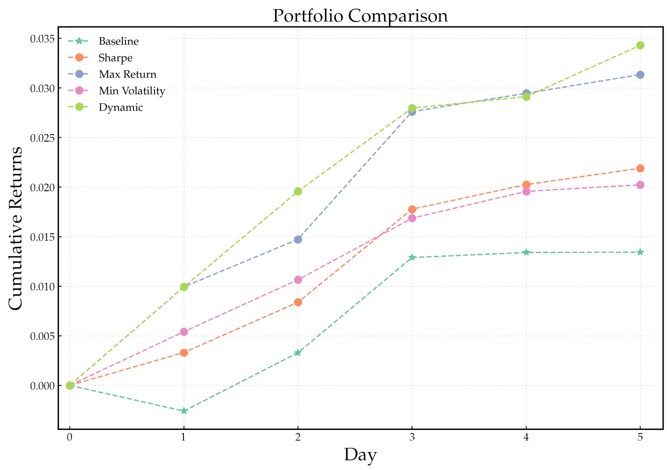

Figure 4.1: Portfolio Comparison

4.3 Backtesting Results

Provide detailed results from the backtesting, including cumulative returns, volatility, Sharpe ratios, and transaction costs for each strategy.

4.3.1 Transaction Costs impact

4.4 Robustness Tests

4.4.1 Different market conditions

4.4.2 Hyperparameter sensitivity

4.4.3 Out-of-sample performance

4.4.4 Statistical significance tests

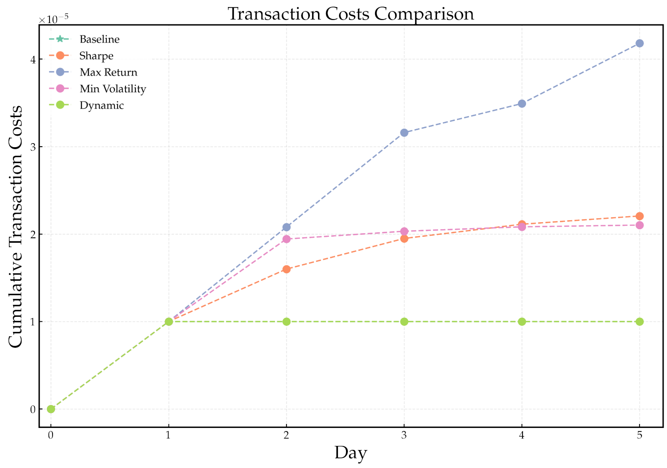

Figure 4.2: Transaction Costs Comparison

5 Discussion

5.1 Interpretation of Results

5.1.1 Key findings and insights

Discuss what the results mean in the context of your research objectives.

5.1.2 Dynamic Strategy Insights

Delve into why the dynamic strategy outperformed others, considering market conditions and model performance.

5.1.3 Model robustness and generalization

5.2 Implications for Practitioners

Explain how your findings can be applied in real-world portfolio management.

5.2.1 Limitations of the approach

Acknowledge any limitations in your study, such as data constraints, model assumptions, or external factors.

5.2.2 Future research directions

Suggest potential areas for further research based on your findings.

5.3 Comparative Analysis

Comparison with existing methods

5.3.1 Advantages and disadvantages

Discuss the pros and cons of your approach compared to traditional portfolio optimization strategies.

5.3.2 Implementation challenges

Discuss any practical difficulties in applying your approach to real-world scenarios.

5.3.3 Market impact considerations

6 Conclusion

6.1 Summary of Findings

6.1.1 Summary of Findings

Recap the main results and how they address your research questions.

6.1.2 Key findings and insights

Discuss what the results mean in the context of your research objectives.

6.2 Recommendations

Offer suggestions for practitioners based on your findings.

6.3 Future Research Directions

6.3.1 Model improvements

Discuss potential enhancements to your model or methodology.

6.3.2 Additional strategy considerations

Suggest new strategies or modifications to existing ones.

6.3.3 Alternative applications

Propose other areas where your approach could be useful.

6.3.4 Scalability considerations

Abbreviations

TUM Technical University of Munich

GPR Gaussian Processes Regression

GP Gaussian Processes

ML Machine Learning

ARIMA AutoRegressive Integrated Moving Average

MPT Modern Portfolio Theory

List of Figures

List of Tables

Bibliography

[HPW17] J. B. Heaton, N. G. Polson, and J. H. Witte. "Deep Learning for Finance: Deep Portfolios". In: *Applied Stochastic Models in Business and Industry* 33.1 (2017), pp. 3–12.

[Lam94] L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.

[Lin65] J. Lintner. "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets". In: *The Review of Economics and Statistics* 47.1 (1965), pp. 13–37.

[Mar52] H. Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[Mos66] J. Mossin. "Equilibrium in a Capital Asset Market". In: *Econometrica* 34.4 (1966), pp. 768–783.

[Qia05] E. Qian. *Risk Parity Portfolios: Efficient Portfolios Through True Diversification*. Panagora Asset Management, 2005.

[Sha64] W. F. Sharpe. "Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk". In: *The Journal of Finance* 19.3 (1964), pp. 425–442.