Executive Summary

Problem

With the rise of machine learning and artificial intelligence, research in the financial sector has gained new powerful resources and tools for improving investment performance. As a result, algorithmic trading programs emerged. These mostly rule-based algorithms have numerous advantages over human traders. For instance, they are able to execute trades faster, allowing them to react to the smallest price changes. Additionally, they are not driven by emotions and do not place trading orders by mistake. For these reasons, they quickly gained a lot of interest and accounted for approximately 75% of the U.S. stock exchange's trading volume, at least as of 2009 (Chan, 2009). However, these algorithmic trading programs lack intuition because of their rule-based characteristics. Moreover, they are often based on predictive signals, which constantly change due to the financial market's dynamic nature and do not incorporate the preferences of an investor. To overcome these issues, we introduce reinforcement learning (RL) to trading.

Method

We apply the RL approach to futures trading. Therefore, we choose the two liquid and publicly traded futures contracts, the S&P 500 E-Mini (ES) and the 10-Year U.S. Treasury Notes (ZN), from January 2010 to January 2021, assuming a roll-over over the quarterly expiration contracts. Furthermore, we extend our intraday data of 15-minute time intervals with several technical indicators, to provide our program with less noisy data.

Unlike ordinary algorithmic trading programs, RL learns based on interaction. The trading process thus must be reformulated as an interactive and sequential decision-making problem. In this problem, we have the two entities of an agent, which is the trader, and an environment, which represents the exchange. These two components build an interactive cycle. While the environment presents the agent with the current market situation, the agent places its trades in the environment. These trades are then evaluated by the environment, and the agent is given a reward indicating how good the trades were, as well as a new market situation. This loop is then repeated. Eventually, the agent develops the lacking intuition from rule-based models, as it is able to sense the environment and therefore adapt to the next market situation. This process can be formalized by the Markov Decision Process (MDP) theory. Its fundamental assumption is that every market situation is the result of only its last previous situation, and all the information about the past is contained within every subsequent state. Furthermore, we can incorporate investor preferences in the formulation of the reward function as it directs the RL trader in its trading decisions.

We introduce the components in section 2, the handling of the data in section 3, as well as the explanation of the underlying theory in section 4. The manual for running the program is in the README.md file of the code repository.

Results

We train our trading agents in total on ten years' worth of data, over 10 million trading steps. Then the agent is applied to the last year of the data set to trade the two futures contracts. Its performance is evaluated against a long-only benchmark with fixed positions. We find that our RL agent is not able to outperform the benchmark, although the mean returns are mostly not statistically significant. Additionally, the agent often wants to execute trades without having the required capital, which results in a short trading period and thus a less valid result. Upon investigation, we discovered that the trading agent almost exclusively invests in the ZN futures contract and ignores the other one. Furthermore, we find that the agent's contract positions fluctuate throughout the trading period, generating massive transaction costs. The results are further elaborated in section 5.

Evaluation

These issues indicate that either our trading agent could not interpret the capital as a limiting factor in its trading decision and thus did not sufficiently learn the trading problem, or that algorithmic trading with RL does simply not work. With the implementation of different reward functions, we see that the training process of models could be improved and the trading periods prolonged. It could not completely stop the agent from executing too large trades, but it indicates that progress can be made and that we just did not train our agent sufficiently.

For the application of our RL algorithm, we have assumed that financial data contains all information about its past. This is necessary to make use of the MDP formalization of the problem. In reality, however, this assumption does not hold, as financial data is highly non-stationary. Moreover, we assume a constant bid-ask spread for the traded futures. In reality, this bid-ask spread is subject to the prevailing market situation. Although, we rely on intraday data for our trading problem, we are limited to the events between 2010 and 2021. Thus, events such as the Financial Crisis in 2008 were not included in our data but could have contributed to the agent's learning and eventually trading performance.

There are several aspects of this thesis that could be further explored. The first possibility is a longer training sequence for the RL agent, as this could improve its learning and thus its trading performance. The framework of trading in RL can be applied to more trading titles and other asset-classes. Thirdly, additional technical indicators can be analyzed and introduced to the agent. Lastly, the formulation of the reward function can be investigated and how it contributes to the success of the RL trader.