. MASSAT Framework : Financial Portfolio Playoff

Summary

Documentation
Soham Mukherjee

Abstract

This paper introduces a seemingly new method MASSAT which cleverly incorporates(i.e. fuses) Bootstrap-Aggregation and Boosting simultaneously over the same set of data(though the independent variables differing for different parallel modules); thus resulting into a stability and reduced-variance in results(reminiscence of Ensemble methods in machine-learning like Random-Forest etc.). Also the reinforcement-learning module makes sure to learn from previous mistakes as in Boosting based methods.
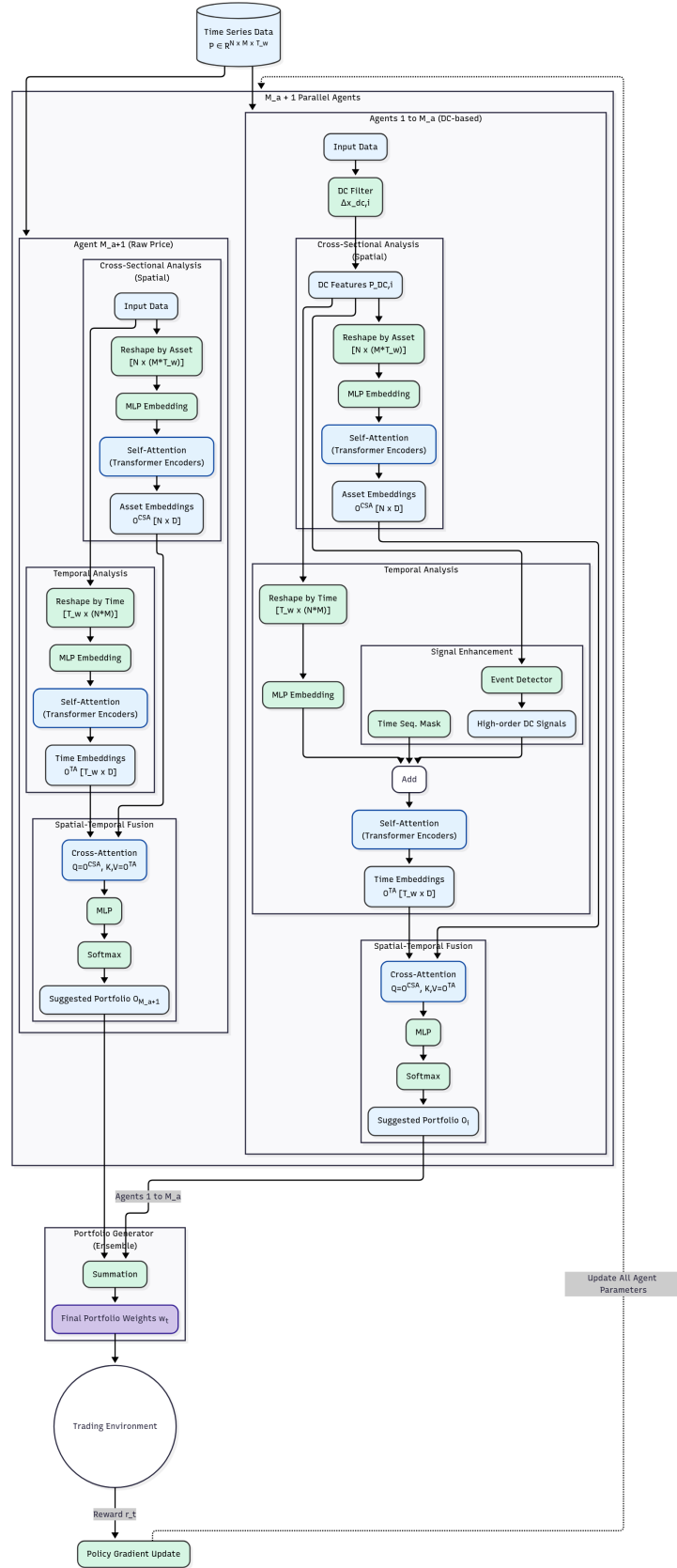
Figure 1: **High Level Architectural Design**

Contents

1. Flow of Architecture

The MASAAT framework processes raw market data through a multi-stage, parallelized architecture to produce an optimized portfolio.

1.A. Multi-Perspective Data Preparation

Input Data: The process begins with a raw time series tensor $P \in \mathbb{R}^{N \times M \times T_w}$, representing N assets, M features, over an observation window of T_w time steps.

   Parallel Processing Streams: The input data is fed into $M_a + 1$ parallel streams.

- **Price Agent Stream:** One stream uses the raw price tensor P directly.

- **DC Agent Streams:** M_a other streams first pass the price data through distinct Directional Change (DC) filters. Each DC filter has a unique predefined threshold (Δx_{dc}) to capture price movements of different magnitudes. This generates M_a different DC feature maps, $\{P_{DC,1}, \ldots, P_{DC,M_a}\}$, which are less noisy than the raw price data.

1.B. Dual-Analysis within Each Agent

Each of the $M_a + 1$ agents performs two types of analysis concurrently on its assigned data.

1.B.I. Cross-Sectional Analysis (CSA)

1. **Tokenization:** The data tensor is reshaped so that each of the N assets becomes a single token.

2. **Embedding:** A Multi-Layer Perceptron (MLP) converts each asset token into a high-dimensional embedding vector.

3. **Self-Attention:** Transformer encoders calculate attention scores between every pair of assets, learning correlations.

4. **Output:** The CSA module outputs an asset-oriented embedding $O^{CSA} \in \mathbb{R}^{N \times D}$.

1.B.II. Temporal Analysis (TA)

1. **Tokenization:** The data tensor is reshaped so each of the T_w time points becomes a single token.

