Phase Two Presentation: RL vs. Naive Diversification

Robustness and Frictions Across Market Regimes

Andre Sealy

Advisor: Jingrui (Victoria) Li
Financial Engineering 800
Stevens Institute of Technology

November 25th, 2025

Overview

- Addressing the stationarity issues of the Decision Transformer.
- Addressing the computational issues with extracting state trajectories for the Decision Transformer.
- Model updates and performance.

Background

Reinforcement Learning Environment

Recall the environment, $\mathcal{M}$, which we have established with the following parameters

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \mathcal{R}, \Theta)$$

where

- $\mathcal{A} \in \mathbb{R}^N$ is the action space, where N are the number of assets.
- $\mathcal{S}$ is the state space, which is a linear combination of different position states.
- P are the transition dynamics governed by market dynamics and portfolio evolution.
- $\mathcal{R}$ is the reward function or the return on the portfolio.
- Θ are the custom trading parameters

Reinforcement Learning Environment

The custom parameters of Θ are denoted by the following:

$$\Theta = \begin{cases} H_{\text{max}}, & H_{\text{max}} = 100 \\ V_0 \in \mathbb{R}^+ & V_0 = 1{,}000{,}000 \\ \tau \in [0,1), & \tau = 0.005 \\ N \in \mathbb{N}, & N = 394 \\ I \in \mathbb{N}, & I = 8 \\ K \in \mathbb{N}, & K = 2 \\ \alpha \in \mathbb{R}^+, & \alpha = 10^{-1} \end{cases}$$

where H_{max} is the maximum shares traded per stock, V_0 is the initial portfolio value, τ is the transaction cost percentage, N are the number of assets, α is the reward scaling factor, I and K are the number of indicators and regime variables respectively.

Stationarity Issues

Stationarity Issues

Running reinforcement learning (RL) algorithms with a decision transformer (DT) across many different states introduces stationarity issues, such as

- **Stationarity and Regime Shift:** DT may conflate regimes, failing to recognize when a feature distribution changes.

Stationarity Issues

Running reinforcement learning (RL) algorithms with a decision transformer (DT) across many different states introduces stationarity issues, such as

- **Stationarity and Regime Shift:** DT may conflate regimes, failing to recognize when a feature distribution changes.
- **Poor Feature Scaling and Instability:** Transformer architectures are sensitive to outliers and input scale.

Stationarity Issues

Running reinforcement learning (RL) algorithms with a decision transformer (DT) across many different states introduces stationarity issues, such as

- **Stationarity and Regime Shift:** DT may conflate regimes, failing to recognize when a feature distribution changes.
- **Poor Feature Scaling and Instability:** Transformer architectures are sensitive to outliers and input scale.
- **Empaired Risk and Exposure Control:** The DT may unknowingly overweight risky assets in high-volatility regimes or fail to detect regime-dependent risk.

State Statistics

State statistics help us diagnose market structures and create tailored signals for robust model training and evaluation. The purpose issues

- **Characterize Market Environments:** State statistics quantify the typical and extreme behaviors of the features that describe the market or economic regimes.
- **Enable Regime-Specific Modeling:** By segmenting historical data based on regime labels, conditional statistics can be computed.
- **Support Feature Engineering and Normalization:** Aggregate statistics allow for robust feature scaling, improved training stability, and the direction of non-stationarities or structural breaks across different periods

State Statistics

- **Mean:** Central tendency of a state feature.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- **Variance and Standard Deviation:** Measure the dispersion or risk.

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2 , \quad \sigma = \sqrt{\sigma^2}$$

- **Covariance and Correlation:** Quantify comovements.

$$\mathrm{Cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y) , \quad \rho_{xy} = \frac{\mathrm{Cov}(x, y)}{\sigma_x \sigma_y}$$

State Statistics

Utilizing state statistics for Transformers is supported by research from Liu, Wu, Wang, and Long (*Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting*, 2022).

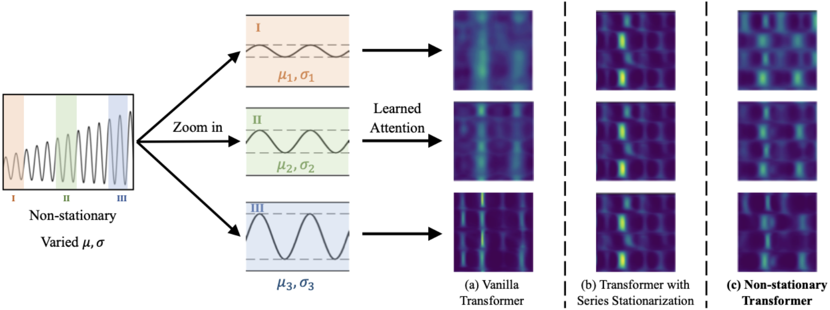

Figure 1: Visualization of learned temporal attentions for different series with varied mean μ and standard deviation σ.

Computational Issues

Trajectories

Recall that DT frames RL problems as a sequence modeling problem, which leverages the mathematics and architecture of Transformer models by training pre-collected (offline) trajectories and encoding each input sequence to predict future returns.

Trajectories

Recall the sequence of states, actions, and rewards collected over time s_t, a_t, r_t and the state, action, and reward at timestep t. This sequence is defined as the **trajectory** in the MDP, defined by the following:

$$\tau = \{(s_0, a_0, r_0), (s_1, a_1, r_1), \ldots, (s_T, a_T, r_T)\},$$

where T is denotes the final timestep in the trajectory.

Trajectories

This process of encoding trajectories is computationally expensive because the process involves collecting entire episodes of agent-environment interactions and handling, storing, and processing large datasets.

