Technical Analysis as a Foundation for Reinforcement Learning in AI-Driven Portfolio Management

Financial markets are complex dynamic systems with billions of interacting variables and agents. Understanding market movements is an ever-evolving challenge due to their intricate and non-static nature. One method to combat this challenge is in the form of traditional technical analysis. Technical analysis utilizes historical price data and trading volume to suggest actions of when to buy, sell, or hold an equity. It has been widely adopted in portfolio management but has significant limitations. The approach is static and does not adapt to the changing dynamics of financial markets.

This has been the catalyst for novel approaches to portfolio management. The recent advancements in Artificial Intelligence have created an opportunity to modernize traditional techniques, through methods such as Reinforcement Learning (RL). The RL methodology incorporates the Artificial Intelligence (AI) to turn static traditional approaches into intelligent dynamic agents. RL utilizes innovative Machine Learning (ML) techniques to interact with live environments to model its approach to the stock market.

The objective of this capstone project is split into three major components.

1. **Technical Analysis Exploration**: Conduct an in-depth analysis of technical analysis techniques across different sectors, market caps, and macroeconomic cycles. This can be compared to traditional hold-and-buy strategies that take place during similar time periods across portfolios to evaluate success.
2. **Reinforcement Learning Integration**: Development of reinforcement learning algorithms which are adaptive to live market conditions which have been created from the insights from the number of technical analysis approaches. Using Return On Investments (ROI) and decreased risk as the reward for models.
3. **Trading Bot Development**: Design and deploy an AI trading bot built of the accumulation of knowledge from technical analysis and algorithms from reinforcement learning.

The success of this project will come in the form of significant improvements on passive and algorithmic investment techniques which are utilized by most portfolio managers.

The Yahoo Finance API will serve as the primary data source for the project. It has candle stick data (open, high, low, close, adjusted close price and trading volumes), as well as offering financial statements, balance sheets and other fundamental business data. The Yahoo Finance API is reliable as it is used by millions in Python environments. The data is comprehensive and relevant to the demands of this project.

There will need to be significant preprocessing of data and an effective way to store it. This is going to be done through nested dictionaries which can store the data locally within Python.
The project will employ a combination of techniques for the different objectives.

Technical Analysis

The technical analysis component is going to use renowned and high-volume traded ETF's (Exchange Traded Funds) of the 11 industry sectors using Global Industry Classification Standard (GICS). This will be used later as a parameter when looking at companies to invest in as they belong to a specific sector. The technical analysis is also going to look at different macroeconomic cycle periods and split up success based on that. `

Methods used:

- Algorithm creation
- Parameter Optimization
- Exploratory Data Analysis
- Portfolio investing style - looking at the portfolio as a whole rather than multiple stocks

Techniques (will be expanded upon)

- Bollinger Bands - John Bollinger created a channel which is where 95% or 2std above and below a moving average hold the data. If data goes above or below this line it triggers a buy or a sell signal
- Relative Strength Index (RSI)
- Moving Average Convergence/Divergence (MACD)
- Ichimoku cloud

Reinforcement Learning

Methodology

Using the reinforcement learning framework of agents, environments, states, actions and rewards allows for a number of techniques to adopt.

Techniques

- Q-learning : the agent learns a Q-value function and utilizes it when given actions. It balances exploring new actions and exploiting the current understandings to maximize rewards.
- Deep Q-Learning (DQN) : the extension of Q-learning which includes a deep neural network which stores past experience to find and break correlations to improve learning.

AI Trading Bot

Architecture

- Data Ingestion: Use the YF API to process real time data
- Feature Engineering: Use technical indicators are features to represent state of market
- RL Agent: Implement the RL methods
- Action Execution: Based on agents decision, bot will simulate purchasing a stock.
- Feedback loop: Bot will review its outcomes and update policy or value functions

Training and Simulation

- Backtesting : before deploying bot to live environment train it on historical data, utilizing paper trading to not risk real capital

- Live trading : this will not be used with real capital but it will following the market and make decisions at the end of each day and will hopefully run for a week or two prior to completion of capstone project

This project harnesses a lot of publicly used data but ensuring that if further data methods such as web scraping are required not using private or inside data to make trading decisions is important. Ensuring that the AI does not engage in manipulating or contribute to market instability. There is also an overarching question of whether utilizing a bot to purchase and sell stocks is ethical, this is simply for research purposes and no real capital investment will be used.

The timeline of this project begun during summer break (June 2024) with investigating how this project could be set up but no serious milestones were made. However, it does take out some of the nuance that is associated with the beginning of such a large-scale project. The project will likely go in this order

Weeks 1-5 : Technical Analysis

1. Create algorithm to be implemented
2. Utilize buy/sell techniques
3. Read research papers on techniques
4. Optimize parameters using ML techniques
5. Finalize technique
6. Evaluate portfolio management success
7. Repeat steps 1-6 for all technical analysis techniques

Weeks 6-9 Development of Reinforcement Learning Algorithms

Weeks 10 - 12 Backtest and optimize to create the AI trading bot

Weeks 13-14 Evaluate model

Week 15 Test model with live data

Week 16 Present Findings

The overall goal of this experiment is to implement a real-time trading bot that utilizes technical analysis and reinforcement learning. This should come alongside a full evaluation of which traditional technical analysis techniques perform best with different parameters being optimzied. Looking at a time period of 2000-2024 covers all cycles of the macroeconomic business cycle which can allow for a better understanding of which methods perform better during which time periods. Utilizing ETF's to represent different sectors will also act as a way of describing which sectors perform best with different techniques during different states of the economy.

The greatest challenge that will face this is the complexity of the model. Reinforcement learning and AI trading bots have billions of dollars invested into this because of their dynamic and extremely complex nature. This represents the magnitude of resources that need to go into creating an adaptive and successful model. Another issue is overfitting, overfitting can happen when the historical data and its metrics are similar to live scenarios but do not follow the same patterns and this is because of how complex financial markets can be.

I am extremely excited by the possibilities capable of pursuing such a project. It incorporates modern approaches such as ML, RL, and AI to overcome an extremely complex and dynamic issue. There are going to be challenges and I expect that there will be many hours spent trying to solve a problem with slow progress. However, I am excited to get started and be a part of a new wave of approaches to portfolio management.