2. **Signal Enhancement (DC Agents):** Two additional signals are incorporated: High-Order DC Signals (emphasizing significant events) and a Time Sequence Mask (weighting recent data more heavily).

3. **Self-Attention:** Transformer encoders calculate the relevance between every pair of time points.

4. **Output:** The TA module outputs a time-oriented embedding $O^{TA} \in \mathbb{R}^{T_w \times D}$.

1.C. Fusion

Within each agent, the two output embeddings (O^{CSA} and O^{TA}) are fused using cross-attention, where O^{CSA} acts as the "query" and O^{TA} as the "key" and "value". This step determines which historical time points are most important for each asset, resulting in a suggested portfolio vector $O_i \in \mathbb{R}^{N \times 1}$ from each agent i.

1.D. Ensemble Portfolio Generation

The final Portfolio Generator takes the suggested portfolio vectors from all $M_a + 1$ agents and combines them through summation, followed by a Softmax function to produce the final portfolio weight vector, w_t.

1.E. Reinforcement Learning Loop

The portfolio w_t is executed, and a reward r_t is calculated. The experience tuple (w_t, r_t, P, P_{DC}) is stored. Periodically, a *policy gradient method* uses these experiences to update all learnable parameters to maximize cumulative reward.
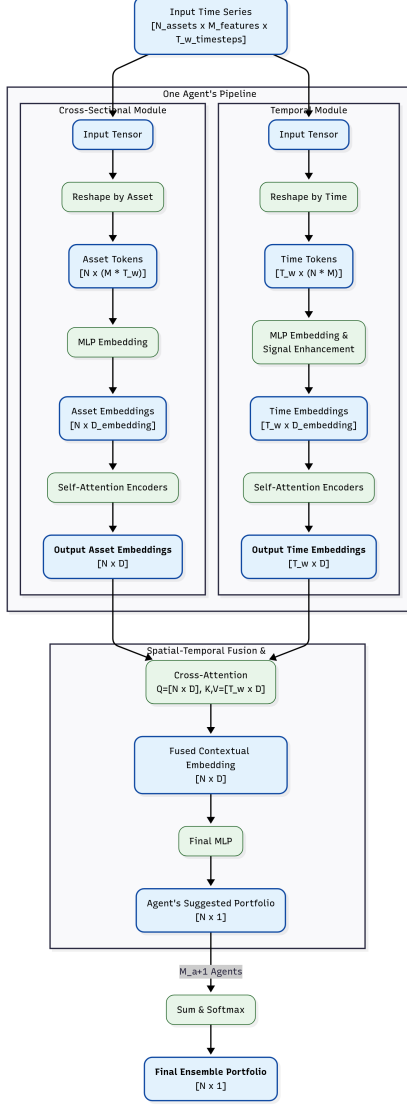
2. The WHY's behind the design

2.A. A Multi-Agent System?

Reason: A single model learning from noisy price data can easily develop biases or miss representative correlations of the market.

Advantage: By creating multiple agents observing the market through different *lenses* (raw price vs. different DC thresholds), the framework induces diversity. If one agent commits an error, its mistake is averaged out by the others(the advantage of using an ensemble approach). This makes the final decision more robust and less prone to bias.

2.B. Directional Change (DC)?

Reason: Traditional time-based data is filled with random fluctuations that obscure the true underlying trend.

Advantage: Directional Change (DC) is an event-based sampling-method for analyzing financial market data that identifies significant turning points in asset price trends, rather than relying on fixed-time interval data. This acts as a natural noise filter. Using multiple DC thresholds allows agents to focus on trends at different scales.

2.C. Attention over CNNs or RNNs?

Reason: RNNs/LSTMs struggle with long-term dependencies due to issues like vanishing-gradients. CNNs are sensitive to relative order of the assets and fail to capture global correlations and long-term dependencies.

Advantage: The self-attention mechanism in Transformers captures global dependencies. It can calculate the direct relationship between any two assets (in CSA) or any two time points (in Temporal Analysis), regardless of their sequence position, which is crucial for portfolio analysis.

2.D. Separate CSA and TA Modules?

Reason: Learning inter-asset (spatial) and time-series (temporal) patterns simultaneously can be difficult and lead to uninterpretable features.

Advantage: A clear division of labor ensures that spatial and temporal patterns are learned explicitly and independently first. The CSA module becomes an expert on various asset correlations, on the other hand the TA module becomes an expert on market trends. These specialized insights are then intelligently fused.

3. Scope for Improvements and improvisions

3.A. simple Agent Fusion

scope: The framework combines agent outputs via simple summation, treating all agents as equally important at all times. In a low-volatility market, the raw price agent might be more informative, while in a volatile market, a high-threshold DC agent might be more representative.

improvision: Dynamic Agent Weighting. An attention-based gating mechanism can be added in the Portfolio Generator. This addition will learn to assign dynamic weights to the suggestions from each agent based on the current market state.

3.B. Fixed, Predefined DC Thresholds

scope: The DC filter thresholds are pre-defined constants and require extensive manual tuning. Optimal values may change across different market regimes.

improvision: Learnable DC Thresholds. The DC thresholds could be redefined as learnable parameters. The RL training loop could dynamically adjust these thresholds, making the system truly "self-adaptive."

2

3.C. Limited Information Scope

scope: The model operates solely on quantitative price data, ignoring qualitative information like news, sentiment, and fundamentals.

improvision: Multi-Modal Agents. The multi-agent architecture has a well-defined template for expansion with a new agent. A new Language Model-based agent(that can use models fine-tuned on financial data like FinBERT etc.) could be added to analyze news sentiment for each stock. Its output embedding could be fused with the outputs from the price-based agents, providing a more holistic view of the market.