Computational Issues

Running reinforcement learning (RL) algorithms with decision transformers (DT) across many different states introduces several computational challenges, such as,

- **High Memory Requirements:** Significant amounts of memory are required for storing and processing trajectories.
- **Data Inefficiencies:** Many different states demand more samples to cover the space adequately, compounding dataset size and training time.
- **Sequence Alignment and Temporal Credit Assignment:** The more states the DT has, aligning and credit outcomes to earlier actions becomes computationally more difficult.
- **Model Size and Training Cost:** DT may need larger transformer models, which require more computational resources, longer training times, and powerful hardware.

High Performance Computing

High-performance computing (HPC) can significantly accelerate research on RL for portfolio optimization, especially when using computationally demanding models like decision transformers in complex regimes.

- **Parallelization and Speed:** HPC environments allows computational to be distributed across many CPUs and GPUs, drastically reducing the time for training RL models and grid searches over hyperparameters.
- **Larger Models and Datasets:** We can train deeper transformer architectures and process longer trajectories or higher-dimensional state spaces, overcoming the memory and compute limits of a regular PC.
- **Handling Frictions and Constraints:** Simulating realistic trading frictions (transaction costs, slippage, etc.) requires substantial computation and state tracking.

JARVIS

Stevens Institute of Technology's HPC cluster JARVIS is a state-of-the-art computing resource designed to support advanced research across the university.

- **Compute Resources:** JARVIS consists 55 codes, providing 3,168 CPU cores and 32 GPUs, including 8 advanced Nvidia L40s GPUs.
- **Memory:** The cluster has 14 TBs of memory, which enables it to handle complex, memory-intensive workloads.
- **Storage:** It includes 1.2 petabytes (PB) of storage, supporting larged dataset and model checkpoint management.

Figure 2: Not actually Steven's JARVIS

Workstation Computational Comparisons

Table 1: Benchmark of DT Runtime with Various Computer Specs.

	Hanlon Lab	$ 10 Million Studio (mine)	JARVIS
Workstation Specifications:			
CPU	Intel i9-11900	AMD Ryzen 9 5950X	Intel
GPU	NVIDIA GeForce 3070	NVIDIA GeForce 4090 TI	NVIDIA L40S
Memory	64GB (2x32 GB)	128GB (4x32 GB)	17TB
Runtime Detection	GPU	GPU	CPU or GPU
Total Training Steps	600,000	1,000,000	Significantly more
Total Runtime	125 Hours	27 hours	Significantly faster
Crashes Occasionally?	✓		

Model Updates

Agent Response

Anthony Curcio-Petraccoro said 7 hours ago

Hi Andre,

We are contacting you regarding your ticket: https://support.stevens.edu/helpdesk/tickets/110020

Sincerely,
Anthony Curcio-Petraccoro
Systems

To view the status of the ticket or add comments, please visit https://support.stevens.edu/helpdesk/tickets/110020

You have been granted access to the HPC Cluster "JARVIS".

You can access it using your Stevens's credentials and MFA.

You can connect from on-campus or by using VPN if off-campus.

To connect, type this in the terminal:

ssh username@jarvis.stevens.edu

More information can be found at JARVIS (SharePoint site) and Research Computing Services (SharePoint Site)

Thank you!

Figure 3: Ticket with Tech Support Approving JARVIS Access.

Model updates

Model Updates

- Incorporating state statistics to summarize the behavior and properties of the states encountered by the RL agent.
- Incorporating macroeconomic variables for regime detection in accordance with the Chen, Peiger, and Zhu paper (Deep Learning in Asset Pricing)
- Including a single regime increased the state size from 3,931 to 4,324.

Model Updates

Figure 4: Macroeconomic Regimes to be included in the Decision Transformer.

Model Updates

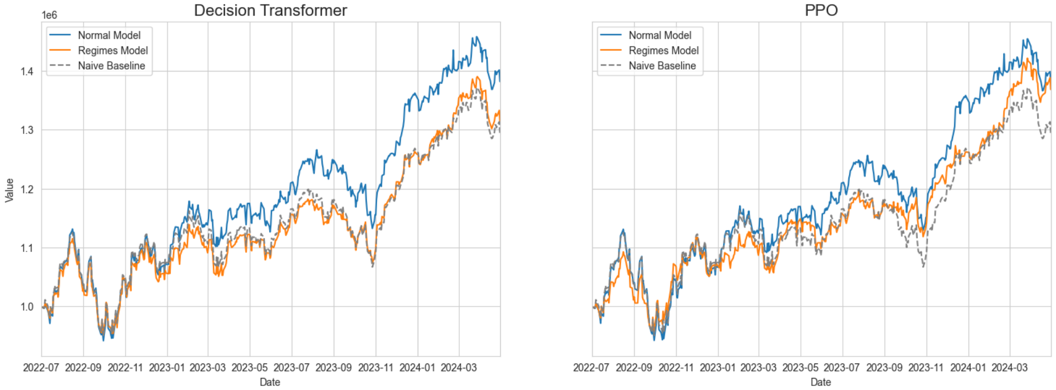

Figure 5: Performance of Decision Transformer and PPO with and without regimes

Table 2: Reinforcement Learning Algorithms Parameters

	Naive	DT	PPO
Without Regimes:			
Annual Return	0.151657	0.160969	0.187503
Sharpe Ratio	0.98472	1.071349	1.300689
Max Drawdown	-0.155387	-0.145308	-0.122458
With Regimes:			
Annual Return	0.151657	0.148791	0.249243
Sharpe Ratio	0.98472	0.966856	0.15500
Max Drawdown	-0.155387	-0.150140	-0.159952

Next Steps

Next Steps

The following steps will be incorporated, as soon as I figure out how JARVIS works

- Increasing the step size and credit assignment for the model.
- Work with different regimes (unemployment, oil prices, interest rates, etc.).
- Increase the number of assets.

Questions?